University of Kaiserslautern-Landau,
Department of Electrical and Computer Engineering,
Institute for Wireless Communications (WiCoN)

Relevance Based Radio Resource Management for Machine Learning Units

Doctoral Dissertation approved by the Department of Electrical and Computer Engineering of University of Kaiserslautern-Landau for the award of the degree
Doctor of Engineering (Dr.-Ing.)
to

Afsaneh Gharouni

Dean of the Department:	Prof. Dr. rer. nat Marco Rahm
Reviewer I:	Prof. Dr.-Ing. Hans D. Schotten
Reviewer II:	Prof. Dr.-Ing. Peter M. Rost
Defense Date:	14.12.2023

DE-386

Eidesstattliche Erklärung:

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Munich, 2024/01/29 _____

Afsaneh Gharouni

Abstract:

Machine Learning (ML) is expected to become an integrated part of future mobile networks due to its capacity for solving complex problems. During inference, ML algorithms extract the hidden knowledge of their input data which is delivered to them through wireless links in many scenarios. Transmission of a massive amount of such input data can impose a huge burden on the mobile network. On the other hand, it is known that ML algorithms can tolerate different levels of distortion on their input components, while the quality of their predictions remains unaffected. Therefore, utilization of the conventional approaches implies a waste of radio resources, since they target an exact reconstruction of transmitted data, i.e., the input of ML algorithms. In this thesis, we propose a novel relevance based framework that focuses on the quality of final ML outputs instead of such syntax based reconstruction of transmitted inputs. To this end, we quantify the semantics or relevancy of input components in terms of the bit allocation aspect of data compression, where a higher tolerance for distortion implies less relevancy. A lower relevance level is translated into the allocation of less radio resources, e.g., bandwidth. The introduced formulation provides the foundations for the efficient support of ML models with their required data in the inference phase, while wireless resources are employed efficiently.

In this dissertation, a generic relevance based framework utilizing the Kullback-Leibler Divergence (KLD) is developed that is applicable to many realistic scenarios. The system model under study contains multiple sources transmitting correlated multivariate input components of a ML algorithm. The ML model is seen as a black box, which is trained and has fixed parameters while operating in the inference phase. Our proposed bit allocation accounts for the rate-distortion tradeoff. Hence, it is simply adjustable for application to other problems. Here, an extended version of the proposed bit allocation strategy is introduced for signaling overhead reduction, in which the relevancy level of each input attribute changes instantaneously. In another expansion, to take the effect of dynamic channel states into account, a resource allocation approach for ML based centralized control systems is proposed. The novel quality of service metric takes outputs of ML algorithms into consideration, and in combination with the designed greedy algorithm, provides significantly improved end-to-end performance for a network of cart inverted pendulums.

The introduced relevance based framework is comprehensively investigated by considering various case studies, real and synthetic data, regression and classification, different estimators for the KLD, various ML models and codebook designs. Furthermore, the reliability of this proposed solution is explored in presence of packet drops, indicating robustness of the relevance based compression. In all of the simulations, the relevance based solutions deliver the best outcome in terms of the carefully chosen key performance indicators. In most of them, significantly high gains are also achieved compared to the conventional techniques, motivating further research on the subject.

Kurzfassung:

Es wird erwartet, dass maschinelles Lernen (ML) aufgrund seiner Fähigkeit, komplexe Probleme zu lösen, ein integrierter Bestandteil künftiger Mobilfunknetze wird. Während der Inferenz extrahieren ML-Algorithmen das verborgene Wissen ihrer Eingabedaten, die ihnen in vielen Szenarien über drahtlose Verbindungen übermittelt werden. Die Übertragung einer großen Menge solcher Eingabedaten kann eine enorme Belastung für das Mobilfunknetz darstellen. Andererseits ist bekannt, dass ML-Algorithmen unterschiedlich starke Verzerrungen ihrer Eingangskomponenten tolerieren können, während die Qualität ihrer Vorhersagen davon unberührt bleibt. Daher bedeutet die Verwendung herkömmlicher Ansätze eine Verschwendung von Funkressourcen, da sie auf eine exakte Rekonstruktion der übertragenen Daten abzielen, d. h. auf die Eingabe der ML-Algorithmen. In dieser Arbeit schlagen wir einen neuartigen, auf Relevanz basierenden Rahmen vor, der sich auf die Qualität der endgültigen ML-Outputs konzentriert, anstatt auf eine solche syntaxbasierte Rekonstruktion der übertragenen Inputs. Zu diesem Zweck quantifizieren wir die Semantik oder Relevanz von Eingabekomponenten in Bezug auf den Aspekt der Bit-Zuweisung bei der Datenkompression, wobei eine höhere Toleranz für Verzerrungen eine geringere Relevanz bedeutet. Eine niedrigere Relevanzstufe bedeutet, dass weniger Funkressourcen, z. B. Bandbreite, zugewiesen werden. Die vorgestellte Formulierung bietet die Grundlage für die effiziente Unterstützung von ML-Modellen mit den erforderlichen Daten in der Inferenzphase, während gleichzeitig die Funkressourcen effizient genutzt werden.

In dieser Dissertation wird ein generischer, auf Relevanz basierender Rahmen unter Verwendung der Kullback-Leibler-Divergenz (KLD) entwickelt, der auf viele realistische Szenarien anwendbar ist. Das untersuchte Systemmodell enthält mehrere Quellen, die korrelierte multivariate Eingangskomponenten eines ML-Algorithmus übertragen. Das ML-Modell wird als eine Blackbox betrachtet, die trainiert wird und während der Inferenzphase feste Parameter hat. Die von uns vorgeschlagene Bit-Zuweisung berücksichtigt den Kompromiss zwischen Rate und Verzerrung. Daher ist sie für die Anwendung auf andere Probleme einfach anpassbar. Hier wird eine erweiterte Version der vorgeschlagenen Bit-Zuweisungsstrategie zur Reduzierung des Signalisierungs-Overheads eingeführt, bei der sich die Relevanzstufe jedes Eingabeattributs sofort ändert. In einer weiteren Erweiterung wird ein Ansatz zur Ressourcenzuweisung für ML-basierte zentralisierte Kontrollsysteme vorgeschlagen, um die Auswirkungen dynamischer Kanalzustände zu berücksichtigen. Die neuartige Dienstgüte-Metrik berücksichtigt die Ergebnisse von ML-Algorithmen und bietet in Kombination mit dem entworfenen Greedy-Algorithmus eine deutlich verbesserte End-to-End-Leistung für ein Netzwerk von Karrenpendeln.

Der eingeführte, auf Relevanz basierende Rahmen wird umfassend untersucht, indem verschiedene Fallstudien, reale und synthetische Daten, Regression und Klassifizierung, verschiedene Schätzer für die KLD, verschiedene ML-Modelle und Codebook-Designs berücksichtigt werden. Darüber hinaus wird die Zuverlässigkeit der vorgeschlagenen Lösung in Anwesenheit von Paketverlusten untersucht, was die Robustheit der relevanzbasierten Kompression zeigt. In allen Simulationen liefern die auf Relevanz basierenden Lösungen die besten Ergebnisse in Bezug auf die sorgfältig ausgewählten Leistungskennzahlen. In den meisten Fällen werden im Vergleich zu den konventionellen Techniken auch signifikant hohe Gewinne erzielt, was zu weiteren Forschungen zu diesem Thema motiviert.

Acknowledgments:

Many individuals have assisted me during this journey. I have learned many lessons from them and I am grateful for having them by my side.

Firstly, I would like to thank Prof. Dr. Ing. Hans Schotten for trusting me with the opportunity to join his big team of researchers and supervising my work. I would also like to thank my Nokia Bell Labs supervisors, Prof. Dr. Ing. Peter Rost and Dr. rer. nat. Andreas Mäder. I am truly grateful for your support and mentorship during these years and for what I have achieved. Thank you sincerely for your constructive guidance and goodwill.

I am forever thankful to my mother, Nasrin Khazaei, my aunt, Shima Khazaei, and my uncle, Vahid Zabeti. Without them, I would not have been where I am today. They made a dream come true.

I would like to express my appreciation towards my friends, many of whom went through a similar journey. Hence, they could always provide meaningful support. Thank you, Arman, Gelare, Ghazal, Marton, Mitra, Pegah, Vida, and especially Farnaz for being there for me by understanding my difficulties and fueling me with hope and new insights.

I would also like to extend my profound gratitude to Marton for his companionship, the peace and comfort that he brought to my mind.

Finally, a special thanks to Prof. Dr. Ing. Wehn for chairing my defense and examination, the last bit of my expedition.

Munich, 2024/01/29

Afsaneh Gharouni

Contents

1. Introduction

1.1 Motivation and State of the Art

Artificial Intelligence (AI) and Machine Learning (ML) are envisioned to become an integrated part of future wireless networks due to their capability for solving complex problems. In many studies including [1–8], AI is identified as an enabler for the 6^{th} Generation of Mobile Network (6G). It is also considered a solution for addressing some challenges of the 5^{th} Generation of Mobile Network (5G) and 5G-Advanced in Release 17 and 18 [9, 10]. AI and its power can be employed in various fields from the physical layer and network management to more traditional areas such as localization and object recognition. These AI based services are deployed at user and/or network side. In other words, it is not only expected that AI vastly serves future networks but also that communications systems are capable of serving AI with its massive data requirements. Thus, unlike the state-of-the-art research utilizing ML to solve a problem, we address the gap in supporting Machine Learning Based Units (MLUs) in wireless networks by taking their peculiar characteristics and requirements into account. As a result, training MLUs, introducing MLU architectures and similar topics are out of the scope of this dissertation. Here, our aim is to serve ML based entities operating in inference mode in a network such that best effort MLU performance is achieved with given limited communications resources, or a target on MLU performance is reached with least utilization of network resources. For this purpose, we investigate the relevancy of information for these non-linear units. In this chapter, several well-known fields of study which seem to be remotely related to this subject, but are unable of fulfilling our goal, and afterwards our objectives and contributions are presented.

Many use cases in which AI is the solution or paves the way for performance enhancements are explored in literature. To give some examples, authors of [11–14] consider channel prediction problems, and [15] focuses on a deep learning based millimeter wave massive Multiple-Input Multiple-Output (MIMO) for hybrid precoding. Different aspects of intelligent network management, e.g., resource allocation and mobility management are discussed in [16–19], and [20] surveys various ML approaches applied to communications and security in vehicular networks. Moreover, many Internet of Things (IoT) and Industrial Internet of Things (IIoT) applications require connected devices such as robots and smart drones to perform sophisticated tasks. One of the main candidates to cope with these tasks is ML. Several cases facilitating perceiving and functioning of IoT systems with AI are reviewed in [21]. A more comprehensive overview of the potential role of ML in mobile networks is presented in Chapter 2.

Therefore, considering the functionality of these intelligent elements in design of future communications systems is beneficial in order to both improve system performance and utilize radio resources efficiently, while delivering the required information to the MLUs. In human-to-human use cases, the ultimate goal of communications systems is to deliver the exact syntax to an end user as it was originally transmitted. However, when dealing with intelligent units, such syntax based communications implies a waste of network resources, since MLUs can tolerate distortion, e.g., lossy data compression at their input. As a remotely analogous case, consider the MP3 compression and its concept. Due to the dynamic adaptation of human hearing, parts of sounds become inaudible to most of human ears which can be discarded resulting in huge compression gains without affecting our perception. Hence, the questions arise whether determining the relevant data for MLUs and defining a corresponding measure of relevancy that can be used in network design is possible, and are the conventional metrics such as fairness in scheduling capable of capturing the demands of future wireless systems. On our journey to provide an answer to these questions, instead of just considering a special use case, we look for a generic solution that is applicable to many real world situations, while keeping an eye on the feasibility of translating it into matters which are related to communications systems such as radio resource management and signal overhead reduction.

For a MLU, relevant information can be discussed from various points of view. Feature selection is one area of research aiming at the concept of relevancy with several methods only taking the individual effect of features into account. However, some solutions such as [22] apply the sensitivity measure introduced in [23] to select a subset of relevant features for training process of learning problems. Although these approaches seem to be related to the concept under study in this script, they are performed when designing the MLU and in many cases, by a third-party vendor implying that changing parameters and procedure of MLU design could be out of control. Thus, these approaches are not useful for optimizing the overall system performance from a network perspective. In this dissertation, we look for a problem formulation that targets ML based inference units as black boxes in the network. The same circumstance holds for the sample selection by means of sensitivity analysis, e.g., [24] and [25], in which the size of the training set is reduced while keeping the more relevant samples for the learning process.

Explaining ML and its decisions is another seemingly related category of approaches, which in some cases measures the importance of different input attributes for a given ML model. As an instance, the authors of [26] define a rate-distortion framework and claim that finding a subset of relevant components with a cardinality significantly smaller than the trivial set, while meeting the distortion condition, is not feasible in a systematic manner. Therefore, the problem is reformulated and a relevance score is defined for each input component to explain the behavior of deep neural network classifiers. Sensitivity analysis is another approach that provides measures for quantifying the effect of MLU input attributes on their output. However, once again these scores cannot be used in communications systems when efficient use of resources such as bandwidth is concerned.

Semantic information theory is another subject investigating the relevancy of data. It attempts to expand Shannon's information theory by answering the question of "How much useful or meaningful information?" instead of "How much information?". An overview of different measures for this purpose is presented in [27], where many aspects such as truth, misinformation, comprehensiveness, and providing sufficient discrimination are discussed for each measure. Agent relative information and inconsistency of input are among other topics that are explored. However, semantic information theory is based on a propositional framework, also referred to as statement logic. Consequently, apart from the maturity of the topic, it cannot be employed simply and vastly for communications systems.

Finally, data compression and its methods belong to a related topic for our research questions. Data quantization consist of three main parts: determining the number of clusters, codebook design and assignment of data to the codewords. Most of the current research such as autoencoders and vector quantization generally focuses on codebook design, and assignment of codewords. These solutions, e.g., the one presented in [28], assume a given number of clusters in order to work on a tractable quantization problem or find a suboptimal value using simple trials and errors. However, the first aspect of compression, i.e., determining the number of quantization bits, directly points to the rate-distortion theory and can be used from a communications system point of view. Therefore, in this thesis, we concentrate on the quantization bit allocation.

In information theory and the current literature, the achievable rate-distortion regions for distributed scenarios are only derived for special cases, deploying either a syntax based or a relevance based distortion measure. The syntax based metrics measure the distortion between source sequences and their quantized versions, while the relevance based techniques account for a third variable of interest. All strategies deploying a syntax based distortion violate our purpose for considering the relevancy of data and taking output of MLUs into account. And, among the relevance based solutions, either the simplifying assumptions including Gaussian distributions and conditional independence do not hold in many real world scenarios or the structure of the special use case does not allow for applying the solution to a wide range of problems. The approaches introduced in [29–36] are instances that belong to the former case with simplifying assumptions and as an example, the localization scenario in [37] refers to the latter case. A more elaborate overview of these techniques is given in Section 3.1.1.

Although explaining the behavior of MLUs, especially in presence of dependencies among input variables, is non-trivial, it is known that a MLU input space contains attributes with different levels of relevance and redundancy regarding the output. Accordingly, the severity of performance degradation in response to corrupted inputs depends on the relevancy of features. To this end, as mentioned earlier, we revisit the bit allocation problem and suggest an automated way to determine the levels of distortion that MLUs can tolerate while reducing their prediction errors given a constraint on network resources such as bandwidth. This formulation is afterwards expanded and utilized for Signal Overhead Reduction (SOR), and wireless resource allocation in a network with ML based centralized control system. The proposed solutions are examined for various case studies, including (but not limited to) classification and regression problems, data sets with synthetic and real measurements, different hypotheses from k-Nearest Neighbors (k-NN) and support vector machines to neural networks, considering both scalar and vector quantization for codebook design. Depending on the problem under study, the relevant state of the art is detailed in the corresponding chapter, i.e., Subsections 3.1.1, 5.1.1 and 6.1.1.

1.2 Our Objectives and Problem Description

On our way to formulate the main problem statement for feeding ML based entities in communications systems with relevant information, the following objectives are taken into consideration.

- Accounting for the relevancy and semantics of the input attributes for given MLUs
- Accounting for dependencies and interactions among input attributes
- Applicability of the proposed framework and solutions to a wide variety of real world problems

- Applicability of the proposed framework and solutions to Multiterminal and Multivariate scenarios
- Avoiding prior assumptions on statistics
- Treating MLUs as black boxes in inference mode, since if provided by a third-party vendor, changing the MLU design, e.g., retraining of the model is infeasible.
- The possibility of extending the main formulation and applying it to typical problems of communications systems
- Assuming no application layer information to be employed in the proposed approach

In order to meet these objectives, we assume that while MLUs operate in a network, they remain fixed and unchanged. Therefore, the proposed methods of this dissertation are applicable to any of such ML based entities regardless of the learning paradigm and hypothesis, but they are not applicable to the entities performing adaptive learning that change their parameters during inference.

The main problem statement of this dissertation is to find a bit allocation such that its resulting pattern of input distortion can be tolerated at a given MLU. Equivalently, the bit allocation is capable of keeping the relevant input information for the MLU. Therefore, we opt for a distortion measure $d_{\text{rel}}(\cdot)$ which is a function of both $\hat{\mathbf{x}}$ and $\mathbf{y}$ to take the impact of quantization on MLU input and output into account. The bit allocation, $\boldsymbol{\eta}^*$, is then selected according to

***Problem Statement* 1:**

$$\boldsymbol{\eta}^* = \underset{\boldsymbol{\eta}}{\operatorname{argmin}}\, d_{\text{rel}}(\hat{\mathbf{x}}, \mathbf{y}), \tag{1.1}$$

subject to

$$\eta_n > 0, \tag{1.2}$$

$$\sum_n \eta_n \leq \eta_{\text{sum}}, \tag{1.3}$$

where $\boldsymbol{\eta}$ is a feasible bit allocation, η_n stands for the number of bits allocated for data compression of the nth terminal serving a given MLU, and η_{sum} is the bandwidth constraint translated into the number of bits. More details about the problem statement and the proposed solution are discussed in Chapter 3. This formulation and its corresponding outcome and benefits are thoroughly analyzed for different scenarios in Chapter 4 while a modified version of the solution from Chapter 3 is proposed to fit the conditions of MLUs with high dimensional input. Note that although a full explanation of the MLU behavior is impossible, the potential contributing factors affecting the gains achieved by the relevance based approach are discussed in this chapter.

The next fundamental questions that we seek to answer in this thesis are: "How to use the relevance based bit allocation for SOR?" and "How to extend the basic problem formulation for wireless resource management of a network consisting of MLUs when the channel quality information is available?". These questions are described in ***Problem Statements*** 2 and ***Problem Statements*** 3, in Chapters 5 and 6, respectively.

1.3 Main Contributions

Apart from formulating problems and providing solutions regarding the three raised questions in Section 1.2, we explore other aspects such as the robustness of the relevance based bit allocation in presence of Packet Drop (PD). The main contributions of this dissertation and their corresponding chapters are briefly highlighted here, and a list of own publications is presented at the end of this section.

- Formulating a general framework for capturing the relevancy of MLU input attributes while meeting the objectives of Section 1.2, and proposing a divergence based solution for ***Problem Statement*** 1. → Chapter 3

- Discussing various estimators for the Kullback-Leibler Divergence (KLD) and their impact on the Key Performance Indicator (KPI) of an inverted pendulum on a cart, i.e., steady state error probability. The KPIs are defined according to the case study under exploration. In this chapter, it is additionally explained that caution should be taken in assuming a fixed distribution over input attributes of the MLU when a feedback loop is present in the case study, i.e., MLU decisions impact MLU input. In such scenarios, the MLU output can change this distribution in case of having highly coarse quantization. → Chapter 3

- Modifying the divergence based solution of Chapter 3 for high dimensional MLU inputs. → Chapter 4

- Evaluating the performance of the proposed relevance based bit allocation for a wide range of problems covering regression and classification, data sets consisting of synthetic and real measurements, vector and scalar quantization for codebook design, various ML hypotheses including k-NN, support vector machine, decision tree, Neural Network (NN) and Convolutional Neural Network (CNN), different case studies being inverted cart pendulum and a network of them, indoor environment classification and AI assisted Conditional Handover (CHO). → Chapters 3 to 6, but mainly Chapter 4.

- Studying the sensitivity of the proposed bit allocation to PD which is a common imperfection in communications systems. Simulation results for the indoor environment classification case study show that bit allocations of the KLD approach have higher robustness in case of PD occurrence. → Chapter 4

- Employing the main problem formulation for SOR and extending the concept of relevancy to account for relevant information in time domain, while a challenging problem of handover is to be solved by a given MLU. → Chapter 5

- Introducing a heuristic to shrink the search space of the SOR problem and solving (1.1) – (1.3) with fewer computations. → Chapter 5

- Proposing a solution to deal with the alternating relevant input components for the MLU in our SOR problem. In the AI assisted CHO case study, a particular input can be a more relevant attribute in a given time step but not in all time steps. We overcome this dilemma by suggesting a grouping for the attributes using a heuristic. → Chapter 5

- The individual and joint impact of the bit allocation and the proposed classifier that accounts for the relevancy in time domain are investigated on the AI assisted CHO case study by numerical results. It is demonstrated that the former mainly affects the number of CHO preparations and the latter increases the Radio Link Failure (RLF) and outage rate. → Chapter 5

- Expanding our generic framework to utilize it in wireless resource allocation for a ML based centralized control system considering the case study with a network of inverted cart pendulums. This includes defining a novel Quality of Service (QoS) that takes the relevancy and outcome of MLUs into account and proposing a greedy algorithm to solve the optimization problem. → Chapter 6

- Although in a few cases employing the proposed relevance based approaches deliver no gain comparing with the conventional benchmarks, they provide the best outcome in terms of the KPIs in all our studies without exception. The cases indicating no gain and their characteristics are discussed in Chapter 4. → Chapters 3 to 6
- In most of the scenarios investigated in this dissertation, the proposed divergence based approaches enhance system KPIs significantly. As an instance, a SOR of more than 50% is achieved in AI assisted CHO case study. → Chapter 3 to 6

As it can be seen from the list of contributions, the introduced framework to account for the semantics or relevancy is sufficiently flexible such that it can be applied to many real world problems, as long as the MLU is trained and fixed. The proposed solutions can also be modified to fit the specific use case requirements by experts, as it is done for the SOR to cope with the alternating relevant inputs and reduce the needed computations. Although using the proposed methods requires extra calculations and roughly accurate KLD approximations, it is still feasible to deploy them in practice, since these computations are performed only once and offline. The studies and their numerical results presented in this thesis motivate the shift from syntax to relevance based communications for ML based entities, and are the first steps towards achieving this goal. An overview of the difficulties to reach such a target and suggestions about future steps are provided in Chapter 6.6.

List of Own Publications

- A. Gharouni, P. Rost, A. Maeder and H. Schotten, "Impact of Bit Allocation Strategies on Machine Learning Performance in Rate Limited Systems",
  IEEE Wireless Communications Letters, vol. 10, no. 6, pp. 1168-1172, June 2021.
- A. Gharouni, P. Rost, A. Maeder and H. Schotten, "Divergence-based Bit Allocation for Indoor Environment Classification",
  IEEE 7th World Forum on Internet of Things (WF-IoT), pp. 639-644, 2021.
- A. Gharouni, P. Rost, A. Maeder and H. Schotten, "Relevance-Based Wireless Resource Allocation for a Machine Learning-Based Centralized Control System",
  IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2021.
- A. Gharouni, U. Karabulut, A. Enqvist, P. Rost, A. Maeder and H. Schotten, "Signal Overhead Reduction for AI-Assisted Conditional Handover Preparation",
  Mobile Communication - Technologies and Applications; 25th ITG-Symposium, Osnabrueck, November 2021.

1.4 Outline of the Dissertation

This dissertation is organized as follows. A brief introduction to ML and NNs along with an overview of ML applications in mobile networks are presented in Chapter 2. The main problem statement and the proposed bit allocation approach using the KLD, which is examined for an inverted pendulum on a cart, is introduced in Chapter 3. While studying this regression problem, two estimators and their impact on the outcome are evaluated by numerical results.

Chapter 4 introduces a modification to the KLD based solution of the last chapter for MLUs with high dimensional input. This chapter also provides insights regarding the question of "whether the proposed divergence based bit allocation is capable of delivering

gains in different scenarios in terms of system KPIs", and discusses the factors which can potentially impact these gains. This comprehensive analysis of the divergence based solution considers various aspects including different MLU hypotheses, codebook designs and packet drops.

In Chapter 5, the overhead reduction problem for AI assisted CHO is investigated. The study on radio resource allocation for a ML based centralized control system consisting of MLUs controlling a network of inverted pendulums on carts is presented in Chapter 6, assuming the knowledge of channel quality coefficients. This dissertation is concluded in Chapter 6.6, where a potential outlook based on the current results is discussed.

Each of the Chapters 3 to 6 start with an overview. The overview points out the motivations behind the presented study, addresses the novelty of our proposed solution considering the related state of the art and highlights the main contributions of the chapter. Afterwards, the system model is elaborated and then, we propose a solution for the given problem of the chapter. These sections are followed by a detailed description of the simulation setup and numerical results. Finally, conclusions are drawn from the content of each chapter in the last section.

2. Machine Learning in Mobile Networks

In this chapter, a brief introduction to Machine Learning (ML) is presented in Section 2.1 and as an example, the training process of a Neural Network (NN) along with some essential learning aspects are overviewed in Section 2.2. These sections provide the basic knowledge of their topics to facilitate reading the upcoming chapters of this dissertation. Afterwards in Sections 2.3 and 2.4, ML use cases and studies in the area of mobile networks are surveyed, and the common challenges and obstacles in this specific field are pointed out.

2.1 Introduction to Machine Learning

Machine learning is a field of study and a subset of Artificial Intelligence (AI). It refers to a collection of approaches that imitate humans in learning to reach a goal by extracting patterns and hidden knowledge from data, without any explicit instruction. Support vector machines, decision trees, NNs and even k-Nearest Neighbors (k-NN) are categorized as ML based techniques. ML algorithms are utilized if three conditions are met [38]: There is no implicit or efficient mathematical approach to solve a given problem. A pattern exists,

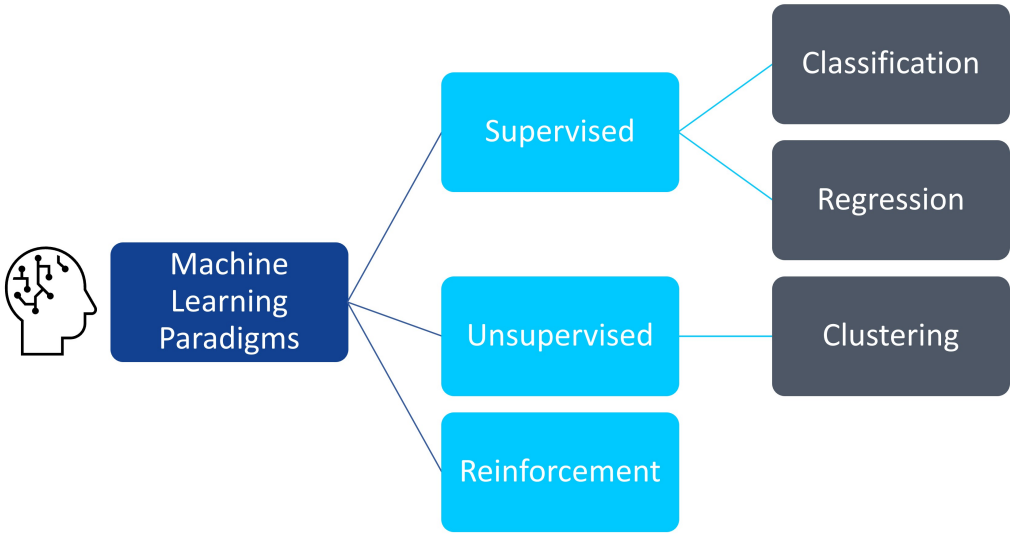


Figure 2.1: Overview of the learning paradigms.

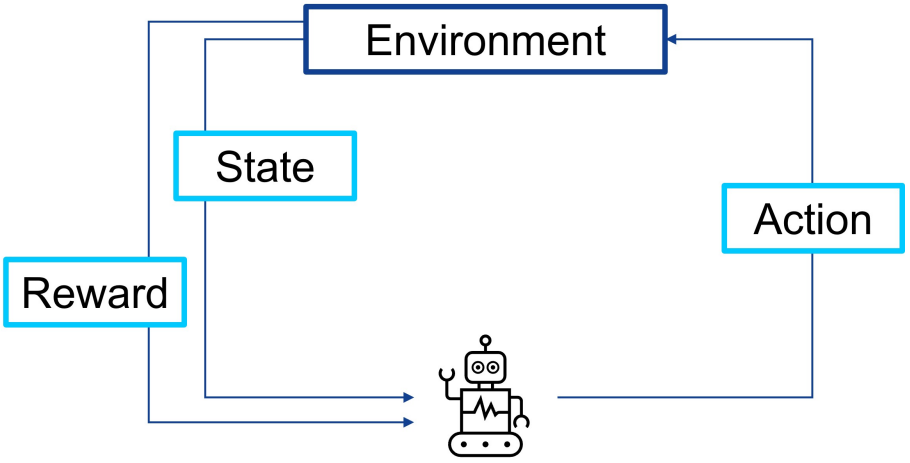


Figure 2.2: A simple illustration for reinforcement learning.

and there exists available data on the problem. While the third condition is a particularly challenging aspect in mobile network use cases, it is important not to overlook the first condition and implicitly assume that a ML algorithm is the best tool to attack all sorts of problems.

An outline of the three ML paradigms is depicted in Fig. 2.1. Supervised learning finds a function mapping inputs to outputs given tuples of input and output samples. Classification and regression are subcategories of this learning paradigm. An example of classification is to feed a Machine Learning Based Unit (MLU) with pictures of different objects and their class labels so that the machine can decide which objects exist in a new picture that it has not seen before. On the other hand, predicting continuous variables like temperature is referred to as regression.

In unsupervised learning, the goal is to learn hidden patterns and categorize unlabeled data, in absence of outputs or tags. A well-known example of this branch is clustering for anomaly detection. The last category is Reinforcement Learning (RL) in which an agent is trained using trials and errors and a reward or punishment system, without accessing explicit output compared with supervised learning. A simple illustration for this type of learning is presented in Fig. 2.2. As it can be seen, after making a decision, the agent gets feedback from the environment about its state and a reward for the taken action. Hence, it can optimize its behavior depending on the state of environment by maximizing the reward. Learning how to solve a tic-tac-toe game is one instance in which RL can be utilized [39]. In the rest of this chapter, we focus on supervised learning.

The main components of supervised learning are a data set containing input and target output samples $\left(\mathbf{x}_j, g_t(\mathbf{x}_j)\right) \in \mathcal{D}$ which are assumed to be noise-free in this section, a target function $g_\mathrm{t} : \mathcal{X} \to \mathcal{T}$, a hypothesis set $\mathcal{G}$ and a learning algorithm. The target function is unknown and we only have access to a set of Independent and Identically Distributed (i.i.d.) samples drawn from joint distribution $p(\mathbf{x}, g_t(\mathbf{x}))$, i.e., $(\mathbf{x}_j, g_t(\mathbf{x}_j)) \overset{\text{i.i.d.}}{\sim} p(\mathbf{x}, g_t(\mathbf{x}))$. The goal of learning is to find a final hypothesis g^* which approximates g_t over $\mathcal{X}$. This is performed by using the data set and a learning algorithm to choose g^* from a hypothesis set. The hypothesis set determines the set of formulas or models, e.g., NNs or support vector machines.

The learning algorithms which pick g^* from $\mathcal{G}$ are addressed in Section 2.2. In the rest of this section, we talk about many aspects such as the impact of the number of samples and a selected hypothesis set on the learning ability. More details on the content of the introductory sections can be found in [38–40]. These discussions are started with

theoretically answering the question of "whether learning is feasible", since given a limited number of samples, learning tries to pick a hypothesis that delivers correct output regarding samples that it has never seen. This matter is referred to as generalization in ML literature. The mathematical framework addressing this question is known as computational learning theory. Here, we briefly introduce the Vapnik-Chervonenkis (VC) dimension and bias-variance tradeoff without providing their mathematical proofs, for binary classifiers and regression with squared error, respectively. Each of these subjects provides insights into learning and can be extended. Afterwards, this knowledge assists us to discuss more practical techniques and learning concepts.

In order to facilitate our discussion, we establish the mathematical notations as follows. Let $E_{\text{in}}(g)$ and $E_{\text{out}}(g)$ stand for the in-sample and out-of-sample errors of hypothesis g. *In-sample* refers to the calculations using an available data set and *out-of-sample* points generally to the whole input space. In theory, $E_{\text{out}}(g)$ represents the out-of-sample error defined over $\mathcal{X}$. In practice, $E_{\text{in}}(g)$ is calculated using a set of samples called a training data set, and $E_{\text{out}}(g)$ is computed using a test set, while the loss function can be the Mean Squared Error (MSE) between targets and MLU predictions, error rate and others depending on the use case. Training and test sets are partitions of the available data samples. Therefore, test samples are not utilized during the training and are capable of providing an estimation for the out-of-sample error $E_{\text{out}}(g)$. In every supervised learning scenario, training samples are employed to find a final hypothesis such that $g^* \approx g_{\text{t}}$. This approximation is performed by reducing $E_{\text{in}}(g)$. Simultaneously, it is vital to ensure that generalization holds, i.e., $E_{\text{out}}(g)$ follows $E_{\text{in}}(g)$ closely.

VC inequality: The VC inequality holds for binary classifiers with continuous attributes and noise-free samples. It provides an upper bound δ on the probability of picking a final hypothesis such that the difference between its in-sample and out-of-sample errors are more than ϵ, assuming any given data set with J samples and any hypothesis picked from $\mathcal{G}$ as the final one. By limiting the probability of this undesirable event, the feasibility of learning, i.e., the feasibility of choosing a hypothesis with similar out-of-sample and in-sample outcomes is proved. The theorem states that

$$p\{|E_{\text{out}} - E_{\text{in}}| > \epsilon\} \leq \delta, \tag{2.1}$$

where both E_{in} and E_{out} are error rates, e.g., number of erroneous decisions divided by J for E_{in}. Dependency on g is dropped, since we assume any g from the hypothesis set can be selected as the final model. In addition,

$$\delta = 8 S_{\mathcal{G}}(J) \exp(\frac{-1}{32} J \epsilon^2), \tag{2.2}$$

where $S_{\mathcal{G}}(J)$ is the shatter coefficient of the hypothesis set $\mathcal{G}$ given J. The shatter coefficient or growth function is defined by

$$S_{\mathcal{G}}(J) = \max_{\{\mathbf{x}_1, \cdots, \mathbf{x}_J\} \in \mathcal{X}} |\mathcal{G}(\mathbf{x}_1, \cdots, \mathbf{x}_J)|, \tag{2.3}$$

where $\mathcal{G}(\mathbf{x}_1, \cdots, \mathbf{x}_J) = \{(g(\mathbf{x}_1), \cdots, g(\mathbf{x}_J)) \in \{0,1\}, g \in \mathcal{G}\}$ considering the binary classification under study. Furthermore, $|\mathcal{G}(\mathbf{x}_1, \cdots, \mathbf{x}_J)|$ denotes the number of different label sequences or dichotomies that can be built by $\mathcal{G}$ given $\{\mathbf{x}_1, \cdots, \mathbf{x}_J\}$. Clearly, we can write $S_{\mathcal{G}}(J) \leq 2^J$ with 2^J being the maximum number of dichotomies in binary classification. However, the shattering coefficient is much smaller than 2^J for many hypothesis sets that are used in practice. Moreover, $S_{\mathcal{G}}(J)$ can be interpreted as the effective size of a hypothesis set and is the maximum number of dichotomies that can be induced given any data set of size J. In other words, the complexity or equivalently, the flexibility of a hypothesis set is abstracted by its shatter coefficient.

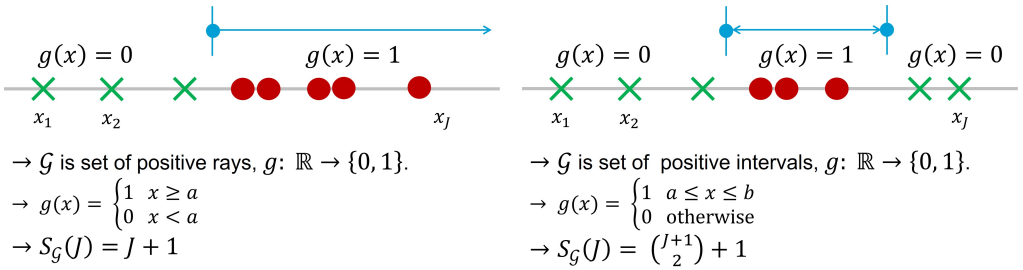


Figure 2.3: An example considering the positive rays and intervals as hypothesis sets to illustrate the shatter coefficient $S_{\mathcal{G}}(J)$ for different $\mathcal{G}$.

Table 2.1: VC dimensions of some hypothesis sets, assuming $d_{\mathbf{x}}$ is the number of input attributes and $\mathbf{x} \in \mathbb{R}^{d_{\mathbf{x}}}$.

Hypothesis set	d_{vc}
Hyperplane	$d_{\mathbf{x}} + 1$
Hypersphere	$d_{\mathbf{x}} + 2$
Quadratic	$\frac{(d_{\mathbf{x}}+1)(d_{\mathbf{x}}+2)}{2}$
r_{p}-order polynomial	$\binom{d_{\mathbf{x}} + r_{\text{p}}}{r_{\text{p}}}$

Fig. 2.3 shows an example to discuss the concept of growth function $S_{\mathcal{G}}(J)$ considering two hypothesis sets with different complexity levels, i.e., the positive rays and intervals. For the set of positive rays as $\mathcal{G}$, the maximum number of dichotomies that we can build is $J+1$ for any combination of J samples. Hence, $S_{\mathcal{G}}(J) = J + 1$. When $\mathcal{G}$ is the set of positive intervals, for any combination of J samples, the maximum number of dichotomies that can be created is $\binom{J+1}{2} + 1$ by placing two ends of an interval in two of $J+1$ available spots plus one case with both ends in the same spot. Therefore, $S_{\mathcal{G}}(J) = \binom{J+1}{2} + 1$. In this example, the set of positive intervals represents a more complex hypothesis set providing more flexibility for classification and has a larger shatter coefficient compared to that of the positive rays.

As mentioned, $S_{\mathcal{G}}(J) \leq 2^J$. If there exists at least one set with J samples for which $\mathcal{G}(\mathbf{x}_1, \cdots, \mathbf{x}_J)$ contains all 2^J different sequences of labels, $S_{\mathcal{G}}(J) = 2^J$. This condition occurs when $J \leq d_{\text{vc}}(\mathcal{G})$, where $d_{\text{vc}}(\mathcal{G})$ is the VC dimension of $\mathcal{G}$. Therefore, the VC dimension of $\mathcal{G}$ is defined as the largest value of J for which $S_{\mathcal{G}}(J) = 2^J$. Equivalently, it is the largest number of samples that the hypothesis set can shatter. For any value of J such that $J > d_{\text{vc}}$, $\mathcal{G}$ cannot shatter any set of points, these values are referred to as break points.

If it is established that a break point for $\mathcal{G}$ exists, it can be proved that $S_{\mathcal{G}}(J)$ is a polynomial of order d_{vc} in J. By replacing these polynomials in (2.2), δ can become arbitrarily small due to the dominance of the exponential term by a proper selection of J and the hypothesis set. This proves that generalization is feasible. For many well-known learning models, either the VC dimension or an upper bound on it is known. As an instance, Table 2.1 indicates VC dimensions for several hypothesis sets, and for NNs a bound on break points is known which is sufficient to prove the generalization ability of this hypothesis set in presence of a sufficient number of samples.

To summarize the analysis, d_{vc} can be seen as the *effective* number of parameters or degrees of freedom that a hypothesis set offers. It is an indicator for the complexity of

a hypothesis set. And, although a higher VC dimension can achieve lower E_{in}, more samples are required for learning to achieve generalization. The VC inequality holds independently of the learning algorithm and for any distribution even if the data set is not a proper representation of the input space, since it considers a worst case scenario and claims probably approximately correct learning. It means that when δ and ϵ are set to small values, probability of having $|E_{\text{out}} - E_{\text{in}}| > \epsilon$ is less than δ with J and hypothesis set selected carefully. By practical observations, it is shown that J and d_{vc} are proportional in the region of our interest with low values for δ and ϵ. As a rule of thumb, $J \geq 10 \times d_{\text{vc}}$ should hold for reaching reasonable generalization [38].

Bias-variance analysis: As it can be noted, there is a tradeoff between approximation and generalization in learning. A hypothesis set with a higher level of flexibility can deliver lower E_{in}. However, a less complex $\mathcal{G}$ implies lower $S_{\mathcal{G}}(J)$ and a better chance of generalization according to the upper bound introduced in (2.1) – (2.2). Another way to study this tradeoff is through the bias-variance analysis, which targets noise-free real valued functions. This theory states that the out-of-sample error can be decomposed into two terms called bias and variance as shown below.

$$\mathbb{E}_{\mathcal{D}}\{E_{\text{out}}(g^{*(\mathcal{D})})\} = \mathbb{E}_{\mathbf{x}}\{\big(\bar{g}(\mathbf{x}) - g_{\text{t}}(\mathbf{x})\big)^2\} + \mathbb{E}_{\mathbf{x}}\{\mathbb{E}_{\mathcal{D}}\{\big(g^{*(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\big)^2\}\}, \tag{2.4}$$

where $E_{\text{out}}(g^{*(\mathcal{D})}) = \mathbb{E}_{\mathbf{x}}\{(g^{*(\mathcal{D})}(\mathbf{x}) - g_{\text{t}}(\mathbf{x}))^2\}$ stands for out-of-sample squared error when any g^* is picked as the final hypothesis using $\mathcal{D}$. Furthermore, $\bar{g}(\mathbf{x})$, known as the average hypothesis, is defined as

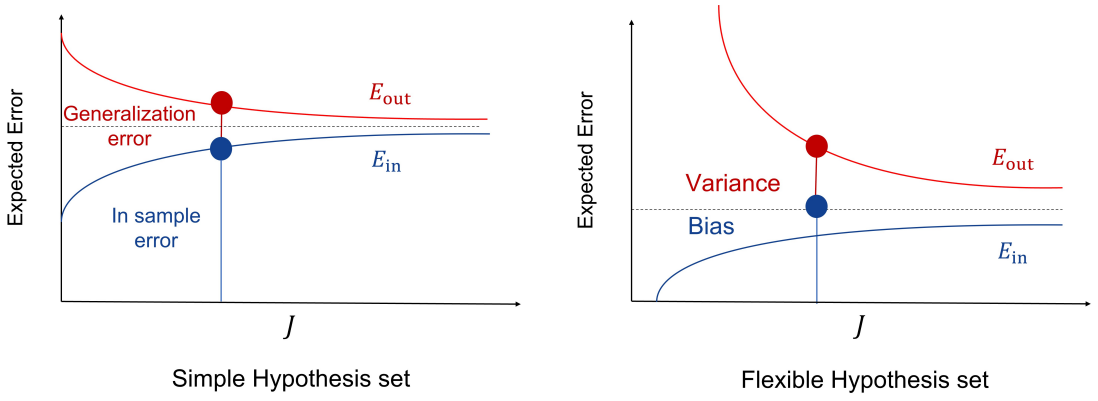
$$\bar{g}(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}\{g^{*(\mathcal{D})}(\mathbf{x})\}. \tag{2.5}$$

The first term on the right side of (2.4) stands for the bias and the second term presents the variance[1]. The bias indicates how far the average choice is from the target function, considering that different functions are picked as final hypotheses given various data sets. A more sophisticated set of models has the capacity of getting closer to the target function and reducing the bias. On the other hand, the variance addresses the deviation between the final choice of hypothesis and the average choice $\bar{g}(\mathbf{x})$, which increases with a flexible set of models.

In summary and similar to the VC inequality, this decomposition demonstrates that for a rigid hypothesis set with a lower variety of models to pick from, the variance becomes smaller, while the bias is increased. The closed form expressions for the bias and variance can be derived, e.g., for linear regression.

Learning curves: In order to provide an abstract description of lessons learned from the VC dimension and bias-variance analysis, Fig. 2.4 depicts expected errors vs. the number of samples J. This figure shows that increasing J leads to finding a better hypothesis with lower out-of-sample error. In the meanwhile, the in-sample error increases because we can generally find a better fit when fewer samples are present. However, the main performance measure is the out-of-sample error that improves with the increasing size of the data set. This behavior holds for both subfigures related to simple and complex hypothesis sets. By comparing the subfigures, it can be observed that a simpler set of hypotheses is capable of reaching better generalization; however, it cannot fit the samples as precisely as a complex hypothesis set. Hence, while using a rigid set is a reasonable choice for low values of J, a sophisticated hypothesis set can provide better out-of-sample errors for a sufficiently large number of samples. Since data sets are given in many learning scenarios and their size is fixed, it is important to select a hypothesis set to properly balance the approximation-generalization tradeoff according to J.

[1]The first term has also been defined as (bias)^2 in literature, e.g., in [40].



(a) (b)

Figure 2.4: Learning curves for rigid and complex hypothesis sets, and demonstration of VC and bias-variance analysis.

In addition, Fig. 2.4 indicates similarities and differences between the VC and bias-variance analyses. The VC inequality can be reformulated, and it can roughly be written as $E_{\text{out}} \leq E_{\text{in}} + \Omega(J, \mathcal{G}, \delta)$, where $\Omega(J, \mathcal{G}, \delta)$ summarizes the dependency of generalization on $J, \mathcal{G}$ and δ. In this case, $\Omega(J, \mathcal{G}, \delta)$ quantifies generalization error in terms of the difference between E_{in} and E_{out} as shown in Fig. 2.4a. The bias and variance concepts are demonstrated using curves of Fig. 2.4b to avoid putting many markers on the same sketch. The bias term represents how well the average prediction over all data sets deviates from the target function. The variance term quantifies how much the solutions achieved with different data sets vary comparing with their average. Therefore once more, the two terms provide measures for quantifying approximation and generalization abilities, respectively.

In practice, instead of dealing with theoretical frameworks, it is usually sufficient to know and remember the tradeoff between generalization and approximation, and how different factors such as J and the complexity of a hypothesis set influence them. In the rest of this chapter, we concentrate on ML concepts from a practical point of view. For this purpose, we begin by investigating a simple NN.

2.2 Introduction to Neural Networks

NNs and their structure, like many other inventions, are inspired by biological systems. These powerful models are made of many nodes and links connecting them. A simple example of a NN known as multilayer perceptron with two hidden layers is drawn in Fig. 2.5, where the first and second layers have three and two nodes called neurons. Since we deal with a scalar function, the output layer has one neuron. This network is fully connected, i.e., all neurons of consecutive layers are linked with each other, while no communication among neurons of the same layer exists. For this network, $\mathbf{x} \in \mathbb{R}^{10\times 1}$ stands for input features, $\boldsymbol{\omega}^{(1)} \in \mathbb{R}^{3\times 10}$, $\boldsymbol{\omega}^{(2)} \in \mathbb{R}^{2\times 3}$ and $\boldsymbol{\omega}^{(3)} \in \mathbb{R}^{1\times 2}$ are the NN weights, and superscripts (l_{NN}) for $\boldsymbol{\omega}$ represent the layer number. The output of each neuron is calculated by feeding the weighted sum of inputs from the previous layer to a transfer function ϕ. After performing computations carried out by all layers, the output or prediction of the NN becomes

$$y = \phi\Big(\boldsymbol{\omega}^{(3)}\phi\big(\boldsymbol{\omega}^{(2)}\phi(\boldsymbol{\omega}^{(1)}\mathbf{x})\big)\Big), \tag{2.6}$$

where ϕ is also called an activation function operating at each neuron of the NN. Fig. 2.6 shows three commonly used non-linear activation functions for NNs, which allow construction of non-linear models. These functions are typically differentiable to facilitate the

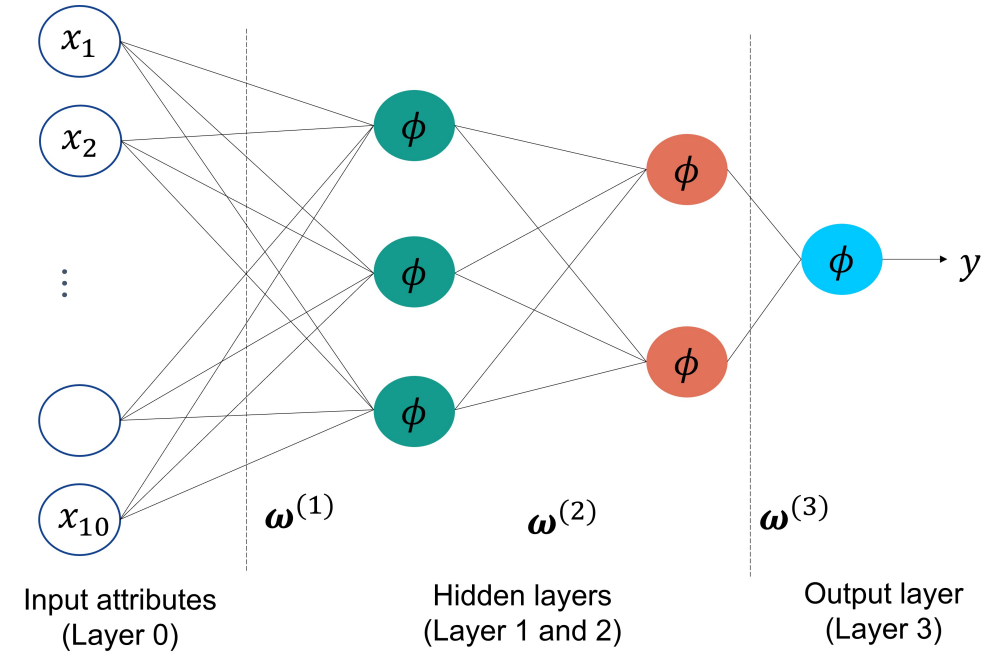


Figure 2.5: A simple example of a neural network known as multilayer perceptron.

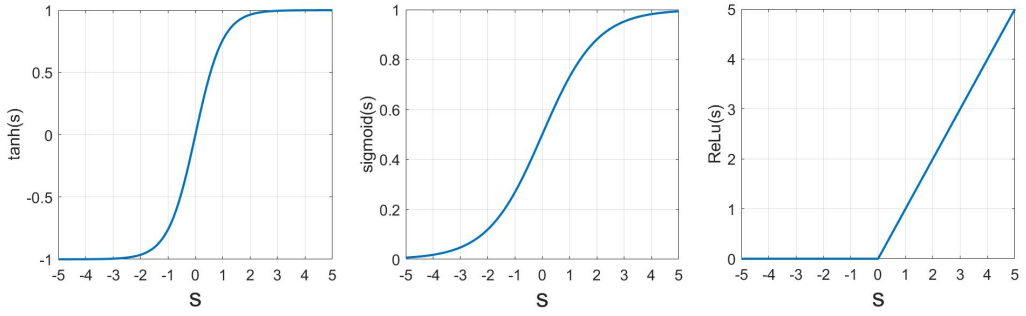


Figure 2.6: Typical activation functions used in NNs, where tanh and ReLU stand for tangent hyperbolic and rectified linear unit, respectively.

gradient based learning algorithm during training. In many cases, ϕ is selected based on the NN architecture, while the choice of activation function usually differs for hidden and output layers. For example, the Rectified Linear Unit (ReLU) is typically used in hidden layers of Convolutional Neural Networks (CNNs). And, for the output layer of regression and multiclass classification, the linear and softmax activations are utilized, respectively. Here, we consider a regression problem with the same ϕ in all neurons to simplify notations.

The introduced NN in Fig. 2.5 is a feedforward network, i.e., computations of each layer are performed in a row during inference, starting from the input to the output layer, without skipping any layer or going backwards. When dealing with the training of the NN, we address backward propagation. This term refers to calculating partial derivatives of the loss with respect to each weight, where loss is propagated backward, from the output towards the input layer with similar conditions, i.e., without skipping any layer or going forward.

In order to elaborate the training process of the NN, new notations should be introduced. Let us concentrate on the (l_{NN})th layer of neurons. The input signal for oth activation function in this layer is shown by $s_o^{(l_{\mathrm{NN}})}$. For $1 \leq (l_{\mathrm{NN}}) \leq L_{\mathrm{NN}}$, we have

$$s_o^{(l_{\mathrm{NN}})} = \sum_{i=1}^{d^{(l_{\mathrm{NN}}-1)}} \omega_{io}^{(l_{\mathrm{NN}})} x_i^{(l_{\mathrm{NN}}-1)}, \tag{2.7}$$

where $d^{(l_{\mathrm{NN}}-1)}$ is the number of neurons connected to the oth activation function from the $(l_{\mathrm{NN}}-1)$th layer, and $\omega_{io}^{(l_{\mathrm{NN}})}$ is a weight associated to the link connecting the ith neuron of layer $(l_{\mathrm{NN}}-1)$ to the oth node of the (l_{NN})th layer. Note that with this notation, when any layer is studied, its neurons are shown by subscript o and the neurons of the previous layer are marked by subscript i. Hence, $x_i^{(l_{\mathrm{NN}}-1)}$ is the output of the ith neuron in the $(l_{\mathrm{NN}}-1)$th layer. In our example, $L_{\mathrm{NN}} = 3$. Moreover, the output of the oth activation function in layer (l_{NN}), $x_o^{(l_{\mathrm{NN}})}$, can be computed based on

$$x_o^{(l_{\mathrm{NN}})} = \phi(s_o^{(l_{\mathrm{NN}})}). \tag{2.8}$$

With these notations, y can also be written as $x_1^{(3)}$ for our NN instance, and $x_1^{(0)}, \cdots, x_{10}^{(0)}$ are representing the NN input components.

In practice, the available data set $\mathcal{D}$ is partitioned into training and test sets, $\mathcal{T}_{\mathrm{train}}$ and $\mathcal{T}_{\mathrm{test}}$, respectively. Then a learning algorithm and $\mathcal{T}_{\mathrm{train}}$ are used to choose a final hypothesis that minimizes the in-sample error. This process is referred to as training and points to the approximation part of learning. Since our hypothesis set is the set of NNs with the described structure in Fig. 2.5, a gradient descent algorithm can find the weights $\boldsymbol{\omega} = \{\boldsymbol{\omega}^{(l_{\mathrm{NN}})}\}$ while minimizing $E_{\mathrm{in}}(\boldsymbol{\omega})$. After performing each step of this optimization, $\boldsymbol{\omega}$ is updated according to

$$\Delta\boldsymbol{\omega} = -\eta_{\mathrm{lr}} \nabla E_{\mathrm{in}}(\boldsymbol{\omega}), \tag{2.9}$$

where $\nabla E_{\mathrm{in}}(\boldsymbol{\omega})$ is the gradient of $E_{\mathrm{in}}(\boldsymbol{\omega})$, $\Delta\boldsymbol{\omega} = \boldsymbol{\omega}(t_{\mathrm{GD}}+1) - \boldsymbol{\omega}(t_{\mathrm{GD}})$, t_{GD} stands for the t_{GD}th step of the algorithm, and η_{lr} is the learning rate. After updating the weights, the in-sample error is computed for new weights and the procedure is repeated until a termination condition is met, e.g., reaching a predefined maximum number of iterations. Moreover,

$$E_{\mathrm{in}}(\boldsymbol{\omega}) = \mathbb{E}_{\mathbf{x}\in\mathcal{T}_{\mathrm{train}}}\{e(\boldsymbol{\omega})\}, \tag{2.10}$$

where $e(\boldsymbol{\omega})$ is the error on each sample of the training set and depends on the NN weights. There are various choices for measuring $e(\boldsymbol{\omega})$. As an example, it can be defined as the Euclidean distance between the target value for the NN output from $\mathcal{T}_{\mathrm{train}}$ and the actual output of the NN given $\boldsymbol{\omega}$. As it can be seen, this algorithm calculates the direction of change for $\boldsymbol{\omega}$ using all the training samples which is computationally expensive.

A commonly used alternative for the gradient descent is the Stochastic Gradient Descent (SGD). This optimization updates $\boldsymbol{\omega}$ by only taking one sample or a subset of all samples into consideration at a time. Here, we focus on the case using one sample. In other words, the SGD employs gradients of the error on one training sample instead of the in-sample error, i.e., the vector of partial derivatives $\partial e(\boldsymbol{\omega})/\partial\omega_{io}^{(l_{\mathrm{NN}})}$ for all $(l_{\mathrm{NN}}), i, o$. Except from the computational superiority, it is expected that the SGD escapes shallow local minimums because of its randomized nature.

For efficient computation of $\partial e(\boldsymbol{\omega})/\partial\omega_{io}^{(l_{\mathrm{NN}})}$ required by the SGD, the chain rule is applied and we can write

$$\frac{\partial e(\boldsymbol{\omega})}{\partial\omega_{io}^{(l_{\mathrm{NN}})}} = \frac{\partial e(\boldsymbol{\omega})}{\partial s_o^{(l_{\mathrm{NN}})}} \times \frac{\partial s_o^{(l_{\mathrm{NN}})}}{\partial\omega_{io}^{(l_{\mathrm{NN}})}}. \tag{2.11}$$

Algorithm 2.1: Backpropagation algorithm and the SGD optimization.

Initialization: $\boldsymbol{\omega}$ randomly
for $t_{\text{GD}} = 0, 1, \cdots$ **do**
  Pick a sample from $\mathcal{T}_{\text{train}}$;
  In forward direction, calculate all $x_o^{(l_{\text{NN}})}$;
  In backward direction, calculate all $\delta_o^{(l_{\text{NN}})}$;
  Update all weights using (2.16);
  Repeat until a termination condition is met.
Return final values of weights.

From (2.7), the second term of (2.11) becomes

$$\frac{\partial s_o^{(l_{\text{NN}})}}{\partial \omega_{io}^{(l_{\text{NN}})}} = x_i^{(l_{\text{NN}}-1)}. \tag{2.12}$$

If we define $\delta_o^{(l_{\text{NN}})} = \frac{\partial e(\boldsymbol{\omega})}{\partial s_o^{(l_{\text{NN}})}}$, for the last layer and the NN output, we have

$$\delta_1^{(L_{\text{NN}})} = \frac{\partial e(\boldsymbol{\omega})}{\partial s_1^{(L_{\text{NN}})}}. \tag{2.13}$$

Furthermore, $e(\boldsymbol{\omega})$ depends on y and the target output from the training set which is a constant value. And, y is equivalently $x_1^{(L_{\text{NN}})}$. Hence, computing $\frac{\partial e(\boldsymbol{\omega})}{\partial s_1^{(L_{\text{NN}})}}$ needs to be performed through the intermediate variable $x_1^{(L_{\text{NN}})} = \phi(s_1^{(L_{\text{NN}})})$ using the chain rule once more. This computation is straightforward assuming, e.g., a tangent hyperbolic activation and $\phi'(s) = 1 - \phi^2(s)$.

The same procedure can be extended to compute the partial derivative of $e(\boldsymbol{\omega})$ with respect to weights of the last hidden layer, recursively. In this case, we need to calculate $\delta_i^{(L_{\text{NN}}-1)} = \frac{\partial e(\boldsymbol{\omega})}{\partial s_i^{(L_{\text{NN}}-1)}}$ and hence,

$$\begin{aligned} \delta_i^{(L_{\text{NN}}-1)} &= \sum_{o=1}^{d^{(L_{\text{NN}})}} \frac{\partial e(\boldsymbol{\omega})}{\partial s_o^{(L_{\text{NN}})}} \times \frac{\partial s_o^{(L_{\text{NN}})}}{\partial x_i^{(L_{\text{NN}}-1)}} \times \frac{\partial x_i^{(L_{\text{NN}}-1)}}{\partial s_i^{(L_{\text{NN}}-1)}}, \\ &= \sum_{o=1}^{d^{(L_{\text{NN}})}} \delta_o^{(L_{\text{NN}})} \times \omega_{io}^{(L_{\text{NN}})} \times \phi'(s_i^{(L_{\text{NN}}-1)}), \end{aligned} \tag{2.14}$$

where $d^{(l_{\text{NN}})}$ is the number of neurons in the (l_{NN})th layer. Similarly for any layer (l_{NN}) assuming a tangent hyperbolic activation, we have

$$\delta_i^{(l_{\text{NN}}-1)} = \left(1 - (x_i^{(l_{\text{NN}}-1)})^2\right) \sum_{o=1}^{d^{(l_{\text{NN}})}} \omega_{io}^{(l_{\text{NN}})} \delta_o^{(l_{\text{NN}})}. \tag{2.15}$$

And therefore, all weights are updated according to

$$\omega_{io}^{(l_{\text{NN}})} \leftarrow \omega_{io}^{(l_{\text{NN}})} - \eta_{\text{lr}} x_i^{(l_{\text{NN}}-1)} \delta_o^{(l_{\text{NN}})} \tag{2.16}$$

This study of error propagation with moving backwards from the last layer of NN towards the first one is referred to as back propagation. A general and simplified overview for the SGD optimization is described in Alg. 2.1. Several modifications can be applied to this algorithm. For instance, $\eta_{\text{lr}} = 0.1$ is a proper fit for starting the optimization as a rule of thumb and it can be adjusted after several iterations. In addition, instead of working only

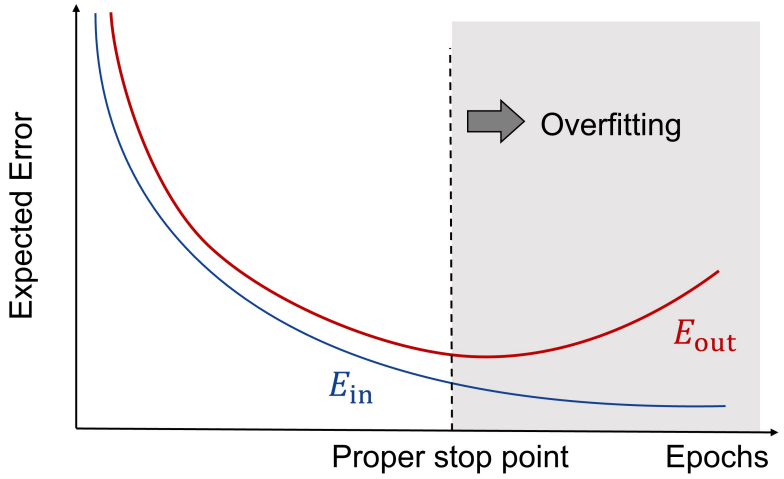
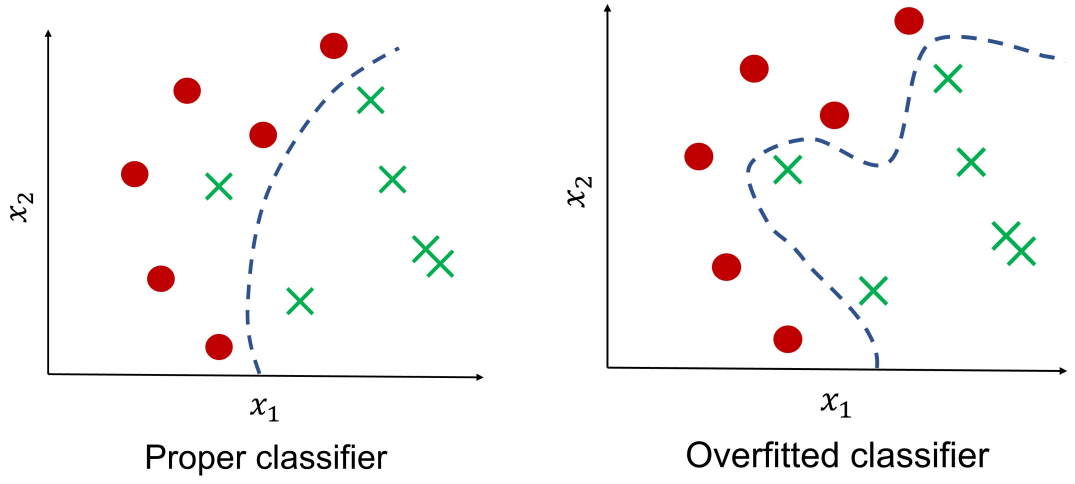


Figure 2.7: Overfitting and impact of termination condition on generalization.



(a) (b)

Figure 2.8: An example of overfitting in presence of noisy samples, where x_1 and x_2 stand for the two input attributes.

on one sample, the algorithm can work on subsets (batches) of training data to benefit from the advantages of both SGD and gradient descent[2].

Apart from having a reasonably sophisticated hypothesis set and sufficiently large J, defining a timely termination for the algorithm presented in Alg. 2.1 is another sensitive subject that directly impacts the quality of learning and its generalization ability. The step through which all weights are updated by going over all samples of the training set is called an epoch. A straightforward approach to determine a stop for the algorithm is to identify a maximum number of epochs. Stopping the algorithm too early avoids the occurrence of the convergence and results in the so-called underfitting. On the other hand, setting this maximum to an unnecessarily large value leads to overfitting. Since the second condition is more likely to happen in learning, we focus on overfitting.

[2]Note that the gradient descent is also referred to as the batch gradient descent in literature, since it works on the whole batch of training samples. It should not be confused with the cases in which batch is referring to a subset of $\mathcal{T}_{\text{train}}$.

When E_{in} is reduced but E_{out} does not follow this reduction, it is said that the model overfits the data as illustrated in Fig. 2.7. This figure depicts in-sample and out-of-sample errors vs. number of epochs. As it can be observed, delaying the stop point causes overfitting. In this case, despite a very low in-sample error, the learned model cannot perform well over the whole input space. To give an example, this problem can arise in presence of noisy data as shown in Fig. 2.8. In this classification, the leftmost green marker among the red ones presents a noisy target in $\mathcal{T}_{\text{train}}$. Training of the classifier in Fig. 2.8a is terminated at a proper point, and although its accuracy on the training samples is less than that of the classifier in Fig. 2.8b, the MLU of Fig. 2.8a avoids changing its parameters to fit the noisy sample. However, for the classifier of Fig. 2.8b, training is not stopped early enough. As a result, overfitting, i.e., fitting even to the noisy sample occurs. That is why this learning unit becomes incapable of generalization and providing accurate decisions on unseen data.

Various remedies including regularization methods are proposed to deal with overfitting in literature, e.g., in [40]. One practical solution is to enforce a so-called *early stop* for the optimization by using a third split of data known as a validation, holdout or development set. This set is used to estimate the out-of-sample error and its estimation is shown by E_{val}. Therefore, when E_{val} increases and diverges from the in-sample error, the algorithm stops iterating and overfitting is prevented. The validation set is also employed for deciding the complexity level of $\mathcal{G}$ and hyperparameters such as the learning rate, batch size, number of hidden layers and neurons of a NN. In order to select suitable hyperparameters, the ones delivering the lowest error value over validation samples are picked and determine the hypothesis set and learning algorithm. Note that E_{val} is utilized to make learning choices and as a result, becomes contaminated. In other words, somewhat similar to the training set, it can no longer provide an unbiased evaluation on the performance of the final hypothesis. For this purpose, the test set is used and $E_{\text{out}}(g^*)$ is the indicator of MLU performance.

The percentages of training, validation and test splits are decided based on the number of available samples and the specific learning problem. By setting a validation set aside, the out-of-sample error of the scenario with fewer training samples deviates from the out-of-sample error of the case using all training samples. This undesirable deviation becomes larger by increasing the number of validation samples. However, having a larger holdout set implies that E_{val} is closer to the actual out-of-sample error. As a rule of thumb, the validation set should contain 20% of the training samples. In situations in which the available data is limited and putting a subset of data aside is harmful to learning, alternative approaches like cross-validation are employed.

To conclude this section, it is worth mentioning that the discussed topics in this and previous sections are introductory to simplify reading the dissertation, and they barely scratch the surface of the ML field. In addition to the more advanced topics such as various network architectures, adaptive learning, deep learning and explaining its behavior, subjects like imbalanced data sets, different performance metrics, adding noise to the inputs during training and dealing with missing values are a few examples of practical aspects to explore before training and deploying a MLU. In the rest of this chapter, instead of talking about ML in general, the specific area of ML in mobile networks is reviewed.

2.3 Use Cases of Machine Learning in Mobile Networks

When it comes to ML and its applications, image recognition and robotics are among the primary use cases which are likely to come to mind. These traditional use cases relate to wireless networks in the context of Internet of Things (IoT) and Industrial Internet of Things (IIoT). Fig. 2.9 provides a picture of several potential areas in which connected

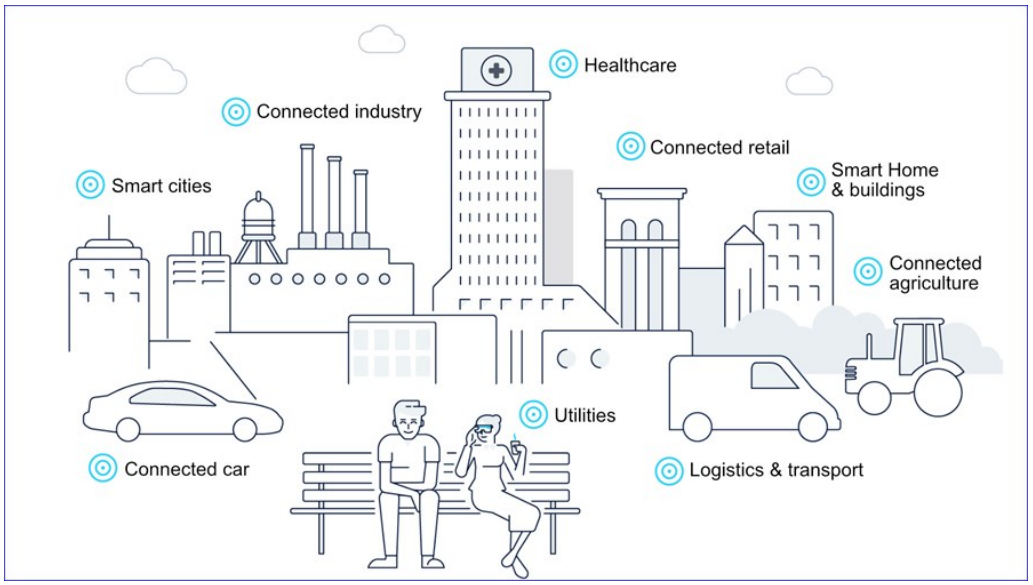


Figure 2.9: An illustration of some future IoT use cases [41].

devices are foreseen to be utilized in the future. A significant impact of ML in many of these areas is expected since devices are the key players performing tasks through the exchange of information and analysis of big data. For instance in autonomous driving, cars need an understanding of the environment for their actions. This perception is achieved, e.g., by employing ML to recognize nearby objects. With the idea of connected industry, robots facilitating manufacturing may use ML for their positioning. Anomaly detection in health related data is another domain for ML utilization. Since exploring all these fields is impractical, we focus on ML and its direct application to problems of the mobile network itself.

During recent years, ML deployment is examined for a wide range of case studies in the field of communications systems, serving different layers of networks from the physical layer to network management. In some scenarios like the 2.4 GHz indoor environment classification, a simple shallow NN or k-NN are shown to deliver desired outcomes [42,43]. In some cases, more sophisticated structures such as deep learning are required and employed, e.g., for the network slicing in [44, 45] and potential applications introduced in [46]. In the rest of this section, we briefly discuss some selected use cases that currently gain attention among researchers. These examples provide the background and fundamentals for our next discussion on the challenges of employing ML in mobile network problems. For this purpose, two physical layer use cases including signal detection and Multiple-Input Multiple-Output (MIMO) beamforming, and handover management are explored.

Signal detection: This topic and its ML based solutions are recently surveyed in [47]. Various ML models such as CNN, Recurrent Neural Network (RNN), autoencoder, and mostly Deep Neural Network (DNN) are deployed for signal detection due to their learning power. Naturally, the input attributes for some of these methods are both the received signal and the Channel State Information (CSI). However, some studies only use the received signal resulting in a reduction of the computations and less power consumption by eliminating channel estimation. It is shown that these MLUs outperform the conventional least-square and Minimum Mean Square Error (MMSE) approaches, e.g., in [48] employing fully connected DNNs in Orthogonal Frequency-Division Multiplexing (OFDM) systems. It is worth mentioning that although the reduced computational complexity by eliminating the channel estimation block is a valid point, employing an estimator might be necessary for other use cases, disproving the aforementioned advantage in a bigger picture. This

is an indication that such modular studies however practical and insightful could lead to suboptimal overall solutions.

In addition, some authors offer a combination of classical and ML based approaches. A deep learning aided tabu search detection is introduced in [49]. The tabu search is a computationally efficient method which is capable of providing a performance near to the maximum likelihood bound in large MIMO systems. Authors of [49] use a so-called fast convergence sparsely connected detection network (FS-Net) to deliver an initial solution for the tabu search. The numerical results for a 32×32 MIMO system with the Quadrature Phase Shift Keying (QPSK) modulation indicate 90% gain in computational complexity while delivering similar performance as that of the tabu scheme.

Each of the proposed solutions for signal detection are applicable to a limited range of systems and parameters determined by the channel model under study, modulation order, number of antennas, etc. The proposed detection network (DetNet) in [50,51] performs well on i.i.d. Rayleigh fading channels with modulations of low order and MIMO systems having much less number of transmitter than receiver antennas. The adaptive deep learning scheme, MMNet, presented in [52] can be applied to the spatially correlated 3rd Generation Partnership Project (3GPP) MIMO channels, for higher modulation orders and a large number of antennas. The MMNet can deliver gains in terms of Signal-to-Noise Ratio (SNR) comparing with the MMSE detector. However, the adaptive nature of this approach imposes extra latency on wireless networks. This paper also demonstrates that many of the proposed learning approaches achieving promising results on simple channel models like i.i.d. Gaussian suffer from performance degradation in presence of spatial correlation, and it motivates the study of more realistic channel models. It is worth noting that authors of [52] published their codes and 3GPP channel data.

In general, performance and computational complexity are the commonly investigated Key Performance Indicator (KPI)s for learning based signal detection. Some of the introduced MLUs only outperform linear receivers with lower complexity by elimination of matrix operations such as inversion, while some MLUs can guarantee near-optimal performance with reduced complexity with respect to the classical existing algorithms. As expected, the applicability of each solution depends on the scenario from which their training data is extracted. Therefore, since some approaches achieve promising outcomes considering simplified assumptions, their generalization and advantages for realistic scenarios need to be investigated in the future.

MIMO Beamforming: According to [47], the state of the art using ML for beamforming is mainly used either to generate or select beamformers or to extract important features facilitating this process. The solutions outputting a beamforming codebook employ unsupervised learning due to the unavailability of targets and imposed limitations on generating them. In general, finding suitable labels with exhaustive search is computationally expensive, and using classical methods to apply supervised learning restrains the performance of the induced MLU to that of the conventional approach. On the other hand, unsupervised learning has the power to cope with this obstacle. Hence, DNNs are designed to generate the beamforming matrix of a MIMO broadcast channel in [53]. The DNN trained with unsupervised learning delivers better performance, close to that of the weighted-MMSE while reducing its computational complexity.

In MIMO beamforming, supervised learning is commonly used in order to select codewords from the beamforming codebook. For instance, [54] studies k-NN, support vector classifiers and DNN models for the uplink of a Millimeter-Wave (mmWave) MIMO scenario, where the configuration selection is translated into a classification problem. Numerical results show that the performance of the DNN-aided solution is near-optimal and outperforms those of the k-NN and support vector classifiers. Unlike [54], authors of [55]

predict achievable rates in case of using various beamformers from the codebook, and the one with the best prediction is selected for beamforming in a highly mobile mmWave system. Simulations based on ray-tracing show gains achieved by the proposed technique in comparison with non-intelligent coordinated beamforming approaches, especially for high-mobility large-array scenarios.

The second category of solutions for MIMO beamforming deals with abstracting the knowledge in form of scalars, which are afterwards used for assisting beamforming. Therefore, the number of training samples can be smaller compared to that of the former category generating beamforming matrices, implying a reduction of overall computational complexity. In [56], CNNs are used to extract some key features from channel information for downlink beamforming in Multiple-Intput Single-Output (MISO) systems. These key components are then fed to a recovery block to construct beamforming matrices. The three trained CNNs are designed regarding Signal-to-Noise-plus-Interference Ratio (SINR) balancing, power minimization and sum rate maximization, and reduce computational complexity. It is shown that near-optimal results are achieved for the first two cases which use supervised learning. The third solution provides performance close to that of the weighted-MMSE while employing a hybrid learning paradigm. A similar setup in [57] is used to design a so-called beamforming prediction network (BPNet) attacking joint optimization of power allocation and virtual uplink beamforming. This proposed solution outperforms the outcome of the weighted-MMSE with a much lower computational complexity.

RL and federated learning are also among the ML based methods which are studied for MIMO beamforming [47]. For instance, a deep RL approach is considered in [58] for passive beamforming through the Reconfigurable Intelligent Surface (RIS) in multiuser MISO systems. Federated learning is a distributed technique as opposed to a central one. In the federated learning based approach of [59], the CNN model is trained at the base station using the collected gradients from users. Compared to the centralized case, the proposed solution offers a higher tolerance to imperfections in channel information and lower computational complexity.

Handover management: Increasing number of base stations per unit area, employment of mmWave and THz spectrum, and having a large number of beams and antennas introduce new challenges to handover management in future networks. Handover management is explored in terms of both base station and beam selections. Here, with one exception, we concentrate on base station selection for mmWave environments, since the topic of ML based beamforming is already covered.

Due to the peculiarities of channels in mmWave and THz range like high penetration loss, detection of obstacles in surroundings using images and methods of computer vision are considered to assist timely handover triggering. Authors of [60] developed a framework that generates data sets for four outdoor mmWave scenarios using a 3D-modeling and ray-tracing software. These data sets contain images, depth maps, wireless channels and user location, and can be accessed via [61]. This framework is then used in [62], where link blockage prediction is performed by an object detector and a gate recurrent unit, using images and information on previously selected beams as input. The blockage predictions are afterwards utilized by the wireless network to initiate handovers. For the studied case, blockage prediction and handover accuracy of up to 90% and 87% are reported, respectively.

Authors of [63] utilize Light Detection and Ranging (LIDAR) sensors, another form of visual data, in a framework that handover optimization is performed through mmWave beam selection. In this paper, a MLU predicts line-of-sight and non-line-of-sight links. These state estimations, LIDAR data, user and base station positions are used to select several candidate beam pairs. Afterwards, the base station is informed to select the best beam

for data transmission. This approach employs deep CNN and a distributed architecture that reduces the signal overhead of beamforming. The centralized version of this proposal is studied in [64]. It is shown that the distributed structure outperforms the centralized one in non-line-of-sight scenarios. This gain is achieved at the cost of providing all users with expensive LIDAR sensors, while in the centralized scenario only the base station is equipped with LIDAR.

Many studies feed the MLU with sensory data such as CSI, received signal power and user location. Observing post-handover trajectories and blockage of line-of-sights along with a multi-armed bandit RL allow for longer connection times achieved in [65]. Authors of [66] account for both handover and power allocation of a heterogeneous network containing macro and mmWave cells. This scheme maximizes the overall throughput, while the frequency of handovers is reduced. For this purpose, first, a global multi-agent RL model is trained in central mode and then, each user obtains a decentralized policy that is executed given local observations. Multi-agent RL is also employed in [67] and [68] for dense mmWave networks. In these studies, signal overhead and computational complexity are decreased because of the utilization of the distributed learning approaches.

Furthermore, a load balancing mechanism is proposed for multiuser mobile mmWave networks in [69]. This study jointly accounts for handover and resource allocation, where a backup base station is selected by the learning algorithm and resources are allocated in order to ensure a target rate and maximize sum rate. The RL based approach of this paper is a deep deterministic policy gradient and targets to associate all users to their optimal base station. Applying ML to handover management is widely investigated in the literature, and mostly RL algorithms are used to tackle this problem. A more elaborate overview of these learning based methods can be found in [70].

Based on the provided examples, a commonly used performance indicator isthe accuracy of MLUs, and depending on the target of the optimization problem under study, a few more parameters such as throughput, latency and frequency of handover are investigated. However, several essential aspects influence each other in mobility and handover management, and all of them should be explored in order to lead us to proper evaluation and conclusions. For instance, decreasing the frequency of handovers even with an increased sum rate may occur at the cost of unwanted service interruptions, or it may imply a desirable reduction in the number of ping pong events. In addition, similar to the previously mentioned use cases, a fair comparison of proposed approaches is infeasible due to the lack of having the same benchmark and platforms. Nevertheless, the state-of-the-art solutions provide insights about potential opportunities of applying ML to handover management use cases and motivate further research and future directions.

2.4 Challenges of using Machine Learning in Mobile Networks

In this section, an overview of shortcomings and problematic aspects of studying and applying ML to use cases of mobile networks are gathered under three categories. Each of these discussed challenges also determines an opportunity to address the current gaps that exist in this field and its state-of-the-art research.

Quality of data and simplifying assumptions: As stated earlier, one of the prerequisites allowing the application of ML is the availability of data with a sufficient number of samples and proper quality to accommodate approximation and generalization aspects of learning. Recent IEEE competitions like [71, 72] attempt to provide platforms for collecting data regarding telecommunications problems. However, when dealing with mobile networks, the scarcity of publicly available, high quality and large enough data sets is tangible compared to mature learning related areas such as computer vision and natural language processing.

One primary reason behind this shortcoming is that mobile service providers, the main owners of network big data, are resistant to share their information due to a lack of clear profits in addition to serious privacy and data protection concerns. Furthermore, owing to the complex nature of wireless network use cases, many parameters are intertwined and impact each other. Generating or collecting data taking all or a considerable portion of these factors into account mostly requires huge investments, which is unattainable in many cases due to a lack of resources. Hence, modular design and data set generation with simplified assumptions are currently preferred in many studies, at the cost of compromised data quality and generalization ability.

The shortage of high quality data with a sufficient number of samples is frequently reported for use cases of wireless networks, e.g., in [46, 70]. For instance, after reviewing the data sets regarding link quality estimation, [73] concludes that the data set of [74] is the best candidate for learning based solutions. In spite of that, even this data set requires improvements in order to account for aspects such as non-artificial noise. Authors of [73] also motivate researchers to address the gap in link quality estimation, where a heterogeneous network is studied and interference is not regarded as background noise.

For mobile network use cases, only specific scenarios own a publicly available data set with real measurements like 2.4 GHz indoor environment classification [42, 43]. However, many data sets used in the literature are generated with simulators, e.g., in [70], which are understandably limited with respect to their abilities for capturing realistic environments. Network simulators employ simplified assumptions and are developed for various platforms, restricting experts from making general conclusions. The challenge of having fair evaluation and consequently, comparisons of various solutions is further elaborated in the following.

Competitiveness of ML solutions and performance evaluation: Another essential challenge that arises in mobile network use cases is the variety of data sets, platforms and conditions assumed for a given use case such as handover management as explained in [70]. This variety and absence of unified benchmarks prevent a fair comparison of the proposed ML algorithms. In addition, for drawing reasonable conclusions and performance evaluation of ML based approaches, a comprehensive set of KPIs should be investigated. Applying a ML algorithm to a problem can improve one KPI while degrading another performance indicator. However, several studies only work on a limited number of KPIs while overlooking the mutual impact of different existing parameters in the system. Consideration of a few and dissimilar KPIs also affects our ability for making comparisons. To give an example, [73] names five important KPIs which are capable of delivering a meaningful evaluation of proposed link quality estimators and emphasizes that none of the reviewed studies consider all of them.

Furthermore, it is worth mentioning that many studies on mobile network topics employ supervised learning and RL, e.g., several use cases discussed in [46, 47]. However, optimal labels are not available for many scenarios introducing another problematic aspect. As a result and as mentioned earlier, many labels are generated by locally optimal solutions restricting the performance of ML. The alternative approach to attack this problem is to learn in an unsupervised manner. For example, autoencoders are used for unsupervised anomaly detection in [75]. In order to discover ML potentials and have a fair estimation of its abilities, unsupervised learning and its power should be explored.

Signaling overhead, restricted resources and time constraints: In wireless communications, limited computational, memory and power resources of mobile users, the sensitivity of some use cases to delays and packet loss, signal overhead, bandwidth constraints and their corresponding tradeoffs introduce additional difficulties to the design of ML based

solutions. In the following, two typical situations in which dealing with these tradeoffs turns out to be challenging are briefly described.

MLUs and specifically deep learning, which is frequently used for mobile network applications, require a massive amount of data to perform their functionality during both training and inference modes. Depending on the deployment method, data needs to be transmitted for analysis, implying signaling overhead and high demands for communications resources. When a central ML model is developed, it is expected that the model delivers a better outcome by taking the global environment and optimizations into consideration. This is achieved at the expense of increased signaling and delays. On the other hand, distributed learning reduces signal overhead and computational complexity, however, compromises performance. These reductions are gained, e.g., when local MLUs act on their local information.

Moreover, because of the dynamic environment of mobile networks, updating ML models is necessary in many use cases. In such scenarios, periodic or constant allocation of resources is required and time constraints for performing calculations are imposed. The ML model can be both trained and updated online, or a combination of offline and online training can be utilized. The promising advantage of offline training is overcoming memory, time and computational problems. However, use cases like handover management often need real-time training according to [70]. As a result, additional solutions should be proposed to tackle this difficulty, e.g., hardware acceleration [76] and reducing the number of parameters to be trained which is performed by clustering in [77]. In addition, online adaptive training is subject to adversarial attacks injecting fake data [78], another issue to consider and prevent from occurrence. As it can be seen, coping with these tradeoffs and limitations that naturally exist in mobile network use cases is not straightforward and requires careful assessment and ideally, accounting for end-to-end achieved gains with simultaneous consideration of many parameters.

2.5 Summary and Conclusion

In this chapter, after providing the basic knowledge of ML, we focus on wireless network use cases employing ML algorithms in order to provide a framework for further discussions of this thesis. Then, the specific challenges of applying ML to such problems are presented from which the scarcity of public data can be seen as the main obstacle. This shortage was also tangible in our research. We could not find a publicly available data set for the multiterminal scenario that is considered in this dissertation. Hence, the reception of data from multiple sources is assumed, which is elaborated in respective case studies and chapters. It is also worth mentioning that among the studied use cases in this thesis, one is facilitated with real data measurements and the rest employ data sets generated by simulators.

Although our goal is not to solve a problem using ML but serving MLUs in networks, many of the explained challenging issues are relevant to our work and are taken into account in our objectives and case studies. In our research, the KPIs are defined such that an appropriate framework for inclusive evaluations is established, and more importantly, oversimplifications are avoided. In addition, benchmarks are carefully selected to provide comprehensible and sensible comparisons. In this dissertation, our focus is to provide ML input data using the least possible wireless resources. This concept is also directly applicable to the discussed signaling overhead problem. The bit allocation strategy in general, and the specific solution tailored for AI-assisted conditional handover prediction represent our proposals regarding signaling overhead reduction.

As it can be observed based on the information provided in this chapter, despite several ongoing research, applying ML to mobile network use cases is at a primitive stage, and

there is enormous room for further enhancements. Studying end-to-end scenarios and more than one use case, where the possibility of information reuse is considered, are among numerous potential subjects to explore in the future. In line with this state, radio resource management for MLUs in mobile networks is another subject to be addressed. This topic is barely considered in the literature as elaborated in Section 1.1. In the upcoming chapters, we concentrate on serving MLUs in wireless networks, and consequently, future research directions are determined in more detail in Chapter 6.6.

3. Relevancy in Terms of Divergence Based Bit Allocation

3.1 Overview

As discussed earlier in Chapter 1, Machine Learning Based Unit (MLU) input space contains attributes with different levels of redundancy and relevance regarding the output. Accordingly, severity of performance loss in response to corrupted inputs depends on relevancy of the features. Explaining this behavior is not trivial, especially in presence of dependencies among input variables. To this end, we revisit the bit allocation problem and suggest an automated way to determine levels of distortion for input attributes that a given MLU can handle while delivering the best effort performance on its predictions given a bandwidth constraint.

From a different perspective, we formulate a problem statement to capture relevancy in terms of required resolution patterns while quantizing data. The underlying reasons for this formulation is the presence of quantization in all communications systems and its direct relation to problems such as signal overhead reduction and wireless resource allocation which are studied later in Chapter 5 and 6, respectively.

3.1.1 State of the Art

The tradeoff between compression and accuracy is a well-known dilemma in lossy quantization. Due to the complexity of distributed scenarios, i.e., multiterminal cases, achievable rate distortion regions are derived only for special cases. These studies can be categorized into syntax and relevance based solutions. The syntax based category presents approaches measuring the distance between source sequences $\mathbf{x}$ and their decoded versions $\hat{\mathbf{x}}$. The rate distortion theory [79], Wyner-Ziv coding and its network extension [80, 81], quadratic Gaussian multiterminal source coding [82] and multiterminal source coding for two encoders under logarithmic loss [83] belong to this first group. These solutions provide the basis for establishing reliable human to human communications. However, syntax based reconstruction of messages is not an optimal criterion when dealing with MLUs operating as inference units in wireless networks. In these cases, achieving a high accuracy on final MLU predictions $\mathbf{y}$ determines the system performance.

The relevance based category of solutions targets to compress $\mathbf{x}$ while preserving the relevant information for prediction of $\mathbf{y}$, by considering a distortion measure which is a function of final MLU outputs. This differs from syntax based distortion measures targeting the

distance between original message and its reconstruction. The current relevance based methods are also tailored for special cases assuming prior knowledge on statistical relation among random variables or their probability distributions. For instance, information bottleneck is a rate distortion function compressing one random variable x in a single encoder-decoder system, where mutual information between the quantized message and another variable of interest y is the distortion measure [29, 30]. The objective function of this optimization problem has also been used for quantization codebook design [84].

Several studies attempted to extend information bottleneck for distributed quantization with multiple sources. Multivariate information bottleneck introduced in [85] employs Bayesian networks for this purpose. In this study, the optimal assignment form is derived. However, the optimality of this proposal in terms of determining rate distortion regions is not discussed, and its cost function has not been used to select number of clusters in literature. It should also be noted that Bayesian network determination is generally far from trivial for ML tasks. Authors of [31] characterize the rate distortion region of distributed information bottleneck for discrete and vector Gaussian sources assuming conditional independence of observations given the main signal of interest which does not hold in many learning problems.

The Chief Executive Officer (CEO) problem considers estimation of a data sequence using its independently corrupted versions observed by different agents [32]. These observations are quantized and communicated to a single decoder. The general formulation of CEO problem can be accounted as relevance based compression, however, its rate distortion region is only investigated for special cases which are not applicable for learning paradigms. The Gaussian CEO [33–35] addresses corruptions caused by additive white Gaussian noise. This simple setup cannot comply with complicated MLU models. As another example, [83] provides the rate distortion region of m-encoder CEO problem conveying information regarding another random variable under logarithmic loss. Similar to all CEO setups, this study assumes conditional independence of observed sequences given the original data, a condition that is not met in many learning scenarios. Considering the mentioned aspects, these CEO studies have not been evaluated for learning tasks.

In a more practical case, 1-bit rate allocation for localization in wireless sensor networks is studied in [37] considering both decoding and localization error, a combination of conventional and relevance based distortion measures. Furthermore, several feature discretization techniques determine the number of quantization intervals for MLU input components using information theoretic metrics. These methods operate on each attribute, independently, making them incapable of accounting for redundant information stored in different attributes. For instance, [36] employs mutual information between a single attribute and MLU output without taking other attributes into account. Thus, the achieved classification accuracy is lower compared with other benchmarks for several data sets. These studies are not applicable to the problem of our interest as discussed in 1.2. Hence, the rest of this chapter is devoted to propose and investigate a more generic solution which can be applied to a wide range of real-world scenarios.

3.1.2 Main Contributions of the Chapter

Fixed-rate quantization has three main aspects: rate or bit allocation, codebook design, and assignment of random variables to codewords. Here, we focus on integer-valued bit allocation for multiple correlated sources performing scalar uniform quantization with arbitrary distributions while MLU is treated as a black box. This includes all non-adaptive Machine Learning (ML) blocks once trained and executing tasks online in network, independently of their hypothesis and learning paradigm such as supervised and reinforcement learning, e.g., the proposed approach can be applied on [58, 86] after the convergence.

Thus, the provided solution can be used in a wide variety of real-world scenarios. An extension for the bit allocation that can be applied in combination with vector quantization is elaborated in Chapter 3.

In this chapter, we propose a criterion using Kullback-Leibler Divergence (KLD) to measure quality of bit allocations. The KLD approximation is performed and discussed for two non-parametric approaches: histogram with smoothing and k-Nearest Neighbors (k-NN). Then, performance of the proposed method is investigated for a cart inverted pendulum with a Machine Learning Based Controller (MLC) which is a shallow Neural Network (NN). The results are compared with those of equal bit sharing and a Mean Squared Error (MSE) based approach inspired by asymptotically optimal integer-valued bit allocation for Gaussian distributed random variables from [87].

Simulation results show significant gain in system performance for low bit rate region. The system performance is evaluated in terms of steady state error probability for the inverted pendulum use case. It can also be seen that a lower quantization noise can be tolerated on two of the features compared to other random variables. The main contributions of this chapter are published in [88] and are listed below.

1. Constructing a generic framework to quantify the relevance of MLU input attributes which are received from multiple terminals in a mobile network.
2. Providing a solution for the defined quantization bit allocation problem in which MLU output has a direct impact on selected bit allocations.
3. Discussing two methods to estimate KLD in a regression problem and their impact on Key Performance Indicator (KPI) of the system.
4. In region with limited bandwidth, the proposed approach achieves significant gains in terms of steady state error probability, the KPI for evaluating system performance of an inverted pendulum.

This chapter is organized as follows. The system model is discussed in Section 3.2. In Section 3.3, the bit allocation approach and KLD estimators are introduced. The simulation setup is elaborated in Section 3.4, and numerical results are presented in Section 3.5. Finally, conclusions are drawn in Section 3.6.

Important Notation: Linear-Quadratic Regulator Controller (LQR) matrices $\mathbf{K}$, $\mathbf{Q}$ and vectors are typeset boldface. $\mathbf{x} = [x_1, \cdots, x_N]$ and $\hat{\mathbf{x}} = [\hat{x}_1, \cdots, \hat{x}_N]$ are vectors of non-quantized and quantized MLU input components, and $\mathbf{y}$ represents MLU output. The nth element of these vectors is denoted with subscript $(\cdot)_n$ as in x_n. The input vector of LQR with fixed bar mass and length is shown as $\mathbf{x}_{LQR}$. $p_{\hat{\mathbf{X}},\mathbf{Y}}(\hat{\mathbf{x}}, \mathbf{y})$ also shown as p, stands for the joint input-output distribution of the MLU assuming a highly accurate quantization. The joint MLU input-output distribution for a given bit allocation $\boldsymbol{\eta} = \{\eta_n\}$ is shown as $q_{\hat{\mathbf{X}},\mathbf{Y}}(\hat{\mathbf{x}}, \mathbf{y})$ or simply q. Data set samples for estimation of KLD are indicated as $\mathbf{z}_j = [\hat{\mathbf{x}}_j, \mathbf{y}_j]$. Finally, $\hat{p}(\mathbf{z}_j)$ and $\hat{q}(\mathbf{z}_j)$ are distribution estimations for p, q with data set samples.

3.2 System Model

3.2.1 General Description

As shown in Fig. 3.1, we study a multiple access channel scenario in which N memoryless stationary sources provide real-valued input attributes $\mathbf{x}$ for a MLU. In presence of complex-valued attributes, the real and imaginary parts can be separated and treated as

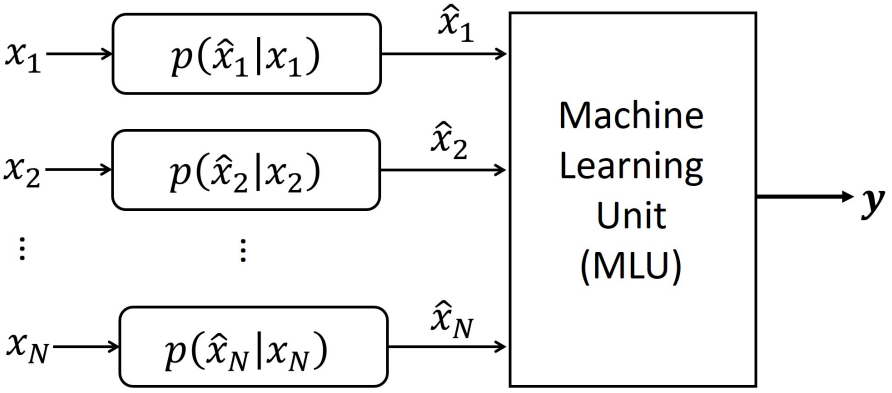


Figure 3.1: Block diagram of the system model.

different random variables. The system performance is evaluated in terms of accuracy on predicting MLU output values $\mathbf{y}$. The scalar uniform quantization with η_n bits for each symbol is performed on random variable of nth source which is shown as $p_{\hat{X}_n|X_n}(\hat{x}_n|x_n)$. It is assumed that quantized vector is received error-free at the receiver. To remove this assumption, $\hat{\mathbf{x}}$ should be redefined to capture the effect of factors such as channel coefficient and receiver noise. Here, we seek to build a system model to be used in practice. So, with no further assumptions, input attributes can be highly correlated and have an arbitrary joint probability density function $p_{\mathbf{X}}(\mathbf{x})$ with $\mathbf{x} \in \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \mathcal{X}_N$.

Given the available bandwidth B and Signal-to-Noise Ratio (SNR) γ, where E_b, T_b and N_0 are energy per bit, bit interval and noise power spectral density, respectively, the capacity of bandlimited channel shown with C_B is $C_B = B \times \log_2(1+\gamma)$ bits/sec. Thus, the constraint for allocating bandwidth B_n to nth source is $\sum_n B_n \leq B$. Assuming same SNR for all terminals, $\gamma_n = \gamma$, and a given symbol interval T_s, the constraint becomes $\sum_n \eta_n \leq \eta_{\text{sum}}$, where $\eta_n = B_n \times \log_2(1+\gamma) \times T_s$ is the number of bits quantizing each symbol of the nth terminal, and $\eta_{\text{sum}} = C_B \times T_s$ bits for each symbol interval. η_n is assumed to be integer-valued as usual in practical systems. The set of feasible bit allocations meeting the constraint are denoted by $\mathcal{H}$. To consider different SNR values, the corresponding possible bit allocations should be added to the feasible set.

In many scenarios, training is performed independently of communications system design and we are not able to modify the MLU, e.g., when the MLU is provided from a third-party vendor. Therefore, it is assumed that learning process is done by non-quantized data and MLU parameters are fixed. In this case, $\sum_n \eta_n >> \eta_{\text{sum}}$ and the joint probability distribution on input and output of the MLU is $p_{\hat{\mathbf{X}},\mathbf{Y}}(\hat{\mathbf{x}},\mathbf{y})$ which is also stated as $p_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})$ to simplify the notation. This distribution is considered as the true distribution and is used as reference to perform comparisons.

Since the MLU model is trained and fixed, and following Markov chain of the system $\mathbf{Y} \leftrightarrow \mathbf{X} \leftrightarrow \hat{\mathbf{X}}$, we can write $q_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}}) = \sum_{\mathbf{x}' \in \mathcal{X}} p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}') p_{\mathbf{x}|\hat{\mathbf{x}}}(\mathbf{x}'|\hat{\mathbf{x}})$ or equivalently, $q_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}}) = \frac{1}{q_{\hat{\mathbf{X}}}(\hat{\mathbf{x}})} \sum_{\mathbf{x}' \in \mathcal{X}} p_{\mathbf{X}}(\mathbf{x}') p_{\hat{\mathbf{X}}|\mathbf{X}}(\hat{\mathbf{x}}|\mathbf{x}') p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}')$, where $p_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}')$ is the fixed distribution learned by the ML, and distribution of quantized data $q_{\hat{\mathbf{X}}}(\hat{\mathbf{x}})$ and conditional distributions on $\mathbf{x}$ and $\hat{\mathbf{x}}$ change for different bit allocations.

3.2.2 Case Study 1: Inverted Pendulum on Cart and its KPI

In order to evaluate performance of bit allocations, we investigate the control problem of an inverted pendulum on a cart. Other case studies are examined in next chapters. The controller is supposed to move the cart to the predefined position $\nu = 0.2\,\text{m}$ in less than 2 sec while the pendulum is in its equilibrium position, i.e., $\theta = 0$, where θ is the angle

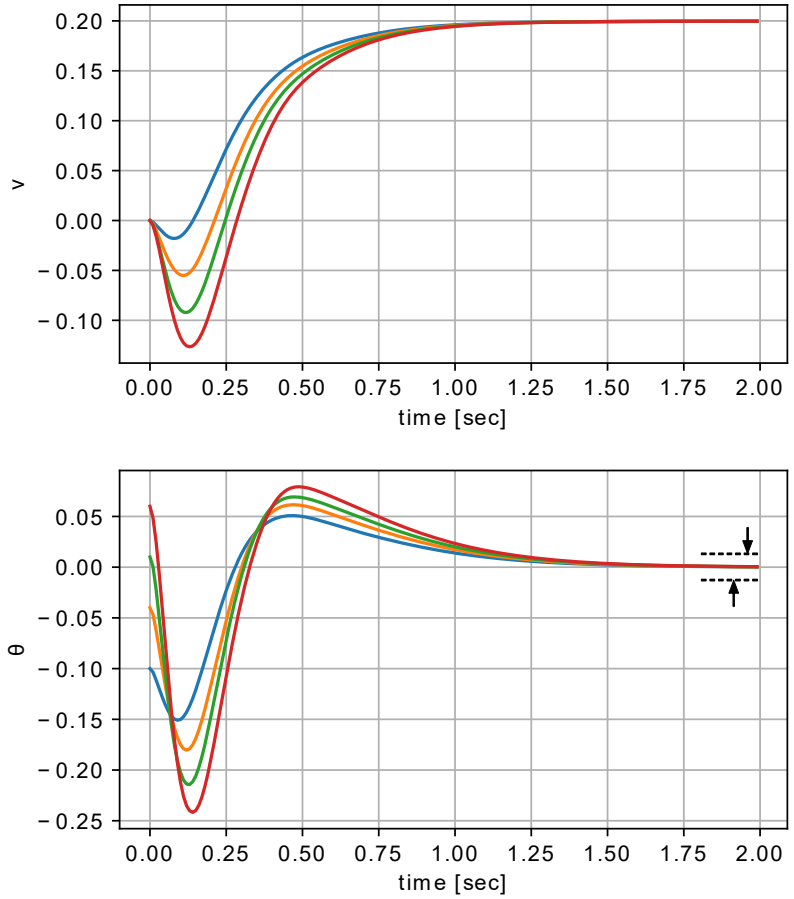


Figure 3.2: Step responses of the cart inverted pendulum with different values of μ_p, l_p and θ and nonquantized data. An error band for θ is marked with dashed lines and arrows.

of pendulum with respect to vertical axis. The initial deviation from vertical position is between -0.1 and 0.1 rad while the pendulum is placed at $\nu = 0$. According to [89], for a given bar length and mass, steady state equations governing the plant are given by

$$\dot{\mathbf{x}}_{\mathrm{LQR}}^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{\mu_p^2 g_{\mathrm{G}} l_p^2}{c} & \frac{-b(I+\mu_p l_p^2)}{c} & 0 \\ 0 & \frac{\mu_p g_{\mathrm{G}} l_p (\mu_c+\mu_p)}{c} & \frac{-\mu_p l_p b}{c} & 0 \end{bmatrix} \mathbf{x}_{\mathrm{LQR}}^{\mathrm{T}} + \begin{bmatrix} 0 \\ 0 \\ \frac{I+\mu_p l_p^2}{c} \\ \frac{\mu_p l_p}{c} \end{bmatrix} f_c, \tag{3.1}$$

where $\mathbf{x}_{\mathrm{LQR}} = [\nu, \theta, \dot{\nu}, \dot{\theta}]$, $\dot{\mathbf{x}}_{\mathrm{LQR}}$ is its derivative with respect to time. ν stands for position of the cart. $c = (\mu_c + \mu_p)I + \mu_p \mu_c l_p^2$ with μ_c, μ_p and l_p being the cart mass, pendulum mass and length to pendulum center of mass, respectively. $I = \mu_p l_p^2/3$ stands for the moment of inertia for bar mass. $g_{\mathrm{G}} = 9.8$ and $b = 0.1$ (N/m/sec) are assumed as standard gravity and coefficient of friction for the cart. Finally, f_c is the force applied to the cart in horizontal direction, determined by the controller.

To calculate the optimal force, LQR controller with precompensation factor is used for different values of bar length and mass. The cost function of LQR is $\int \mathbf{x}_{\mathrm{LQR}}^{\mathrm{T}} \mathbf{Q} \mathbf{x}_{\mathrm{LQR}} + \mathbf{u}^{\mathrm{T}} R_{\mathrm{LQR}} \mathbf{u}$, where $\mathbf{u} = -\mathbf{K}\mathbf{x}_{\mathrm{LQR}}$ and $\mathbf{K}$ is the matrix of controller coefficients. $\mathbf{Q}$ and R_{LQR} are controller parameters to balance the relative importance of error and control effort, e.g., energy consumption. In Fig. 3.2, several step responses of the cart inverted

pendulum are depicted considering different values of μ_p, l_p and θ, while nonquantized data is used at the MLC. The data set to train the MLC is generated by LQR controllers as fully described in 3.4.1

The system performance of this problem is evaluated in terms of steady state errors. The error bands for cart position and angle of pendulum are 0.1 meters and 0.01 radians, respectively. The error band for θ is roughly shown with dashed lines and arrows in Fig. 3.2. Thus, an error is counted when the deviation from equilibrium position is outside of these intervals in the last 100 milliseconds, e.g., $|\theta_{\text{final}}| > 0.01$. The steady state error probability is a standard KPI for performance evaluation of controllers in a predefined period of time. A steady state error can occur while the system becomes stable after the aforementioned 2 sec.

3.2.3 Benchmarks

Prior to elaborating our benchmarks, it is worth mentioning that the focus of state of the art in clustering literature is currently on codebook design. In these studies, the number of clusters is decided with trial and errors such as with elbow method assuming a single terminal setup. On the other hand, sophisticated approaches purely designed for bit allocation are studied for simplified system models and violate at least one of the prerequisites mentioned in our objectives in 1.2 making them inapplicable to the problem statement at hand.

Hence, in order to compare our results with syntax based solutions, a typical MSE based approach is considered as the first benchmark. MSE is a well known and common distortion measure for determining the suitable number of clusters for quantization. It has shown sufficiently high performance and is frequently used in practice. In our first benchmark, the selected bit allocation using MSE is given by

$$\boldsymbol{\eta}^* = \underset{\boldsymbol{\eta}\in\mathcal{H}}{\operatorname{argmin}} \sum_{n=1}^{N} \sigma_n^2, \tag{3.2}$$

where $\sigma_n^2 = \mathbb{E}_{x_n}\{(x_n - \hat{x_n})^2\}$ is the MSE between nth input feature x_n and its quantized version $\hat{x}_n$ which is calculated by employing data sets. Expectation is denoted by $\mathbb{E}\{\cdot\}$.

Equal sharing is the second method that we investigate to provide a comparison baseline, since it can be considered as another practical approach to tackle the multiterminal bit allocation problem. In this case, $\eta_n = \lfloor \eta_{\text{sum}}/N \rfloor$ and $\lfloor\cdot\rfloor$ returns the greatest integer which is equal or less than its input. This choice of η_n complies with our assumption on no exchange of knowledge among sources and integer-valued η_n. Hence, η_n changes only if remainder of η_{sum}/N is zero.

3.3 The Proposed Solution

3.3.1 KLD as Relevance Based Distortion Measure

***Problem Statement* 1:** In this chapter, we opt for a distortion measure $d_{\text{rel}}(\cdot)$ which is a function of $\hat{\mathbf{x}}$ and $\mathbf{y}$ to take impact of quantization on MLU output into account and provide a best effort performance in selecting a bit allocation according to the following optimization problem and constraints.

$$\boldsymbol{\eta}^* = \underset{\boldsymbol{\eta}}{\operatorname{argmin}}\, d_{\text{rel}}(\hat{\mathbf{x}}, \mathbf{y}), \tag{3.3}$$

subject to

$$\eta_n > 0, \tag{3.4}$$

$$\sum_n \eta_n \leq \eta_{\text{sum}}. \tag{3.5}$$

To this end, we consider the distribution over MLU output and highly precise input as the reference, and the goal is to find a bit allocation such that the distribution over MLU predictions and quantized features resembles the ground truth. Therefore, KLD is selected as the cost function and 3.3-3.5 can be reformulated as follows.

$$\boldsymbol{\eta}^* = \underset{\boldsymbol{\eta}\in\mathcal{H}}{\operatorname{argmin}}\, D_{\mathrm{KL}}\Big(p_{\hat{\mathbf{X}},\mathbf{Y}}(\hat{\mathbf{x}},\mathbf{y})||q_{\hat{\mathbf{X}},\mathbf{Y}}(\hat{\mathbf{x}},\mathbf{y})\Big), \tag{3.6}$$

where $D_{\mathrm{KL}}(p_{\hat{\mathbf{X}},\mathbf{Y}}(\hat{\mathbf{x}},\mathbf{y})||q_{\hat{\mathbf{X}},\mathbf{Y}}(\hat{\mathbf{x}},\mathbf{y}))$ is the KLD or relative entropy measuring dissimilarity between two distributions. $\mathcal{H}$ contains all the bit allocations satisfying our constraints $\sum_{i=1}^{N}\eta_n \leq \eta_{\mathrm{sum}}$, where $\eta_n > 0$ is an integer-valued number. To solve this optimization problem, we estimate the two distributions empirically as explained in the following.

The quality and accuracy of the solution provided by (3.6) is highly dependent on KLD approximation accuracy. Here, we estimate p and q using non-parametric methods, histogram with smoothing and k-NN. The histogram estimator is a simple approach with the drawback of having many bins with zero samples. In addition, the number of its required bins increases exponentially with data dimension. We also consider k-NN estimator to investigate the effect of KLD approximation accuracy on system performance. k-NN has been used for mixed continuous-discrete setups, and a high accuracy for strongly correlated data is not guaranteed for this estimator [90]. Here, we employ it for studying impact of coarsely estimated distributions.

Let $\mathcal{T}_n$, $n = 1, 2$, be data sets each containing J samples $\{\mathbf{z}_j; j = 1, \cdots, J\}$ drawn from distributions p and q, respectively. The k-NN estimation of p is

$$\hat{p}(\mathbf{z}_j) = \frac{k}{J} \times \frac{1}{v(\mathbf{z}_j)};\ \mathbf{z}_j \in \mathcal{T}_1, \tag{3.7}$$

where $v(\mathbf{z}_j) = \frac{\pi^{d/2}}{\Gamma(d/2+1)} \times \frac{1}{R_p(\mathbf{z}_j)^{-d}}$ is the volume of a d-dimensional ball with radius $R_p(\mathbf{z}_j)$, $\Gamma(\cdot)$ is the gamma function and $R_p(\mathbf{z}_j)$ stands for the euclidean distance between $\mathbf{z}_j$ and its kth neighbor in $\mathcal{T}_1$. The kth neighbor of $\mathbf{z}_j$ is the kth sample in the list of sorted samples of $\mathcal{T}_1$ from minimum to maximum euclidean distance regarding $\mathbf{z}_j$. And, d is sum of MLU input and output dimensions. Similarly, an estimate of q can be calculated, where $R_q(\mathbf{z}_j)$ is the euclidean distance between $\mathbf{z}_j \in \mathcal{T}_1$ and its kth neighbor in $\mathcal{T}_2$. Therefore, the plugin estimator for KLD of (3.6) becomes

$$D_{\mathrm{KL}}(p||q) \approx \mathbb{E}_{\mathbf{z}}\Big\{ \log\Big(\frac{\hat{p}(\mathbf{z}_j)}{\hat{q}(\mathbf{z}_j)}\Big)\Big\}. \tag{3.8}$$

A well-known difficulty with computing KLD is that to get a finite value, the support set of true distribution should be contained in support set of estimated distribution. While this is reasonable in some applications, it is an extreme condition for learning problems, particularly since distributions are only approximated with limited number of samples. Therefore, data smoothing can be used to overcome the problem. To deal with this situation, the width of histogram bins are selected to be larger than that provided by quantization. Thus, for each sample in support set of p, we assume the existence of at least one sample when approximating q. In this case, instead of $\hat{q}(\mathbf{z}_j) = \frac{n_{\mathrm{bin}}}{J}$, where n_{bin} is the number of samples in histogram bin of $\mathbf{z}_j$, we have

$$\hat{q}(\mathbf{z}_j) = \frac{n_{\mathrm{bin}} + \alpha}{J + \mu_{\mathrm{bin}}}, \tag{3.9}$$

where μ_{bin} is the number of bins in support of p with zero samples from $\mathcal{T}_2$. For $n_{\mathrm{bin}} = 0$, $\alpha = 1$ and otherwise, $\alpha = 0$. It is worth mentioning that in this bit allocation setup,

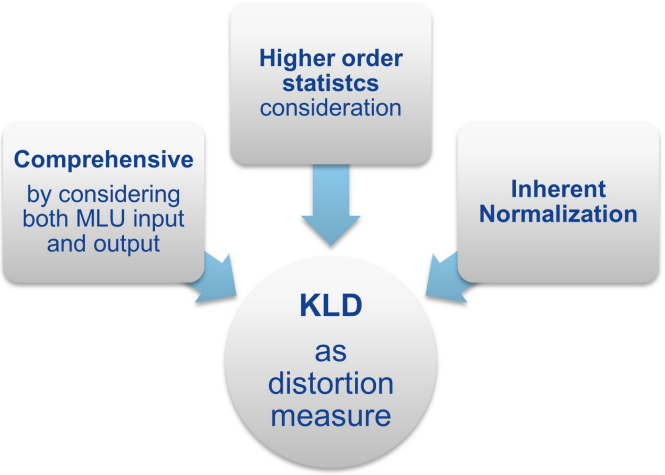


Figure 3.3: A summary of the reasons for selection of KLD as the relevance based distortion measure.

the relative KLD values and their order are decisive, not absolute values. The introduced approach in (3.9) is inspired by additive smoothing for language modeling as described in [91] with a slight modification that makes it applicable to our KLD estimation problem. In additive smoothing, α is always set to one and vocabulary size replaces μ_{bin} in denominator of (3.9). Here, in order to keep the estimation as close as possible to the evidence provided by observed data, α is set to one in problematic cases, i.e., bins with zero samples from q. The normalization factor μ_{bin} is accordingly adjusted to account for the pseudosamples counted by α impacting KLD estimation.

The feasible set of this problem is non-convex due to the integer-valued bit allocation assumption, however, it contains a limited number of members. Thus, for focusing on impact of KLD approach and its approximation on MLU output, estimations of (3.8) are substituted in (3.6) for members of $\mathcal{H}$ and a brute-force search finds the optimal solution. In Chapter 5, we study a system in which exhaustive search is infeasible. Thus, a heuristic approach is employed to tackle the problem.

In a high dimensional space, large number of required samples for meaningful estimations with a simple histogram can be restrictive. k-NN method can circumvent this problem. The required k-NN computations are theoretically expensive for a large data set. However, the calculations for both KLD approximations and solving (3.6) are performed only once and offline. Once the bit allocations are determined for different bandwidth constraints, one of them is picked for quantization according to the available bandwidth. Therefore, dealing with these computations is feasible in practice without affecting applicability of the proposed approach.

It is also worth noting that estimating distributions using k-NN is much simpler in a classification problem. The case study of Chapter 4 employs a classifier, where this approximation is discussed in details. Moreover, when dealing with the case study in Chapter 5, we suggest a heuristic approach to shrink the search space considering requirements of the mobility problem under study.

3.3.2 The Reasons for Selection of KLD

Selection of KLD as the relevance based distortion measure is the result of our attempt to provide a solution which is in theory as close as possible to the optimal one. A summary of the reasons for using KLD is presented in Fig. 3.3. The introduced distortion considers pairs of input and output which makes it a more comprehensive measure when compared

to a choice only taking $\mathbf{y}$ into account. For instance, let $\{(x_1, y_1 = 0), (x_2, y_2 = 1)\}$ which are the ground truth, and imply $p(y = 0) = 1/2$ and $p(y = 1) = 1/2$. If after quantization, we get $y_1 = 1, y_2 = 0$, the estimated distribution over classifier output remains the same, however, we got the wrong class labels for both samples. This simple example is a worst case scenario that may not occur in practice. However, it shows that accounting for all contributing factors, here tuples of MLU input and output, can be beneficial in terms of achieving a better bit allocation.

Moreover, divergence accounts for a full description of data statistics represented by distributions and is not limited to typically used second order statistics. Hence, it is expected that it delivers superior outcome in non-Gaussian and highly non-linear scenarios. Along with employing divergence, the proposed empirical estimation of KLD guarantees an aggregate consideration of MLU input and output variables and their underlying mutual dependencies.

Utilization of a measure based on distributions incorporates an inherent normalization which plays an important role in avoiding destructive impact of outliers. For example, if distortion measure deals with $\mathbf{x}$ and $\mathbf{y}$ directly, a large difference between MLU outputs in presence of high and low resolution quantizations for a single given sample can result in discarding a generally proper bit allocation. This problem is resolved when working with distributions around a given point in divergence.

It is also worth mentioning that, in many use cases, KPIs are parameters rather than MLU output, e.g., number of failures for an inverted pendulum. Taking these KPIs into account while measuring distortion is however infeasible in most scenarios because of inaccessibility. Therefore, we opt for MLU output instead of a direct consideration of KPIs.

3.4 Simulation Setup

3.4.1 Training the MLC

As the MLC, we train a fully-connected shallow NN with 70 neurons. The input features for MLC are mass and length of the bar pendulum, position ν, velocity $\dot{\nu}$, angular position θ and angular velocity $\dot{\theta}$, implying an input layer dimension of 6. Hence, $\mathbf{x} = [\mu_p, l_p, \nu, \theta, \dot{\nu}, \dot{\theta}]$, where values of μ_p and l_p can be selected from the ranges 0.1 to 2 kg and 20 to 50 cm, respectively. In addition, the output of MLC y is the horizontal force applied to the cart which is shown as f_c in (3.1). The NN is trained with a data set generated using LQR controllers for different random values of bar mass and length, with the following parameters: $\mu_c = 0.5\,\text{kg}$, $R_{\text{LQR}} = 0.1$ and $\mathbf{Q}$ is a 4×4 matrix with zero entries except for the first and third diagonal elements being 5000 and 100, respectively. The LQR parameters are selected based on a trial and error procedure as elaborated in [92]. The sampling time is 0.01 seconds. The training and test set contain 600 and 200 sequences, each of length 200, respectively. Validation ratio is $\frac{1}{3}$.

Here, we deal with a regression problem. Sigmoid and linear activation functions are used in hidden and output layer, respectively. MSE is the loss function for training and NN weights are initialized with Xavier uniform initializer. Batch gradient descent with batch size of 1000 is the search algorithm. Furthermore, the learning rate is 0.01 with no decay factor. Stop condition is getting no improvement in validation loss for 50 epochs which occurred after 641 epochs. The final MSE achieved on the test set is ≈ 0.23. In Fig 3.4, several step responses of both the LQR controller and MLC are depicted for different values of μ_p, l_p and initial θ considering nonquantized data. These examples show that the difference between step responses of our trained MLC and the optimal controller is insignificant. Hence, the MLC is capable of replacing this controller in our case study.

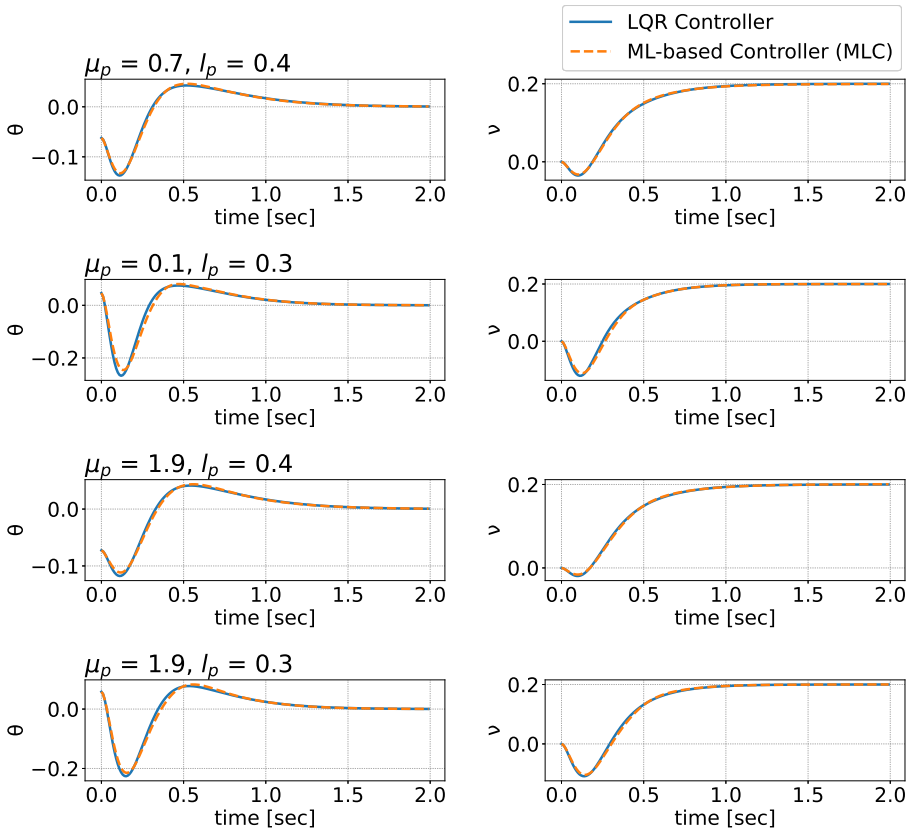


Figure 3.4: A comparison between step responses of the MLC and LQR controller for different values of μ_p, l_p and initial θ.

Note that while LQR parameters depend on pendulum mass and length, a single MLC is trained and works for different values of μ_p and l_p belonging to ranges 0.1 to 2 kg and 20 to 50 cm, respectively.

3.4.2 KLD Estimation and Bit Allocation

We use the MLC to generate data sets for estimation of KLD. For the uniform quantization, minimum and maximum values of each random variable is taken from $\mathcal{T}_1$. Since μ_p and l_p are not expected to change frequently, we assume that their values are transmitted with 10 bits for each feature when needed. Members of $\mathcal{H}$ are selected to satisfy $3 \leq \eta_n \leq 9$ and $\sum_n \eta_n = \eta_{\text{sum}}$, where we have $\eta_{\text{sum}} - 20$ bits to quantize the last four attributes of vector $\mathbf{x}$ described in 3.4.1. This interval choice both limits the search space and is sufficiently large considering the range of random variables in this problem. For estimating p and q, 40000 samples and the typical value of $k = \sqrt{J} = 200$ are used.

As explained in section 3.2, we assume $p_{\mathbf{X}}(\mathbf{x})$ is fixed which is the case for many non-adaptive learning problems. Thus, data set $\mathcal{T}_2$ can be constructed directly from $\mathcal{T}_1$ by simply quantizing its input samples for a given bit allocation and feeding them into the MLU to compute corresponding outputs. This procedure reduces computations significantly, because the alternative is to run simulations for pendulum environment to build a data set for each bit allocation.

On the other hand, for the specific problem of inverted pendulum, very low quality quantization results in force decisions with large distance from the true ones. And after feeding back these force decisions to the plant, $p_{\mathbf{X}}(\mathbf{x})$ starts to diverge from the assumed distribution and consequently, $\mathcal{T}_1$ must be updated. In order to avoid this difficulty, distribution

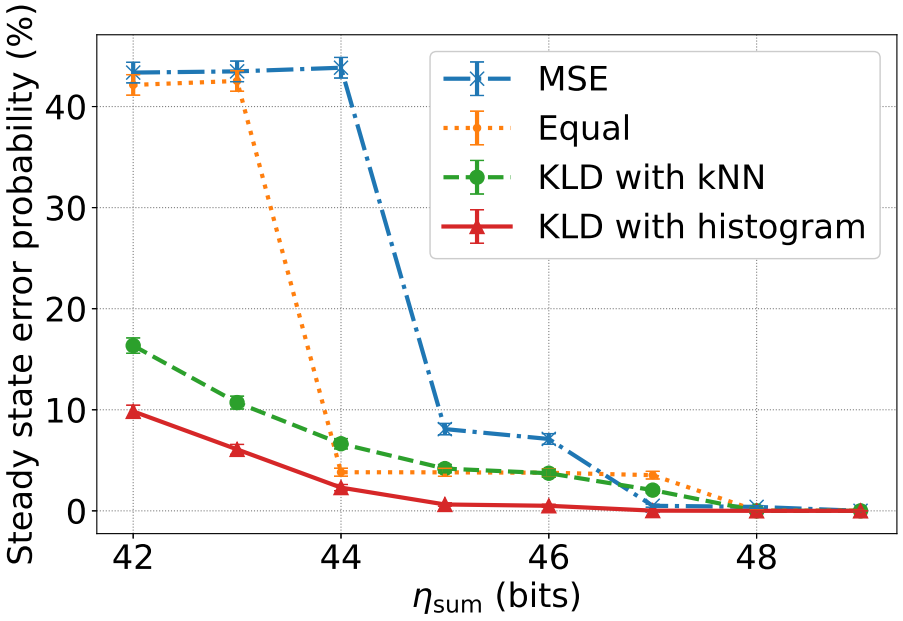


Figure 3.5: Steady state error probability in percentage vs. η_{sum} the total number of quantization bits used in a symbol interval.

on $\mathbf{x}$ is estimated for different sum rate constraints and bit allocations. For this purpose, histogram with smoothing is used as explained in Section 3.3. Then, KLD between distribution of these allocations and the true distribution $p_{\mathbf{X}}(\mathbf{x})$ is calculated. These KLD estimations show a small value for $\eta_{\text{sum}} \geq 42$. Therefore, it is a valid assumption that $p_{\mathbf{X}}(\mathbf{x})$ is almost fixed for sum rate constraints larger than 42 bits.

3.5 Numerical Results

In this section, the step response of cart inverted pendulum is monitored for 10000 iterations while each iteration simulates a period of 2 sec. The steady state error probability P_{e} with confidence intervals of 98% derived by Wald method vs. total number of quantization bits used in a symbol interval η_{sum} is depicted in Fig. 3.5. Simulations are performed for the proposed KLD based approach with histogram and k-NN estimation, equal bit sharing and MSE based bit allocation of (3.2). The proposed method with histogram estimation outperforms other techniques for all sum rate constraints, and indicates a gain of 2 bits in achieving $P_{\text{e}} < 0.001$ at 47 bits with respect to equal sharing and MSE methods. It should be noted that this single inverted pendulum scenario is a sandbox, and the gains and rate of the communication scheme in a real environment with signal overheads and more devices increases rapidly. Particularly, the KLD with histogram picks a significantly better bit allocation for low sum rate values. For instance, if 42 bits can be assigned for the system, error probability for both equal sharing and MSE are larger than 40%. This number can be reduced to $\approx 10\%$ implying a reduction of more than 30% in failures using the KLD. This huge gain is a result of taking ML output into consideration.

In order to study the distribution of quantization noise and its pattern when a low error probability is achieved, consider the KLD approach with histogram at 46 bits and $P_{\text{e}} \approx 0.005$. With this constraint, the number of allocated bits for features of $\mathbf{x}$ are $[10, 10, 6, 6, 6, 8]$. Assuming that quantization error variance is defined as $\sigma_n^2 = \mathbb{E}\{(x_n - \hat{x}_n)^2\}$ for each feature, we have $\sigma_3^2 \approx \sigma_4^2$ of order of 10^{-6}. However, for $\dot{\nu}$ and $\dot{\theta}$, quantization variances are $\sigma_5^2 \approx 0.0003$ and $\sigma_6^2 \approx 0.0001$ which are almost 100 times larger than that of ν and θ. This pattern of having lower quantization noise for θ and ν remains the

same for bit allocations which turn out to provide low probabilities of error. Therefore, it can be concluded that these features have a higher relevancy or importance for the MLU.

For $\eta_{\text{sum}} \leq 46$, rate allocations selected by MSE criterion result in the worst steady state error performance among all the methods under study. This performance gap is larger at lower sum rate values, e.g., a loss of 37.4% and 32.7% at $\eta_{\text{sum}} = 43$ regarding the KLD with histogram and k-NN, respectively. Furthermore, MSE based technique shows a huge improvement from $\eta_{\text{sum}} = 44$ to 45 bits. The reason lies behind the range from which input features take their values, and the fact that MSE is calculated independently of MLC output. In this setup, $\dot{\nu}$ and $\dot{\theta}$ values are picked from intervals which are almost 9 and 21 times bigger than those of θ and ν. Therefore at the beginning, the syntax based MSE allocates more bits for $\dot{\theta}$ and $\dot{\nu}$, although high accuracy on these less relevant random variables does not improve the force decision. The first significant enhancement only occurs when σ_5^2 and σ_6^2 are small enough, so, extra bits are used for θ. Thus, a change from 4 to 5 in number of bits for θ when $\eta_{\text{sum}} = 44$ becomes 45 bits leads to a decrease of $\approx 35.7\%$ in probability of error. The second decrease is also a consequence of allocating 5 bits instead of 4 bits for ν when moving from $\eta_{\text{sum}} = 46$ to 47.

Equal sharing outperforms the MSE results given that $\eta_{\text{sum}} \leq 46$, e.g., $P_{\text{e}} \approx 42.5\%$ instead of $\approx 43.5\%$ for 43 bits. As stated before, the bit allocation provided by this method remains the same, unless sum rate is divisible by 4 which explains improvements at 44 and 48 bits. This method provides better results than KLD with k-NN for the constraint of 44 which can be interpreted as a lucky situation for this approach. With 44 bits, equal sharing allocates 6 bits for each of $\nu, \theta, \dot{\nu}$ and $\dot{\theta}$. This indicates less quantization noise for more relevant random variables θ and ν which only happens because of their smaller intervals in this specific pendulum scenario. On the other hand, KLD with k-NN is not capable of following distributions accurately and settles for a worse bit allocation with $\approx 3\%$ more failures than that of equal sharing.

As expected, changing histogram estimator to k-NN degrades the performance since k-NN is not capable of providing a highly accurate estimation of KLD, particularly for the system under investigation with highly correlated variables. However, it still offers less number of errors compared with the MSE approach for $\eta_{\text{sum}} \leq 46$. For the constraint with 42 bits, it achieves a gain of 27% and 25.8% in comparison to MSE and equal bit sharing methods but the selected bit allocation causes $\approx 6.5\%$ higher error probability with respect to the KLD with histogram estimator. KLD with k-NN also provides a better or equivalent performance regarding equal sharing for most points, except for $\eta_{\text{sum}} = 44$ which was discussed.

As shown by the numerical results, using the relevance based KLD approach with histogram is more beneficial in terms of fulfilling the requirements imposed by ML functionalities in a bandwidth limited system. In operation points with high probability of stability, the quantization noise on angle and position are much smaller than other features which indicates they have a higher level of relevance for the MLU. This knowledge can be used in case of having limited resources for providing a best-effort performance.

In addition to simulations presented in this chapter, we provide numerical results for a different setup of the inverted pendulum problem and a toy data set in Appendix A. The synthetic data set is introduced in scikit-learn to perform classification tasks. All the considered simulations show significant gains when using the proposed KLD method, demonstrating its power and benefits when used in rate limited systems. This is also theoretically expected because the conventional methods such as MSE do not take the final MLU decision into consideration.

3.6 Summary and Conclusion

Since intelligent elements governed by ML become an integrated part of communications networks, we introduced a KLD based bit allocation for quantization of multiple correlated sources delivering input of a MLU. Different KLD estimation methods were studied and simulation results show that the proposed method provides promising gains in system performance of a cart inverted pendulum problem, particularly for more restricted bandwidth constraints.

These observations motivate the shift from syntax to relevance based designs which operate in accordance with MLU requirements considering rate and resource limitations. It should be noted that achieved gains using KLD approach are use case dependent, and they highly rely on KLD estimation accuracy. Therefore, an analysis to find parameters impacting such gain is presented in the next chapter. More importantly, to account for impact of high dimensional MLU inputs, an adjustment to the proposed KLD based bit allocation is made for such scenarios.

4. Curse of Dimensionality and Divergence Based Bit Allocation

4.1 Overview

In this chapter, we propose a similar but modified divergence based distortion measure, when compared to (3.6), for problems with high dimensional Machine Learning Based Unit (MLU) input. The advantage of using this measure and its difference from that of the last chapter are elaborated in Section 4.3. Moreover, this chapter provides insights into benefits of employing the divergence based approach instead of its conventional alternatives using Sum of Squared Errors (SSE) and equal sharing. To this end, indoor environment classification with real measurements is studied. As shown in Fig. 4.1, this study covers both scalar and vector quantization, and different hypotheses: Neural Network (NN), decision tree, 1-nearest neighbor and Support Vector Machine (SVM). Furthermore, MLU performance and its robustness to Packet Drop (PD) with selected bit allocations of Kullback-Leibler Divergence (KLD) and SSE is evaluated. Simulations show that classification accuracy achieved by the KLD approach is significantly higher or in a few particular cases similar to the conventional methods, and it does not affect robustness of the MLU in presence of packet loss. The state-of-the-art solutions related to this study are reviewed in 3.1.1.

4.1.1 Main Contributions of the Chapter

In Chapter 3, we investigated how accuracy of KLD estimation affects bit selection and hence, system performance. This chapter covers a comprehensive study of many factors influencing gains achieved by the KLD approach and justifies its applicability to a wide range of problems. Here unlike the regression problem of the last chapter using a synthetic data set with low dimensional input, the 2.4 GHz indoor environment classification as described in [42, 43] with real measurements is explored. The data set is available in [93].

The proposed KLD method provides the highest classification accuracy in all simulations while delivering significant gains of up to 19%. The numerical results with vector quantization demonstrate that the KLD approach is capable of enhancing MLU performance even in combination with an effective clustering in terms of bit utilization.

Additionally, MLU input data must be transmitted in a limited time or otherwise, become obsolete in many networks. In these situations, MLUs can use several techniques to deal with missing values and reduce performance loss. Hence, we address the question of whether the selected KLD bit allocations affect these efforts and robustness of a given

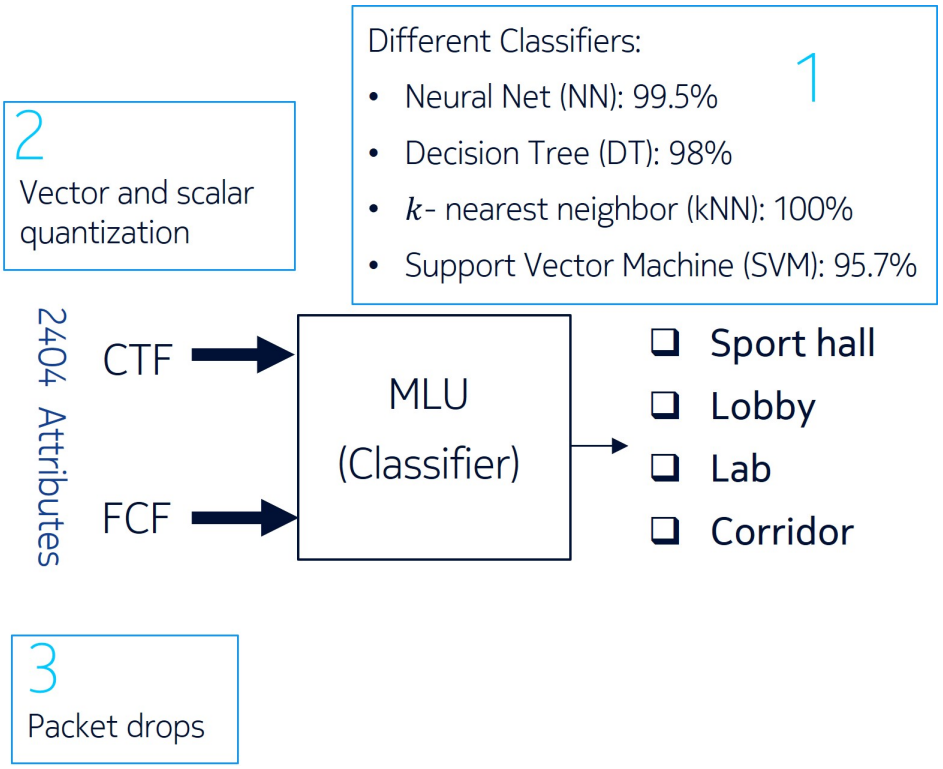


Figure 4.1: An overview of the three main subjects covered in this chapter.

MLU when PD occurs. Simulations performed with vector quantization and NN illustrate that KLD bit allocations still outperform SSE selections. The main contributions of this chapter are published in [94] and are as follows.

1. The system model of Chapter 3 is extended to account for sources performing vector quantization on data.

2. The KLD based distortion measure of (3.6) is modified for problems with high dimensional MLU input.

3. The study of classifiers employing four different hypotheses provides insights about dependence of gain levels on the given MLU and nature of the problem. Classification accuracy is the Key Performance Indicator (KPI) in this system.

4. Both scalar and vector quantization are surveyed to analyze their impact on achieved gains by the proposed KLD approach comparing with two conventional methods.

5. It is shown that bit allocations of the KLD approach have a higher robustness in case of PD occurrence. In other words, if the data is quantized by the KLD selections, least classification degradation occurs for all simulations with different PD rates.

This chapter is organized as follows. The system model is described in Section 4.2, where the problem formulation, indoor environment data set, kmeans as the employed vector quantization and benchmarks are elaborated. In Section 4.5, the KLD based bit allocation for high dimensional scenarios and a KLD estimation for classification problems are introduced. The simulation setup is discussed in Section 4.4, and numerical results are presented. Finally, conclusions are drawn in Section 4.6.

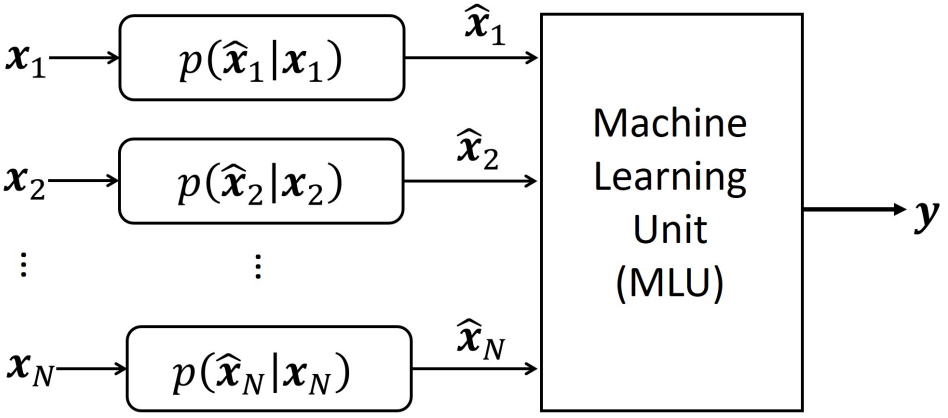


Figure 4.2: Block diagram of the system model.

4.2 System Model

4.2.1 General Description

In this section, we only point out differences between the extended system model considered in this chapter and that of chapter 3. The multiple access channel scenario under study is depicted in Fig. 4.2, where N memoryless stationary sources quantize and transmit input attributes $\mathbf{x} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]$ to a MLU for predicting $\mathbf{y}$. Here, each source can perform either scalar or vector quantization on its data $\mathbf{x}_n, n = 1, \cdots, N$. The input attributes of $\mathbf{x}$ can be highly correlated and have an arbitrary joint probability density function $p_{\mathbf{X}}(\mathbf{x})$ with $\mathbf{x} \in \mathcal{X}_1^{d_1} \times \cdots \mathcal{X}_N^{d_N}$, where $d_n, n = 1, \cdots, N$ is the dimension of nth source vector. Vector quantization on data of nth source is shown with $p_{\hat{\mathbf{X}}_n|\mathbf{X}_n}(\hat{\mathbf{x}}_n|\mathbf{x}_n)$ and it is performed with η_n bits.

The rest of system model characteristics are described in 3.2. Once again, we assume training is done with high-precision data. Afterwards, the MLU remains unchanged while executing tasks online as an inference unit in network with $\hat{\mathbf{x}}$. Therefore, a bit allocation providing sufficiently accurate input components for the given MLU needs to be selected considering input relevancy. For high dimensional $\mathbf{x}$, this selection is performed according to the solution provided in Section 4.3.

4.2.2 Case Study 2: 2.4 GHz Indoor Environment Classification and its KPI

The 2.4 GHz indoor channel measurement data set provides real measurements of scattering parameters, S_{21} parameter, for four different indoor environments: lobby, laboratory, corridor and sport hall. The layout of the floor plan for the first three classes is presented in Fig. 4.3. The sport hall is an open space area. The channel measurements are carried out around 2.4 GHz covering 100 MHz bandwidth. They include 10 sweeps, where each sweep contains 601 frequency points being 0.167 MHz apart. In [42, 43], the data set is described in details and used to design classifiers with different hypotheses. [42] demonstrates that best MLU performance is achieved if Channel Transfer Function (CTF) and Frequency Coherence Function (FCF) are fed to the MLU as its input components.

Therefore, CTF and FCF measurements are assumed to be $\mathbf{x}_1$ and $\mathbf{x}_2$ provided from two sources so that this problem complies with the multiterminal scenario under study with $N = 2$[1]. Furthermore, in case of PD for data of one source, the MLU gets half of its input

[1]Note that working on aspects like feature selection and choosing the best MLU is not our intention. In practice, MLU design is not always rigorous, and if provided from a different vendor cannot be modified. So, the KLD approach is used to obtain best effort performance even in imperfect systems.

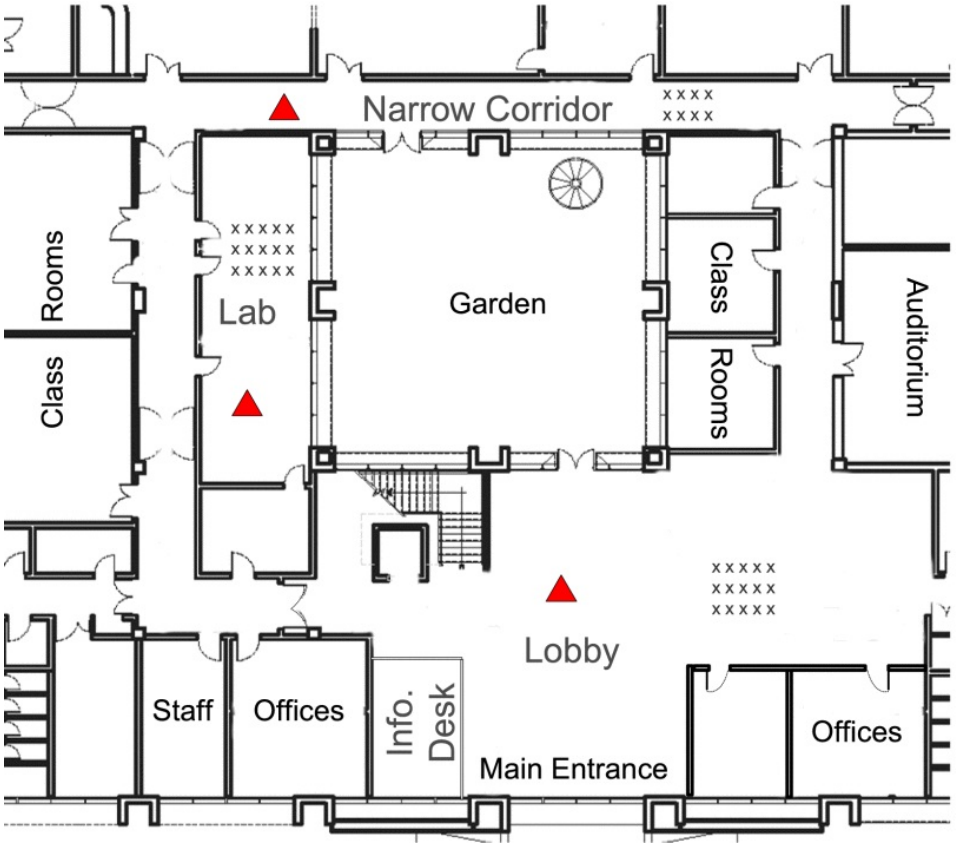


Figure 4.3: The floor plan of lobby, corridor and laboratory channel measurements.

data which gives it a chance to still predict the correct output. Note that MLUs need values at all their input ports. A few common ways to deal with missing values are to feed the MLU with a random codeword, most frequent or an average in case of having missing attributes.

The S_{21} parameters present the CTF $H(f)$. The complex autocorrelation of CTF is FCF shown by $R(f)$ and given by

$$R(f) = \int_{-\infty}^{\infty} H(\hat{f})\bar{H}(\hat{f}+f)d\hat{f}, \tag{4.1}$$

where $\bar{H}(f)$ is the complex conjugate of CTF at frequency f. This data set has 1960 samples for each environment. Each sample contains S_{21} measurements at 601 frequency points. The real and imaginary part of these parameters are treated as separate variables. As a result, CTF and FCF input vectors have 2404 elements in total, which is the number of input attributes for the MLU.

For both vector and scalar quantization, each source performs compression of its data separately, i.e., CTF is quantized and transmitted independently of FCF. The MLU output $\mathbf{y}$ is a vector with four variables representing the four environments, the variable referring to the decided class becomes one and others are zero. In order to quantify system performance, the KPI of this case study is classification accuracy.

4.2.3 Benchmarks and Vector Quantization with kmeans

kmeans is a widely used vector quantization method partitioning a set of observations into a given number of clusters, here 2^{η_n} for source n, while minimizing within-cluster sum of squared error. Intuitively, a cluster is a group of data points selected from a set of

observations. A feasible clustering of $\mathbf{x}_n \in \mathcal{T}_{\text{train}}$ at source n with η_n bits is shown as $\mathbf{S}_n = \{S_{1,n}, \cdots, S_{2^{\eta_n},n}\}$. Each cluster i at source n is denoted as $S_{i,n}$ and represented with a d_n-dimensional codeword $\boldsymbol{\mu}_{i,n}$. At source n and for a given η_n, kmeans finds both clusters and codewords according to

$$\left(\mathbf{S}_n^*, \{\boldsymbol{\mu}_{i,n}^*, i = 1, \cdots, 2^{\eta_n}\}\right) = \underset{\mathbf{S}_n, \boldsymbol{\mu}_{i,n}}{\operatorname{argmin}} J_n(\eta_n), \tag{4.2}$$

where $\mathbf{S}_n^*$ and $\{\boldsymbol{\mu}_{i,n}^*, i = 1, \cdots, 2^{\eta_n}\}$ are the final clustering and set of codewords for nth source provided by kmeans. In addition, $J_n(\eta_n)$ is objective of kmeans and is defined as follows

$$J_n(\eta_n) = \sum_{i=1}^{2^{\eta_n}} \sum_{\mathbf{x}_n \in S_{i,n}} ||\mathbf{x}_n - \boldsymbol{\mu}_{i,n}||^2; \mathbf{x}_n \in \mathcal{T}_{\text{train}}, \tag{4.3}$$

where $||\mathbf{x}_n - \boldsymbol{\mu}_{i,n}||^2$ returns the squared euclidean distance between $\mathbf{x}_n$ and $\boldsymbol{\mu}_{i,n}$. Minimizing $J_n(\eta_n)$ targets having inter-cluster point distances smaller than the distances to points outside of the cluster.

An iterative approach is employed to solve (4.2). In a standard procedure, given a set of initialized $\boldsymbol{\mu}_{i,n}$, each sample $\mathbf{x}_n$ is assigned to the cluster with nearest $\boldsymbol{\mu}_{i,n}$, then $\boldsymbol{\mu}_{i,n}$ is updated based on

$$\boldsymbol{\mu}_{i,n} = \mathbb{E}_{\mathbf{x}_n \in S_{i,n}}\{\mathbf{x}_n\}. \tag{4.4}$$

The assignment and codeword update are repeated until the algorithm converges, i.e., assignment to clusters remains unchanged. Solving (4.2) is NP hard, however, there are a variety of heuristic algorithms capable of reaching a local minimum fast. Elkan algorithm is used in our simulations for this purpose. In order to quantize a sequence $\mathbf{x}_n \notin \mathcal{T}_{\text{train}}$ using kmeans, we have

$$\hat{\mathbf{x}}_n = \underset{i}{\operatorname{argmin}} ||\mathbf{x}_n - \boldsymbol{\mu}_{i,n}^*||^2. \tag{4.5}$$

The first benchmark for comparisons is a conventional syntax based approach, inspired by kmeans objective as follows

$$\boldsymbol{\eta}^* = \underset{\boldsymbol{\eta} \in \mathcal{H}}{\operatorname{argmin}} \sum_{n=1}^{N} J_n^*(\eta_n), \tag{4.6}$$

where $J_n^*(\eta_n) = \sum_{i=1}^{2^{\eta_n}} \sum_{\mathbf{x}_n \in S_{i,n}^*} ||\mathbf{x}_n - \boldsymbol{\mu}_{i,n}^*||^2; \mathbf{x}_n \in \mathcal{T}_{\text{train}}$ is the within-cluster sum of squared error for nth source that kmeans achieves after convergence of clustering process with η_n bits. η_n is the nth element of a given bit allocation $\boldsymbol{\eta}$. We refer to this method as SSE which stands for sum of squared errors.

In case of uniform scalar quantization, codewords are determined by dividing the interval of each variable equally. So, (4.6) is reformulated to $\boldsymbol{\eta}^* = \operatorname{argmin}_{\boldsymbol{\eta} \in \mathcal{H}} \sum_n ||\mathbf{x}_n - \hat{\mathbf{x}}_n||; \mathbf{x}_n \in \mathcal{T}_{\text{train}}$. The study of scalar quantization is considered here, since it remains a practical solution, e.g., if a large enough data set is not accessible for vector quantization. In such cases, vector quantization cannot come up with reliable representations of data.

Equal sharing is the second baseline for comparison, where $\eta_n = \lfloor \eta_{\text{sum}}/N \rfloor$. The operator $\lfloor \cdot \rfloor$ returns the greatest integer which is equal or less than its input to provide integer-valued η_n. Clearly, η_n changes only when remainder of η_{sum}/N is zero.

4.3 The Proposed Solution

In this section, we propose the following KLD based distortion measure to replace $d_{\text{rel}}(\hat{\mathbf{x}}, \mathbf{y})$ in (3.3) for problems with high dimensional MLU input such as the case study of this chapter with 2404 input variables.

$$\boldsymbol{\eta}^* = \underset{\boldsymbol{\eta}\in\mathcal{H}}{\operatorname{argmin}}\, D_{\text{KL}}\Big(p_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})||q_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})\Big), \tag{4.7}$$

with

$$D_{\text{KL}}\Big(p_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})||q_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})\Big) = \mathbb{E}_{(\hat{\mathbf{x}},\mathbf{y})}\Big\{\log\Big(\frac{p_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})}{q_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})}\Big)\Big\}, \tag{4.8}$$

where $D_{\text{KL}}\Big(p_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})||q_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})\Big)$ is the conditional KLD measuring dissimilarities between two conditional distributions, $p_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})$ and $q_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})$. This distortion measure is relevance based, since it takes MLU output into account. Similar to the conditions of (3.6), $\mathcal{H}$ contains all the bit allocations satisfying our constraints $\sum_{i=1}^{N}\eta_n \leq \eta_{\text{sum}}$, where $\eta_n > 0$ is an integer-valued number.

The conditional KLD of (4.7) and the KLD of (3.6) are related to each other as stated below.

$$D_{\text{KL}}\Big(p_{\hat{\mathbf{X}},\mathbf{Y}}(\hat{\mathbf{x}},\mathbf{y})||q_{\hat{\mathbf{X}},\mathbf{Y}}(\hat{\mathbf{x}},\mathbf{y})\Big) = D_{\text{KL}}\Big(p_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})||q_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})\Big) + D_{\text{KL}}\big(p_{\hat{\mathbf{X}}}(\hat{\mathbf{x}})||q_{\hat{\mathbf{X}}}(\hat{\mathbf{x}})\big), \tag{4.9}$$

where $D_{\text{KL}}\big(p_{\hat{\mathbf{X}}}(\hat{\mathbf{x}})||q_{\hat{\mathbf{X}}}(\hat{\mathbf{x}})\big)$ quantifies dissimilarities between distributions on $\hat{\mathbf{x}}$ with different quantization resolutions. Hence, it represents a syntax based distortion measure disregarding impact of MLU output $\mathbf{y}$. Here, we show that in high dimensional scenarios, there is a high chance that this term dominates the conditional KLD and leads to loosing advantages of using a relevance based approach. Let $p_{\hat{\mathbf{X}}}(\mathbf{x}_j)$ be estimated by

$$\hat{p}_{\hat{\mathbf{X}}}(\mathbf{x}_j) = \frac{k}{|\mathcal{T}_1| \times v(\mathbf{x}_j)}; \mathbf{x}_j \in \mathcal{T}_1, \tag{4.10}$$

where $|\mathcal{T}_1|$ is the number of training samples and $v(\mathbf{x}_j)$ is volume of a sphere with its center at $\mathbf{x}_j$ containing exactly k points from $\mathcal{T}_1$ irrespective of their class [40]. Similarly, $\hat{q}_{\hat{\mathbf{X}}}(\mathbf{x}_j)$ can be approximated with neighbors of $\mathbf{x}_j$ from $\mathcal{T}_2$, and we have

$$D_{\text{KL}}\big(p_{\hat{\mathbf{X}}}(\hat{\mathbf{x}})||q_{\hat{\mathbf{X}}}(\hat{\mathbf{x}})\big) \approx \mathbb{E}_{\mathbf{x}_j}\{\log\frac{\hat{p}_{\hat{\mathbf{X}}}(\mathbf{x}_j)}{\hat{q}_{\hat{\mathbf{X}}}(\mathbf{x}_j)}\} \tag{4.11}$$

$$= \mathbb{E}_{\mathbf{x}_j}\{d\log\frac{R_q(\mathbf{x}_j)}{R_p(\mathbf{x}_j)}\} \tag{4.12}$$

where $R_p(\mathbf{x}_j)$ and $R_q(\mathbf{x}_j)$ are radii of spheres determined for estimations $\hat{p}_{\hat{\mathbf{X}}}(\mathbf{x}_j)$ and $\hat{q}_{\hat{\mathbf{X}}}(\mathbf{x}_j)$ according to (4.10). And, d is the dimension of MLU input, i.e., 2404 in indoor environment classification. As a result, KLD of (4.12) returns considerably larger values compared to $D_{\text{KL}}\Big(p_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})||q_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})\Big)$ because of multiplication by d, which avoids an optimization favoring relevancy over syntax. This domination and the resulting performance degradation have also been observed in our numerical results. Therefore, in such high dimensional problems, we propose to utilize the conditional KLD to overcome this problem.

Unlike the regression problem investigated in Chapter 3, classifier output is a label with values such as zero and one, and it is not defined on $\mathbb{R}$ or a continuous interval which is a condition assumed in distribution estimation using (3.7). In other words, (3.7) and

d-dimensional sphere volume is not apt for KLD estimation in classification problems. In addition, as discussed in Chapter 3, employing the histogram is infeasible for high dimensional data, e.g., for the indoor environment classification with 2404 input variables. To solve the optimization problem of (4.7) for classifications, its conditional distributions can be estimated using a simple k-Nearest Neighbors (k-NN) approach [40]. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be data sets each with samples drawn from p and q, respectively. Therefore, the first approximated conditional distribution is given by

$$\hat{p}_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}_j|\mathbf{x}_j) = \frac{k_{\mathbf{y}}}{k}; (\mathbf{x}_j, \mathbf{y}_j) \in \mathcal{T}_1, \tag{4.13}$$

where k is the total number of nearest neighbors for a given $\mathbf{x}_j$ from $\mathcal{T}_1$ in terms of euclidean distance that we consider for each estimation, and $k_{\mathbf{y}}$ is the number of these neighbors with the same class label as $\mathbf{x}_j$, namely $\mathbf{y}_j$. In (4.13), $\mathbf{x}_j \in \mathcal{T}_1$ are full or high-precision samples. The conditional distribution, $q_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})$, is similarly approximated with k nearest neighbors of $\mathbf{x}_j$ from $\mathcal{T}_2$. In this case, we can write

$$D_{\mathrm{KL}}\Big(p_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})||q_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}|\hat{\mathbf{x}})\Big) \approx \mathbb{E}_{(\mathbf{x}_j,\mathbf{y}_j)}\Big\{\log\Big(\frac{\hat{p}_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}_j|\mathbf{x}_j)}{\hat{q}_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}_j|\mathbf{x}_j)}\Big)\Big\}; (\mathbf{x}_j, \mathbf{y}_j) \in \mathcal{T}_1. \tag{4.14}$$

By substituting conditional distribution estimations based on (4.13) in (4.14), a brute force search finds the optimum bit allocation of (4.7). These calculations can theoretically be expensive for an extensive search space, but still feasible to be done in practice, since they are only done once and offline. For more specific use cases, the search method can be modified, as it is later shown in Chapter 5. It i worth mentioning that the estimation approach of (4.13) and the KLD based bit allocation of (4.7) are used for case studies of this and next chapter which deal with high dimensional inputs.

4.4 Simulation Setup

4.4.1 Training the MLU

Each class of the data set with 1960 instances is divided into a training and test set with 60% and 40% of samples, respectively. This training set is used to train classifiers and estimate KLD of (4.7). kmeans employs input samples of the same training set for codebook design. The test set is then used for evaluation of MLU performance and bit allocations.

The first classifier is a shallow fully-connected NN with 16 neurons, trained with validation ratio of 20%. Sigmoid and soft activation functions are used in hidden and output layers. Mean-squared error is the loss function, and batch gradient descent with batch size and maximum iteration number of 100 and 4000 performs the search. The decision tree, 1-nearest neighbor and Gaussian SVM are trained using [95] with default setup. The 1-nearest neighbor classifier finds the closest neighbor of the input in terms of euclidean distance in training set, and returns the neighbor's class as its decision. The classification accuracies on test set for NN, decision tree, 1-nearest neighbor and SVM are $\approx$99.5%, 98%, 100% and 95.7%, respectively.

4.4.2 KLD Estimation for Classification

For distribution estimations with k-NN, $k = 5$. The data set $\mathcal{T}_1$ is similar to $\mathcal{T}_{\text{train}}$ in our simulations, but it can also be generated with highly precise input data. $\mathcal{T}_2$ is constructed by quantizing input samples from $\mathcal{T}_1$, feeding them into the MLU and getting the corresponding outputs. As elaborated in Chapter 3, in case of having $\hat{p}_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}_j|\mathbf{x}_j) \neq 0$ while

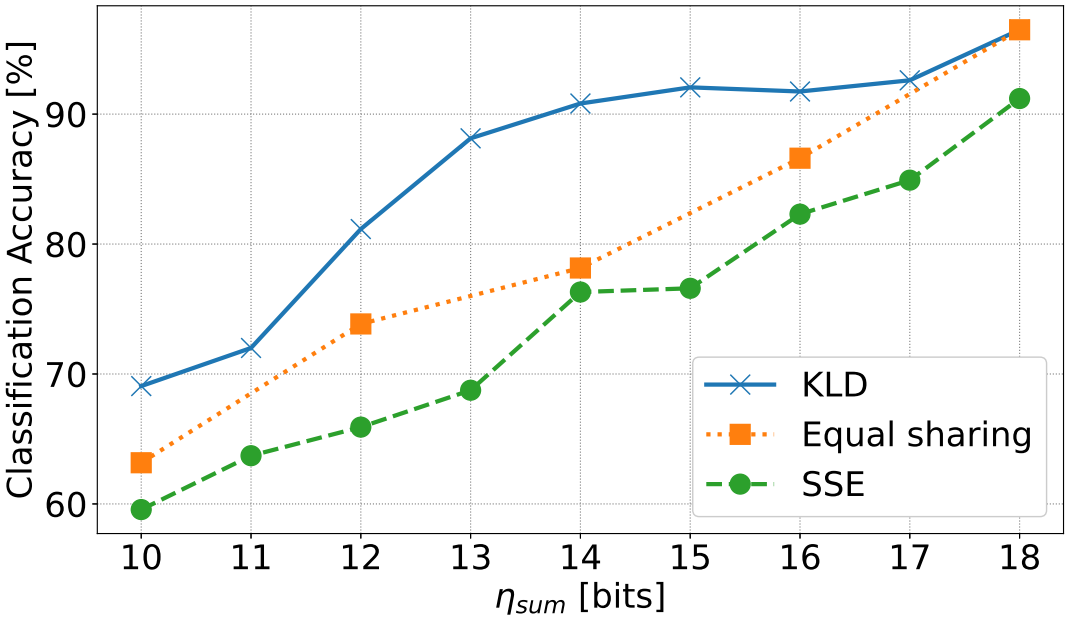


Figure 4.4: Classification accuracy vs. η_{sum} with neural network classifier.

$\hat{q}_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}_j|\mathbf{x}_j) = 0$ for a given sample, smoothing should be employed. Here, $\hat{q}_{\mathbf{Y}|\hat{\mathbf{X}}}(\mathbf{y}_j|\mathbf{x}_j) = 0.1/k$ is assumed for this purpose.

kmeans vector quantization is carried out using [95] and Elkan's algorithm. To consider a reasonable region for the search space, $4 \leq \eta_n \leq 10$. This interval is selected considering range of attribute values and size of $\mathcal{T}_{\text{train}}$. As stated, 4704 instances are used for training and vector quantization codebook design. Thus, the upper-bound on 2^{η_n} should be sufficiently smaller than 4704.

For uniform scalar quantization, it is assumed that elements of each packet have the same number of bits, and CTF and FCF packets pick their number of bits from $\{1, \cdots, 4\}$. For instance, if source 1 and 2 select 1 and 3 bits per each of their elements respectively, in total 1×1202 and 3×1202 bits are used to quantize the two vectors. Equivalently, η_1 and η_2 are selected from $\{1, \cdots, 4\} \times 1202$. This choice limits the search space of (4.7) and more importantly, considers the small range of each vector element.

To take the effect of PD into account, packets of nth source are dropped with probability of P_n, independently. In case of PD, MLU replaces missing values with a random codeword.

4.5 Numerical Results

In this section, classification accuracy of equal sharing, SSE and KLD based bit allocations is investigated. Firstly, different classifiers are studied assuming kmeans for data compression. Then, we discuss the results of 1-nearest neighbor classifier with scalar quantization. For these cases, error-free reception of MLU input is assumed. Finally, a more realistic system with PD is simulated, where data compression is done with kmeans and NN operating as the inference unit.

4.5.1 Error-free Simulations with kmeans and Different MLUs

The classification accuracy vs. η_{sum} for the error-free case with NN is depicted in Fig. 4.4. The KLD method provides the best performance at all given bandwidth constraints. The achieved gains using KLD are up to $\approx 19\%$ in comparison with SSE approach while employing a total of 13 quantization bits. Furthermore, if we target the classification accuracy

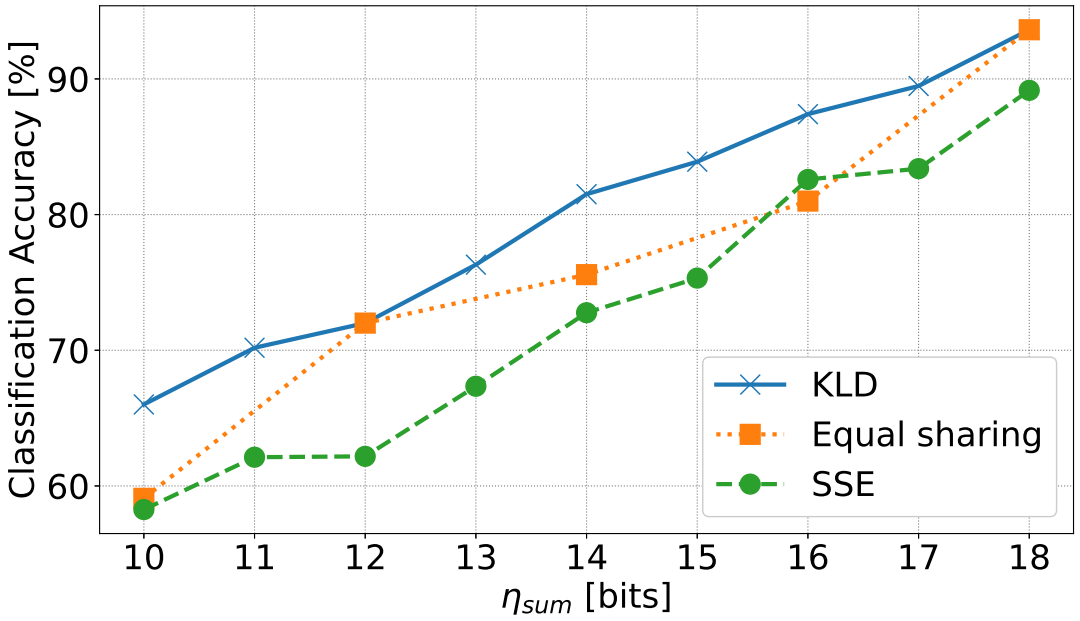


Figure 4.5: Classification accuracy vs. η_{sum} with decision tree classifier.

of 80%, the bit allocation of the KLD needs 12 bits; however, SSE strategy requires 4 more bits. It should be noted that gains and communication scheme rates in real environments with more devices and signal overheads increase rapidly.

The low classification accuracy of SSE can be explained by looking into patterns of quantization error originated from KLD bit allocations. Although a full description of the patterns along with explaining the MLU behavior is infeasible, we discuss the contributing factors in getting the best effort performance based on KLD results. Assuming that quantization error is defined as the mean-squared error between measurements and their quantized version, the KLD picks bit allocations with much lower quantization error for FCF rather than CTF for all η_{sum} values. This pattern yields highest classification accuracy, and it can be concluded that FCF has a higher level of relevancy for the NN. On the other hand, the SSE objective opts for allocations such that FCF quantization accuracy is lower than that of KLD. Therefore even with more accurate CTF compression comparing with KLD selections, considering relevancy of FCF, SSE allocations lead to more erroneous decisions.

Moreover, equal sharing outperforms SSE. SSE allocates more bits for CTF than FCF quantization. Thus, by allocating same number of bits for CTF and FCF, equal sharing enhances quantization accuracy of the more relevant input attributes which play a more important role in improving MLU decisions. Hence, it delivers a gain of 3.6% at $\eta_{\text{sum}} = 10$ regarding SSE allocation. With equal number of clusters, quantization error of FCF is lower than CTF in this problem. However, it is still not low enough to achieve the best effort performance for $\eta_{\text{sum}} < 18$. Therefore, equal sharing outcome is worse than KLD approach, e.g., $\approx$6% lower at $\eta_{\text{sum}} = 10$. For $\eta_{\text{sum}} = 18$, both KLD and equal sharing select the same allocation, implying a low enough quantization error for FCF with 9 bits at this point. Note that the higher relevancy of FCF holds for the given NN, and not necessarily for all classifiers. Furthermore, in our studies, none of the input components are determined as irrelevant by the KLD bit allocations, i.e., getting 0 quantization bits. Hence, we discuss low and high relevancy, not relevant and irrelevant attributes.

In Fig. 4.4, the highest accuracy improvements by KLD occur from 11 to 12 and 12 to 13 bits with large enhancement in FCF quantization accuracy. For the first case, KLD

Table 4.1: Classification accuracy [%] vs. η_{sum} for 1-nearest neighbor classifier. Same bit allocations are selected by SSE and KLD.

η_{sum}	9	10	11	12	13
Accuracy (KLD, SSE)	61.9	71	77.5	86.9	96.5
Accuracy (Equal sharing)	57.6	65.7	65.7	74.3	74.3

Table 4.2: Classification accuracy [%] vs. η_{sum} for Gaussian SVM.

η_{sum}	11	12	13	14	15	16
KLD Accuracy	75.8	80.2	88.8	90	92.2	94.2
SSE Accuracy	75.8	80.2	88.9	90	91.2	93.4
Equal sharing Accuracy	69.2	77.6	77.6	83.3	83.3	87.6

changes the bit allocation from $\boldsymbol{\eta}^* = [5, 6]$ to $[4, 8]$ implying an increase of two bits for FCF. And in the second one, the extra bit is again allocated to source 2 which results in highly adequate FCF quantization. Therefore from $\eta_{\text{sum}} = 13$ to 14, addition of the extra bit for FCF only leads to a small gain. Furthermore, for $14 \leq \eta_{\text{sum}} \leq 17$, number of bits for CTF increases. However, the corresponding reduction of quantization error is not large enough to make a considerable effect on decisions, and thus the outcome is improved slightly.

Fig. 4.5 shows the error-free simulations performed with decision tree classifier. Once more, the KLD approach provides much lower quantization error for FCF at all η_{sum} values, and achieves the highest classification accuracy. For example at $\eta_{\text{sum}} = 10$, it reaches the accuracy of 66% which is $\approx 7\%$ and 8% more than results achieved with equal sharing and SSE based allocation.

Equal sharing allocations result in better FCF quantization, and thus better classification accuracy compared with SSE, yet worse than KLD selections at all points except for 12 and 18 bits. At $\eta_{\text{sum}} = 12$ and 18, equal sharing chooses the same pattern as KLD by chance considering its blindness regarding relevancy, and achieves the best effort outcome, i.e., 72% and 93.6%.

In addition, SSE based results show performance loss between $\approx 4.5\%$ to 10% for $\eta_{\text{sum}} = 18$ and 12 compared with the KLD approach. Similar to the NN case, SSE method cannot keep enough relevant information in compressed data while providing lower quantization error for CTF instead of FCF. It is also worth noting that ability of this decision tree to extract relevant information is less than the NN considering their performance on test set, while a similar relevancy pattern holds for both of them. Therefore, even with the same bit allocation, e.g., [4, 6] at $\eta_{\text{sum}} = 10$, the NN achieves a higher classification accuracy than the decision tree.

Numerical results of the third error-free scenario with 1-nearest neighbor classifier are shown in Table 4.1. Here, bit allocations selected by KLD are the same as ones picked by SSE. Therefore, classification accuracy is the same for both methods. One reason for this phenomenon is the nature of this classification which is not highly complex and non-linear, thus, it can be solved with a simple 1-nearest neighbor method. In addition, this classifier is similarity based and operates in direct compliance with SSE and kmeans objectives by finding the closest neighbor of $\mathbf{x}$, without considering $\mathbf{y}$. As it can be observed, the KLD

Table 4.3: Classification accuracy [%] for $\eta_{\text{sum}} \leq 3\times 1202$, with 1-nearest neighbor classifier and scalar quantization.

	Selected bit allocation	Classification Accuracy
KLD	1×1202, 1×1202 bits	88.7
SSE	2×1202, 1×1202 bits	80.1

Table 4.4: Different setups for simulations with PD.

Setup No.	1	2	3	4
P_1	0.01	0.1	0	0.1
P_2	0.01	0	0.1	0.1

is nevertheless capable of recognizing this behavior and provides the best performance, which is the same as SSE selections in this case. Both SSE and KLD outperform equal sharing, e.g., with 22.2% gain for 13 bits.

Table 4.2 presents the classification accuracy vs. η_{sum} for the last error-free scenario with SVM. These results are similar to the case study with 1-nearest neighbor. While KLD performs slightly better than SSE at $\eta_{\text{sum}} \geq 15$, the gains are negligible considering that evaluations are done over a test set.

These results can be explained as follows. Firstly, centers of SVM basis functions are the data samples of $\mathcal{T}_{\text{train}}$, then a subset of them is selected for decision making. Thus, the first SVM stage measures the distance between input and the selected training samples, here 2209 support vectors. Furthermore, SVM can be seen as a similarity based method like 1-nearest neighbor. The difference is that basis function of the latter classifier acts as a cylinder instead of Gaussian function by keeping impact of the closest neighbor and filtering effect of other neighbors completely. High classification accuracies of the simple 1-nearest neighbor for this particular problem implies further analogy of the induced SVM to this classifier. Consequently, while the KLD still achieves the best performance, its gains are insignificant for these two systems.

4.5.2 Error-free Simulations with scalar quantization and 1-Nearest Neighbor

The classification accuracy assuming $\eta_{\text{sum}} \leq 3 \times 1202$ bits is shown in Table 4.3. The KLD method chooses 1 bit per each element for both CTF and FCF which results in 88.7% accuracy with 2404 bits. The SSE picks an allocation employing 3606 bits, while providing lower accuracy of 80.1%. The KLD gains 8.6% higher accuracy for the 1-nearest neighbor classifier, using 33% fewer bits. Unlike kmeans, scalar quantization cannot provide an effective data representation by independent quantization of attributes. For $\eta_{\text{sum}} \geq 4 \times 1202$, both methods opt for allocations with 100% classification accuracy.

4.5.3 Simulations Considering Packet Drop with kmeans and NN

Fig. 4.6 indicates classification accuracy vs. η_{sum} in presence of PD. Here, the same bit allocations provided by KLD and SSE from error-free setup with the NN are employed; however, during simulations some packets drop. The lost input is treated as missing value by the MLU, and simulation setups studied for this scenario are shown in Table 4.4. For all setups with different PD probabilities, P_1 and P_2, classification accuracy of KLD

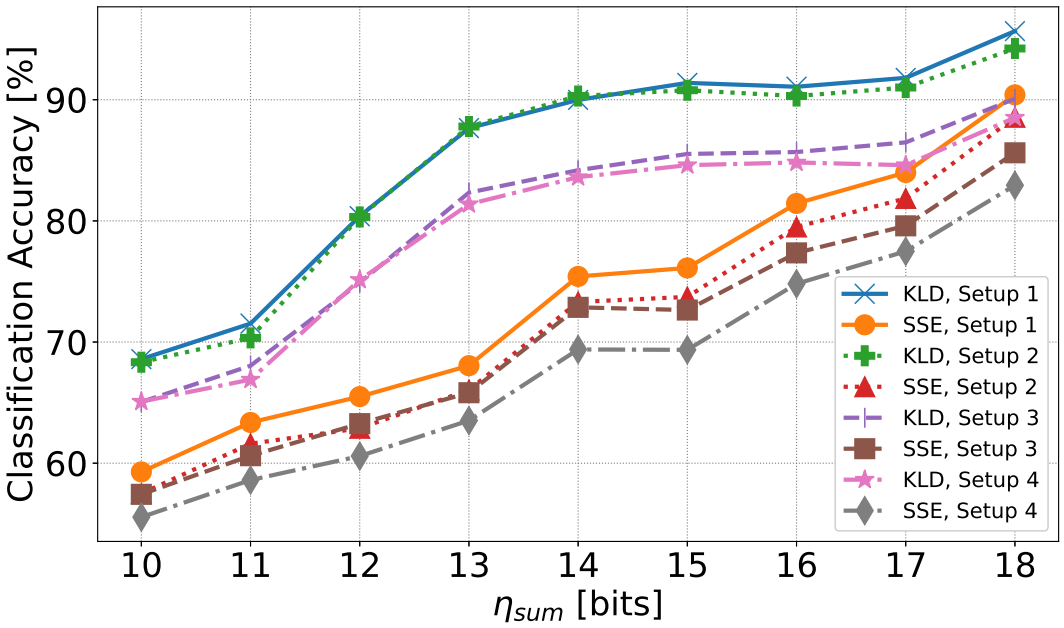


Figure 4.6: Classification accuracy vs. η_{sum} with different PDs according to Table 4.4, and neural network.

selections remains higher than SSE allocations. For $\eta_{\text{sum}} = 11$ and 4th setup with most PD occurrences, KLD provides $\approx 8\%$ higher classification accuracy than that of SSE.

For the first setup with $P_1 = P_2 = 1\%$, MLU tolerates PD with both KLD and SSE bit allocations. In other words, comparing with error-free results of Fig. 4.4, performance degradation for both methods in this setup is less than 1% and negligible. Note that probability of losing at least one packet is $\approx 2\%$, since packets of the two sources drop independently.

When drop rates of both sources increase to 10% in setup 4, classification accuracy decreases for both methods, e.g., it becomes 84.6% and 69.3% for KLD and SSE at $\eta_{\text{sum}} = 15$. Comparing with setup 1, performance loss for lower values of η_{sum} is smaller than loss at larger η_{sum} values for both KLD and SSE approaches, e.g., $\approx 3.5\%$ and 7% at $\eta_{\text{sum}} = 10$ and 18 bits for KLD. With lower number of total quantization bits, more incorrect decisions are made by the MLU and these cases have a higher overlap with PD events. Hence, the number of extra incorrect decisions only caused by PD reduces, and the performance loss is smaller in this region. As it can be seen, degradation in classification accuracy depends on both amount of overlap between packet drops and MLU failure in making correct predictions and ability of the classifier to overcome these situations, where at least one packet is delivered.

In setup 2 and 3, PD effect for data of each source is investigated. We firstly discuss the results of KLD approach. It can be observed in Fig. 4.6 that MLU performance for setup 3 is worse than results of setup 2, since the packet containing more relevant information drops. For $\eta_{\text{sum}} = 14$, classification accuracy of KLD for setup 2 and 3 is $\approx 90\%$ and 84%, respectively.

The largest performance loss of setup 2 comparing with setup 1 is $\approx 1.5\%$ at $\eta_{\text{sum}} = 18$. The reasons are: Firstly, PD only occurs for CTF with lower level of relevancy for the NN. Secondly, the KLD approach allocates high numbers of bits for the more relevant input attributes, and thus provides enough information for the MLU to predict correct output in most of cases. It can also be seen that outcome of setup 3 and 4 are similar, since in both cases FCF packet is lost. However, in setup 3, when CTF is delivered with more

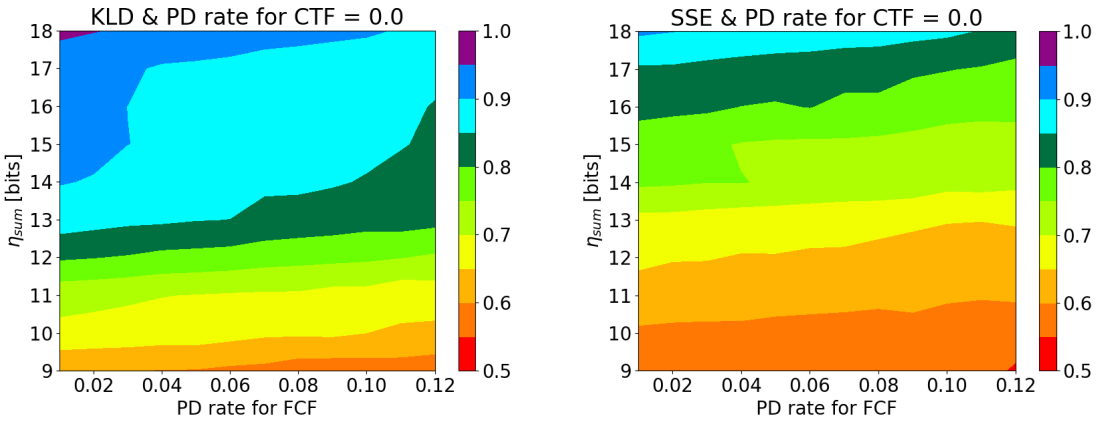
accurate quantization providing some meaningful information for MLU interpretation, e.g., at $\eta_{\text{sum}} = 18$, classification accuracy improves slightly comparing with setup 4.

Unlike results of the KLD approach, MLU performance achieved by SSE in setup 2 is not similar to setup 1, and classification accuracy of setup 3 outperforms outcome of setup 4. The KLD invests more bits in the more relevant input components resulting in high classification accuracy if these packets are delivered to the MLU. However, in case of FCF PD, not only the relevant information is dropped but also the remaining input data, CTF, is quantized coarsely. This leads to large performance loss such that KLD results of setup 3 become similar to performance of setup 4 as if both packets could drop. On the other hand, SSE allocates more bits for less relevant data of source 1. Therefore, in case of PD for CTF, the MLU performance cannot remain as high as in setup 1, but becomes $\approx 2\%$ worse for all η_{sum} values. In case of PD for FCF, adequately quantized CTF provides meaningful information for MLU decisions and partly compensates the absence of FCF data. As a result, performance degradation of setup 3 is not as much as the one occurred with KLD, and classification accuracy of this scenario is better than those of setup 4. With 16 bits, SSE achieves the accuracy of $\approx 77.5\%$ in setup 3 which is $\approx 2.5\%$ better than outcome of setup 4.

Numerical results of Fig. 4.6 with SSE are similar for setup 2 and 3 at $\eta_{\text{sum}} \leq 14$, and start to diverge by further increase in total number of quantization bits. For instance, SSE accuracy is $\approx 73\%$ with 14 bits for both setups. However, with 18 bits, the accuracy in setup 3 is $\approx 85.5\%$. This is $\approx 3\%$ worse than outcome of setup 2. This happens because SSE generally allocates the bits for CTF in the region with lower quantization bits. However, for larger η_{sum}, the number of bits allocated for FCF increases and more relevant information reaches the MLU while CTF packets are dropped. Hence, classification accuracy of setup 2 shows enhancements for $\eta_{\text{sum}} > 14$.

In order to gain more insight into the system behavior in presence of PD, we additionally study contour graphs of various scenarios with different PD probabilities for CTF and FCF. Therefore, η_{sum} vs. FCF PD probability for fixed values of PD rate for CTF, i.e., 0% and 10%, are studied if Fig. 4.7 and 4.8, respectively. In Fig. 4.9 and 4.10, the reverse case is studied. In other words, η_{sum} vs. CTF PD probability is investigated while FCF PD rate is 0% and 10%, respectively. In order to explain impact of color bar levels, Fig. 4.11 shows the same data with various endpoints for color bars. Finally, CTF PD probability vs. FCF PD probability is drawn in Fig 4.12 for the bit allocations selected by KLD and SSE approach when $\eta_{\text{sum}} = 18$ is assumed. All of these graphs are depicted with matplotlib library in python which uses a marching squares algorithm to compute contour locations. It can be seen that in all of the contour studies, the KLD approach delivers higher classification accuracies compared to those of SSE assuming the same set of restrictions.

For the first contour graph, we consider a scenario in which CTF data of source 1 is delivered to the NN with no PDs. Fig. 4.7 demonstrates η_{sum} vs. probability of dropping packets of source 2 containing FCF information. In Fig. 4.7a and 4.7b, bit allocations selected by the proposed approach and SSE are employed, respectively. The system functioning with KLD bit allocations is providing much higher classification accuracy comparing with SSE allocations. For instance, a large area of contour graph in Fig. 4.7a belongs to accuracy values between 85-95% and is painted in blue. Thus, achieving such accuracy is feasible with a large range of η_{sum} and FCF PD rates. However, the aforementioned area shrinks for the SSE assignments in Fig. 4.7b. This calls for more quantization bits, and in some cases with a constraint on η_{sum}, a more restricted FCF PD rate for achieving the same outcome.



(a) (b)

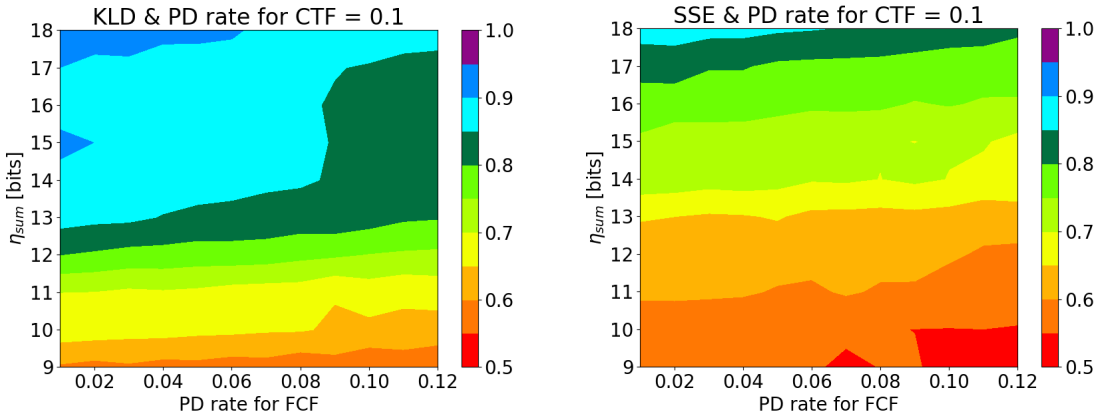
Figure 4.7: Classification accuracy vs. η_{sum} and FCF PD probability, while all CTF packets are delivered to the NN.

In addition, Fig. 4.7a shows that increased probability of PD for FCF reduces the classification accuracy, if number of bits for quantization is fixed. For $\eta_{\text{sum}} = 9$, FCF PD rate of ≤ 0.06 leads to classification accuracy of 60-65%, while an increase in PD probability results in lower accuracy of 55-60%. Moreover, losing FCF information affects the NN performance particularly, for larger values of η_{sum}. For $\eta_{\text{sum}} = 16$ bits, accuracy of 90-95% is only achieved if FCF PD probability does not exceed 3%. As mentioned earlier, a full explanation of the NN behavior is far from trivial. However, making less incorrect decisions and allocating more resources for compression of more relevant packets in this region can be among the parameters causing this phenomenon. Note that if η_{sum} is small, more incorrect decisions are made by the MLU. In this case, a higher overlap between these erroneous decisions and PD events occurs in the system and as a result, the number of extra incorrect decisions only caused by PD reduces, and the corresponding performance loss becomes smaller.

When CTF PD rate is zero, SSE assignments are less sensitive to increases in FCF PD rate in the region with low values of η_{sum} comparing with KLD selections. In Fig. 4.7b and at $\eta_{\text{sum}} = 9$ bits, the NN classification accuracy remains in the same interval, i.e., 55-60%, for all FCF PDs of less than 0.12. Only with 12% probability for the FCF PD, the accuracy falls below 55%. This behavior of SSE allocations varies for larger values of quantization bits, i.e., $\eta_{\text{sum}} \geq 14$, where the SSE approach starts to increase the number of FCF quantization bits. The likely explanation is that prior to this point, each extra quantization bit is allocated for compressing CTF by the SSE and FCF packets are quantized coarsely and incapable of providing meaningful information for the NN. Hence, FCF PDs do not cause a large performance degradation. However, with $\eta_{\text{sum}} = 16$, if we target a classification accuracy between 80 and 85%, the FCF PD probability should be below 4% which is a relatively low PD rate.

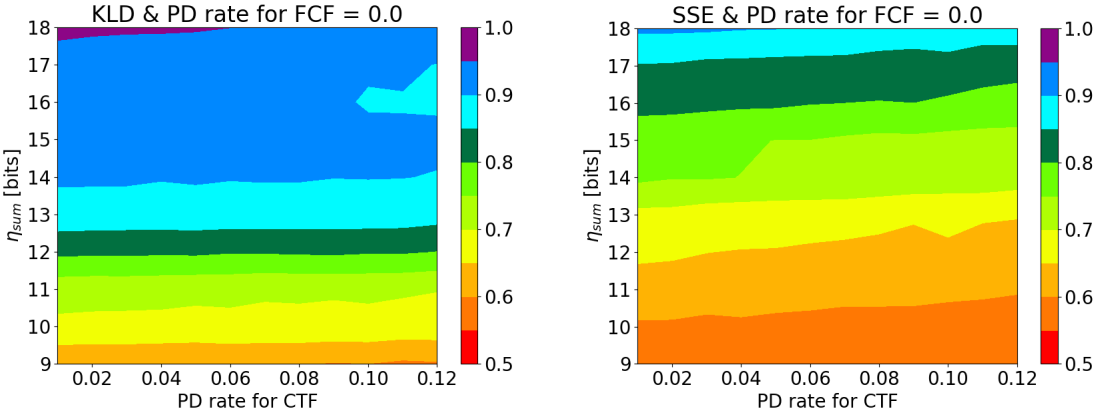
In Fig. 4.8, the CTF PD probability is increased to 10%, and η_{sum} vs. FCF PD rate is depicted. Fig. 4.8a indicates the numerical results for KLD bit allocations which generally show a similar behavior to the results presented in Fig. 4.7a with a degradation in classification accuracy. As an example, even with zero FCF PDs and assuming $\eta_{\text{sum}} = 9$, the NN accuracy is between 0.55-0.6. In other words, unlike the case with no CTF PDs, classification accuracy of 60 to 65% cannot be achieved while total number of quantization bits is 9.

As it can be observed, in the region marked with light blue, i.e., classification accuracy of 85-90%, a small area around $\eta_{\text{sum}} = 15$ and zero probability for FCF PD indicates a



(a) (b)

Figure 4.8: Classification accuracy vs. η_{sum} and FCF PD probability, while PD rate of the less relevant attributes for the NN, i.e., CTF PD probability is 10%.



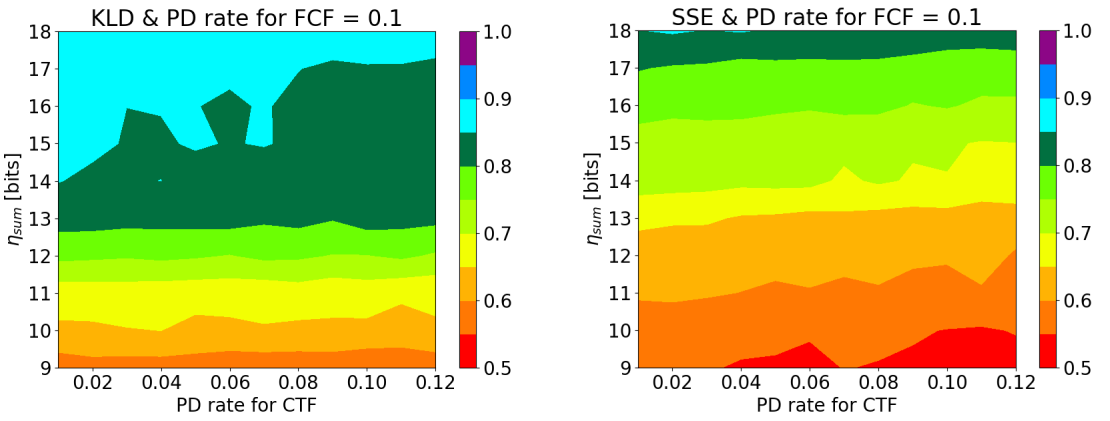
(a) (b)

Figure 4.9: Classification accuracy vs. η_{sum} and CTF PD probability, while all FCF packets are delivered to the NN.

classification accuracy of 90-95%. For this point, the NN classification accuracy is 90.4%, and the change of interval occurs because of the selected endpoints in the color bar intervals. Since the difference between this accuracy and endpoint of the light blue interval is 0.004, and the accuracy is approximated using a test set, this change can be disregarded. It is worth noting that we keep the same endpoints for the color bar intervals in almost all our figures in order to make comparisons feasible. The impact of selecting various levels is later discussed in Fig. 4.11.

For SSE bit allocations in Fig. 4.8b, classification accuracy drops when FCF PD increases. At $\eta_{\text{sum}} = 12$, FCF PD rate of $\leq 10\%$ delivers classification accuracy of 60 to 65%, while FCF PD rate of more than 10% results in a performance loss and classification accuracy of 55 to 60%. Similarly, a decrease in classification accuracy occurs for $\eta_{\text{sum}} = 9$. However, at FCF PD rates equal to 0.08 and 0.09, the classification accuracies are 55.1 and 55.4%. These numbers only indicate a small distance to 50% and since they are only estimated with a test set, we can assume that these points also belong to the accuracy interval of 50-55% marked with red color. Therefore, the visual abnormality can be dismissed.

Unlike the previous figures, we now study scenarios with fixed FCF PD rate. For this purpose, Fig. 4.9 shows η_{sum} vs. CTF PD rate with zero PD for data with more relevance for the NN, i.e., FCF information. If we compare results of Fig. 4.9a with those of Fig. 4.7a



(a) (b)

Figure 4.10: Classification accuracy vs. η_{sum} and CTF PD probability, while the PD rate for FCF which provides more relevant information for the NN is 10%.

which are both related to the KLD approach, increasing CTF PD leads to performance loss in terms of classification accuracy. However, the NN is less sensitive to CTF PD, showing no abrupt and large changes in accuracy. For $14 \leq \eta_{\text{sum}} \leq 17$, the classification accuracy drops for higher CTF PD probabilities but the loss is not visible since it roughly remains in the same interval, i.e., 90-95%. Note that similar to the previously discussed abnormalities, for $\eta_{\text{sum}} = 16$ and CTF PD rate of 0.10 to 0.12, the classification accuracy is between 89.6-89.7 which implies a negligible distance to 90%.

Fig. 4.9b shows the contour graph of SSE selections. As in previous cases, a similar reaction to increased PDs, i.e., reduction in accuracies occurs for a constant value of η_{sum}. However, when FCF PD probability is fixed and zero, slightly better outcomes can be achieved for some points, compared to the results of Fig. 4.7b with CTF PD being set to zero. For instance, the area with classification accuracy of 85-90% covers a larger area in Fig. 4.9b, where no packets with higher relevancy for the NN are dropped. On the other hand, the achieved classification accuracy with a bit allocation selected by the SSE remains much lower than that of the KLD approach under the same conditions. Assuming 13 quantization bits, the SSE selection delivers an accuracy of 65-70%; however, the KLD bit allocation provides an accuracy of 85-90%, implying a significant performance gain of $\approx 20\%$.

In Fig. 4.10, the PD probability for the more relevant packets is increased to 10%. Comparing the two figures with KLD and SSE results in Fig. 4.10a and 4.10b with those of previously investigated scenarios shows a large degradation in classification accuracy. In all of simulation setups that we considered, a classification accuracy of 90-95% with KLD allocations and 85-90% with SSE selections is achievable for the range of PD rates and η_{sum} under study. However, such classification accuracy can no longer be achieved with the high FCF PD rate of 10%.

Similar to the prior cases using SSE bit allocations, increasing the CTF PDs affects NN performance as shown in Fig. 4.10b. For $\eta_{\text{sum}} = 11$, CTF PD probability of more than 0.04 changes the accuracy interval from 65-70% to 60-65%. Having a non-negligible performance drop in presence of PD for any or both of the CTF or FCF packets has generally been observed for all contour graphs of SSE approach. This response is in compliance with results of Fig. 4.6, and are already discussed in details.

Furthermore, in Fig. 4.10a presenting KLD bit allocation results, the two seemingly abnormal drops in classification accuracy with CTF PD probabilities of 0.05 and 0.07 are

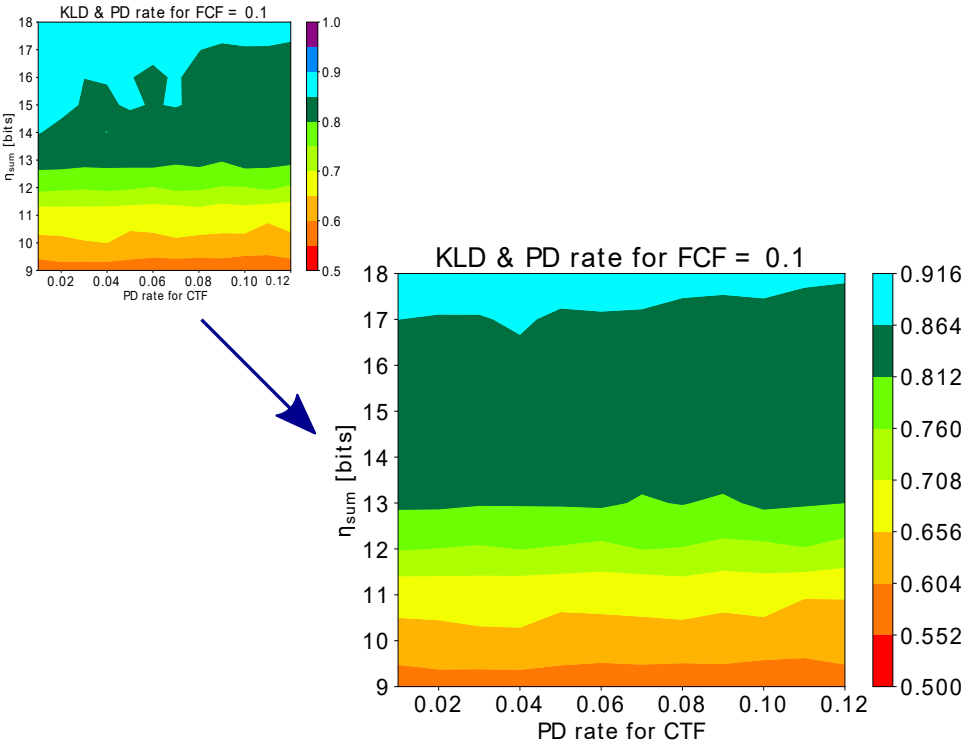


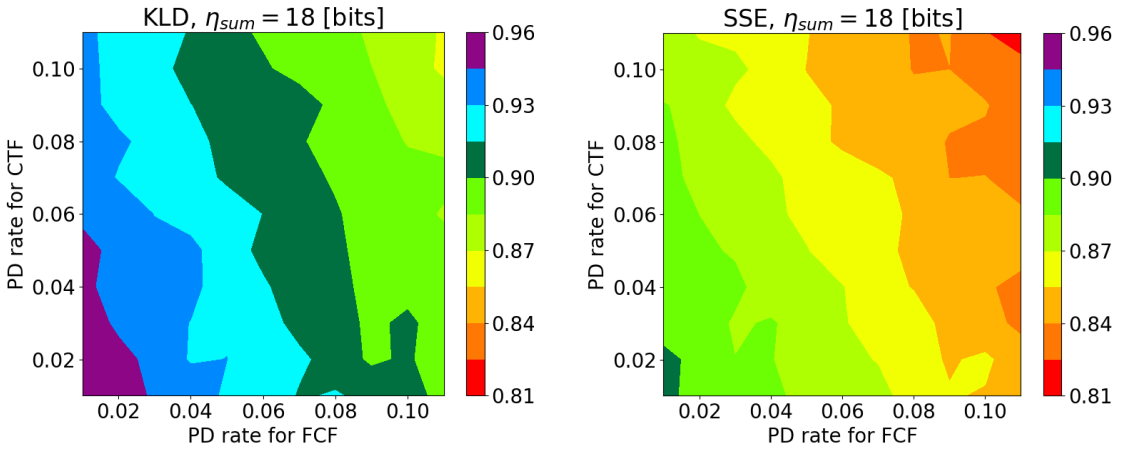
Figure 4.11: Classification accuracy vs. η_{sum} and CTF PD probability with different color bar levels. In both subfigures, the PD rate for FCF is 10% and KLD bit allocation is used.

once again caused by the selected endpoints for the color bar intervals. This condition and KLD outcome for this setup are elaborated using Fig. 4.11.

In Fig. 4.11, the endpoints of the color bar from Fig. 4.10a are slightly modified to investigate their effect on reading contour graph results. Here, the interval length is set to 0.052 instead of 0.05. As it can be seen the two performance drops of Fig. 4.10a with CTF PD probabilities of 0.05 and 0.07 are no longer visible, since the classification accuracy at these points is only slightly more than 85%. In other words, a negligible difference between classification accuracy which is itself an estimation using a test set and the endpoint of 85% results in observing a seemingly abnormal behavior that should be ignored. Studying KLD bit allocations with the new color bar shows that in case of having the high PD probability for FCF packets, increasing CTF PD rate does not lead to a large accuracy degradation[2]. This response is in compliance with that presented by setup 3 and 4 of Fig. 4.6, and its underlying reasons were justified when discussing those results. As it can be observed, interpreting contour graphs should be done carefully not only because of the complicated nature of explaining a MLU behavior but also because of the impacts caused by endpoint selections for the color bar.

Finally, Fig. 4.12 presents CTF PD probability vs. FCF PD rate when the number of quantization bits is set to 18 bits. It is important to note that with such high η_{sum}, both FCF and CTF packets are quantized with high precision. Therefore, in both Fig. 4.12a and 4.12b, losing FCF packets which provide more relevant information for the NN under study is more costly. In Fig. 4.12a, the highest achievable level of classification accuracy is between 94.5 and 96%, and it can be reached if FCF and CTF PD probabilities remain below ≈ 0.025 and 0.055. For the SSE selections as shown in Fig. 4.12b, the highest classification accuracy that can be delivered is between 90 and 91.5%, if FCF and CTF

[2]Note that performance degradation generally occurs with increasing CTF PD rate. However relatively, this loss is not large which is the focus of our discussion here.



(a) (b)

Figure 4.12: Classification accuracy vs. CTF PD probability and FCF PD probability, while the selected bit allocations of η_{sum} by KLD and SSE are used for NN input quantization.

PD probabilities remain below ≈ 1 and 2.3%. Thus, a higher PD rate can be tolerated for CTF packets regardless of the bit allocation method.

In addition, it can be concluded that the KLD approach provides much higher classification accuracy compared to the SSE allocations. These gains occur not only in the area with restricted bandwidth, low η_{sum} values, but also in case of having large bandwidth in presence of PD. Note that the bumps that can be observed in Fig. 4.12a and 4.12b are again a result of small distance between classification accuracy and endpoints of the color bar, and should be ignored in high level interpretation.

4.6 Summary and Conclusion

In this chapter, the KLD bit allocation tailored for scenarios with high dimensional inputs is applied to a data set with real channel measurements for indoor environment classification, considering scalar and vector quantization at multiple sources and four classifiers. Simulation results show its significant gains comparing with SSE and equal sharing for the NN and decision tree even when a more efficient codebook design, i.e., kmeans is utilized.

For the 1-nearest neighbor and SVM, the KLD always provides highest classification accuracies, while only delivering considerable gains for the study with scalar quantization. Therefore, it can be concluded that gain levels achieved by the proposed method not only depend on aspects such as codebook design, but also on the given MLU and nature of the use case. In other words, whether a subset of MLU input components carries more relevant information comparing with other subsets depends on different factors. In any case, the KLD approach delivers the best outcome as indicated by our numerical analysis.

In addition, the simulation results show that KLD selections do not make the MLU performance overly sensitive to PD which is a common imperfection in communications systems. This observation and the remarkable gains justify benefits of using this method for networks with integrated MLUs. As discussed, different system components influence magnitude of the gains. A future line of research is to further investigate these mutual impacts, and demonstrate their underlying relations to improve overall system performance. Modifying the method for using it in combination with adaptive learning algorithms in dynamic environments and performing the PD study while MLU is particularly trained to handle missing values are among other interesting topics to be explored.

5. Signal Overhead Reduction for AI Assisted Conditional Handover Preparation

5.1 Overview

In this chapter, the divergence based technique of (4.7) is used in terms of Signal Overhead Reduction (SOR) for a mobility case study. Since we deal with a specific use case, an additional step is proposed to quantify relevance of Machine Learning Based Unit (MLU) input data in time domain and further reduce the signal overhead.

Due to complexity of Handover (HO) management, Artificial Intelligence (AI) is envisioned as a promising candidate to assist HO procedures. However, the required signal overhead is a huge drawback of many AI based approaches. Here, we focus on signal overhead reduction for AI-Assisted Conditional Handover (AI-CHO). In this AI-CHO, a classifier performs Conditional Handover (CHO) preparations. The users transmit their received measurements from serving and neighbor cells to the classifier imposing a heavy burden on network. To this end, we introduce a 2-step solution which includes employing an additional simple classifier at user side to prevent transmission of unnecessary measurement reports. Furthermore, a pattern for number of bits compressing the remaining data with uniform scalar quantization is selected. The bit allocation is determined in accordance with the heuristic that measurements of stronger links can provide more information for the CHO classifier and need a higher quantization precision. The Kullback-Leibler Divergence (KLD) and Mean Squared Error (MSE) are used for selecting the compression patterns. The proposed approach results in remarkable gain in overhead reduction, i.e., 53% for our simulation setup, while providing similar outcome in terms of mobility Key Performance Indicator (KPI)s such as radio link failure.

5.1.1 State of the Art

With increasing demands on seamless connectivity, higher density of base stations in unit area and stringent requirements of 5G enablers, HO management turns into a challenging problem in future networks. Hence, Machine Learning (ML) capabilities in recognizing underlying patterns can be utilized to enhance mobility robustness [70]. In [96], the base station learns to predict link blockages for millimeter-wave communications, and proactively triggers HOs to provide less disconnections. As another instance, a deep learning approach in [97] reduces the occurrence of unnecessary HOs while system throughput remains unchanged.

A ML based function can be integrated at various parts of networks like User Equipment (UE) or cloud, each offering their benefits and disadvantages. An overview of these options is presented in [98]. In case of deploying ML anywhere in the network rather than in UEs, the limited resources of UEs are preserved and the challenge of transferring a trained ML model to users is overcome. However, this calls for transmission of UE data to the unit governed by AI for processing and thus, it imposes massive signal overhead on the network in addition to basic signaling required for HO preparations and executions.

Reducing the basic HO signaling has been studied in many ML based solutions by reducing the number of unnecessary HOs and Ping Pong (PP) events. For instance, [99] suggests an AI assisted HO parameter optimization for specific locations in LTE femtocells, and [100] removes unnecessary HOs between indoor femtocells and outdoor macrocells. Authors of [101] mitigate HO failures in a 5G setup which results in reduced signaling. In addition, a recent work has employed benefits of CHO and combines it with ML to prevent redundant CHOs for millimeter-wave communications [102]. Consequently, the basic signal overhead can be moderated by the ML model assisting HO procedure, however reducing the signaling for a MLU deployed in the network is not studied in the literature. Therefore, we focus on the measurement reports for the intelligent unit assuming it is not located at UE side. Note that by deployment of MLU in network, we refer to utilization of MLU anywhere in network except at UEs in the rest of this chapter.

In this chapter, an AI-CHO scenario is explored, where Reference Signal Received Power (RSRP) measurements of serving and neighbor cells are transmitted from UEs to a classifier preparing CHOs. The CHO is introduced for 5G New Radio in order to improve the baseline-HO [103] and is used for CHO executions here. The RSRP values are the input attributes of the CHO classifier and transmitting them occupies substantial network resources. Our proposed approach reduces this overhead by both restricting transmission of measurement reports and compressing the remaining data.

5.1.2 Main Contributions of the Chapter

The first step of our proposed solution is motivated by the fact that a need for CHO preparation is a rare event, when compared to instances not requiring CHO. To this end, a simple linear classifier at UE side determines whether the measurement report would trigger a CHO preparation and should be transmitted. We study two classifiers for this purpose and show the one called SOR classifier 1 with better False Negative Rate (FNR) is capable of providing best outcome. Secondly, ML units can tolerate different levels of distortion at their inputs [94]. Considering the heuristic that measurements from stronger cells carry more relevant information for a CHO preparation decision, a bit allocation pattern is selected to compress remaining data that should be delivered to the CHO classifier.

For evaluation, we examine a network with 21 cells and 605 users while shadowing effect is taken into account. The CHO classifier requiring RSRP data is a Convolutional Neural Network (CNN). Numerical results show that our method is capable of approximately halving the overhead, while delivering similar outcome in terms of various mobility KPIs, i.e, CHO preparations, Successful Conditional Handover (SCHO), PP events, Radio Link Failure (RLF)s and outage. The main contributions of this chapter are published in [104], and summarized here.

1. The bit allocation framework is used in terms of SOR.
2. A heuristic is introduced to limit the search space for finding the proper bit allocation, which reduces the corresponding computations.

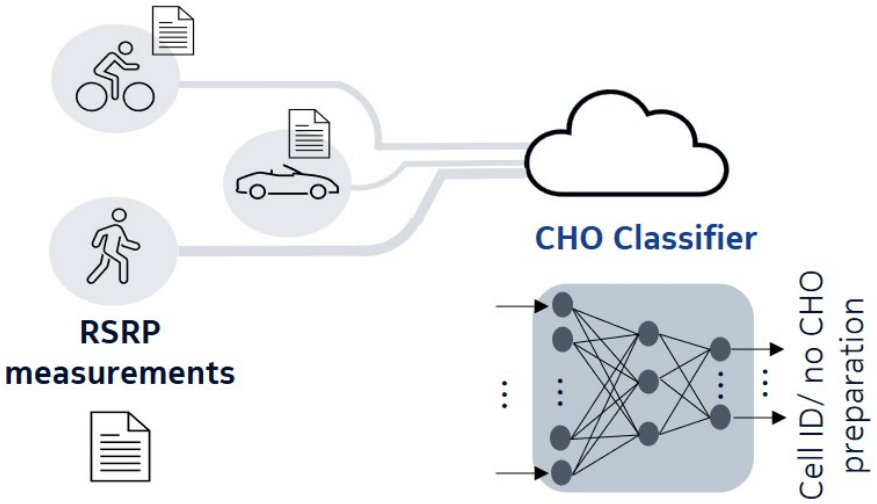


Figure 5.1: Overview of AI-CHO.

3. Unlike previous case studies, the input components with high level of relevancy for the MLU, here the CHO classifier, alternate in this problem, i.e., a particular input can be a more relevant attribute in a given time step but not in all time steps. We deal with this problem by proposing a grouping for the attributes using a heuristic.

4. Since a specific and non-general problem is studied, relevancy in time domain is additionally taken into account. To this end, a simple SOR classifier is introduced to function at the user side.

5. It is shown that in designing the SOR classifier, FNR is of a higher importance compared to the common metric, classification accuracy.

6. A two-step SOR solution is proposed to consider relevancy both in time domain and on input attributes. This solution reduces the signal overhead by 53%, while degrading the mobility KPIs only slightly.

7. It is concluded that the SOR classifier affects RLF and outage of the AI-CHO, while data compression mainly increases the number of CHO preparations.

8. It is shown by numerical results that KLD and MSE bit allocations may opt for the same allocation or different assignments depending on the given SOR constraint defining their search space. Several cases from both of these categories are discussed. Although the KLD approach does not deliver gains in comparison with MSE selection for some given SOR constraints, it always provides the best outcome.

This chapter is organized as follows. The system model is discussed in Section 5.2, where the AI-CHO problem, the CHO classifier, mobility KPIs and benchmarks are elaborated. In Section 5.3, the proposed signal overhead reduction strategy is introduced. The simulation setup is elaborated in Section 5.4, and numerical results are presented in Section 5.5. Finally, a summary and conclusions are drawn in Section 5.6.

5.2 System Model

5.2.1 Case Study 3: AI-CHO and its KPIs

The CHO technique decouples HO preparation and execution. In comparison with the baseline HO, it prepares target cells early when the link to serving cell is still strong.

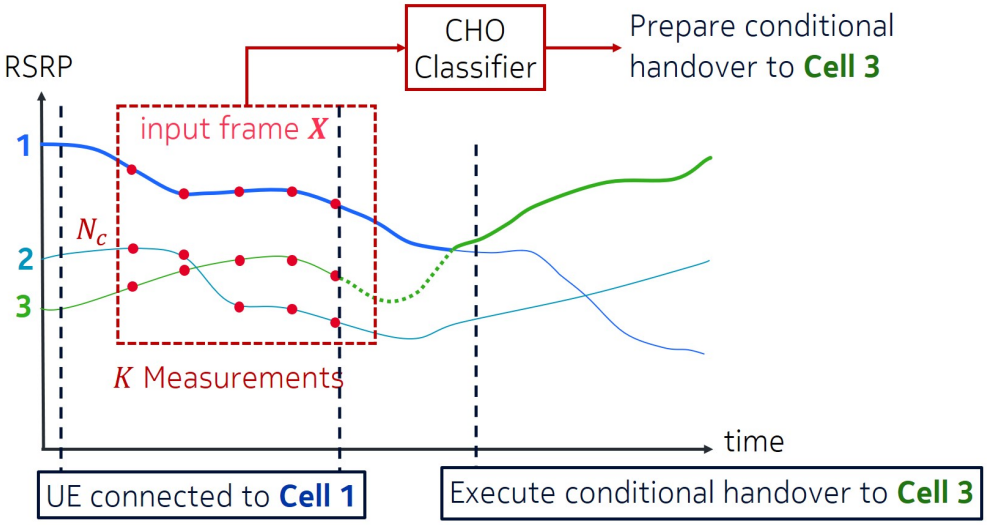


Figure 5.2: Schematic diagram of CHO classifier operation in inference mode. AI-related blocks are shown in red.

Additionally, access to the target cell is performed late when the corresponding radio link is sufficiently strong [103]. In order to further enhance CHO performance and predict best cells for CHO preparation, a classifier is utilized here as depicted in Fig. 5.1. The classifier decides whether to establish a preparation, and in case of a positive decision, determines a cell as the target cell. This classifier and case study are referred to as CHO classifier and AI-CHO. The CHO classifier completely replaces the preparation process and CHO execution is led by the CHO condition which is given by

$$P_s(t) + o^{\text{exec}}_{c_s,c_t} < P_t(t) \text{ for } t_{\text{exec}} - T_{\text{TTT}} < t < t_{\text{exec}}, \tag{5.1}$$

where $P_s(t)$ and $P_t(t)$ are RSRP values of serving and target cells at a given time t, respectively. In (5.1), $o^{\text{exec}}_{c_s,c_t}$ stands for the CHO execution offset defined between the serving and target cells. The execution time is shown by t_{exec} and T_{TTT} presents a certain time interval, called time to trigger, during which the RSRP condition must hold before a CHO execution occurs. Thus, not all CHO preparations triggered by the classifier are being executed.

Here, we study a network in which N_u users collect RSRP values of N_c surrounding cells every t_s sec. For each UE, RSRP reports of K time steps form the input attribute matrix $\mathbf{X}_{N_c \times K}$. This matrix is then transmitted to the CHO classifier which is a CNN with convolutional, rectified linear unit, fully connected and softmax layers. A schematic diagram to visualize operation of the CHO classifier in inference mode is depicted in Fig. 5.2, where the red dot marks in the box are elements of $\mathbf{X}_{N_c \times K}$.

The distance between two consecutive samples is chosen to be $25 \times t_s$. During inference, the knowledge of all RSRP values which are only t_s sec apart is assumed at the CHO classifier. In other words, after transmission of data marked with red dots, the window slides for t_s sec and the new data matrix is transmitted to the classifier. The sampling and sliding procedure are shown in Fig. 5.3. Furthermore, the CHO classifier outputs a vector with N_c softmax probability values for each UE every t_s sec. If a probability value, except the one related to the serving cell, is larger than a predefined threshold L as shown in Fig. 5.3, a CHO preparation is triggered.

The procedure for generating a data set for the CHO classifier is elaborated in Fig. 5.4. In this process, best label is selected based on the mean value of cell RSRPs over a window

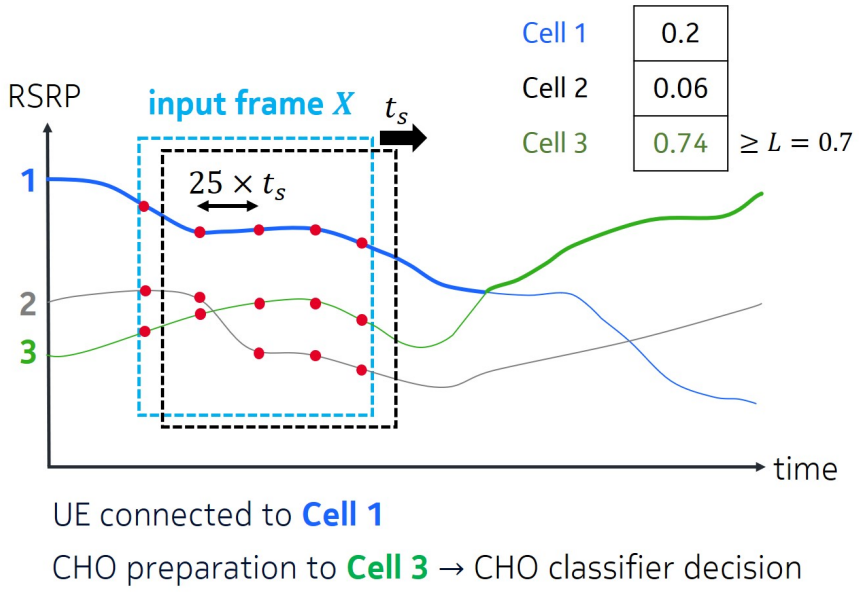


Figure 5.3: Overview of the sampling procedure and decision making process by the CHO Classifier in inference mode.

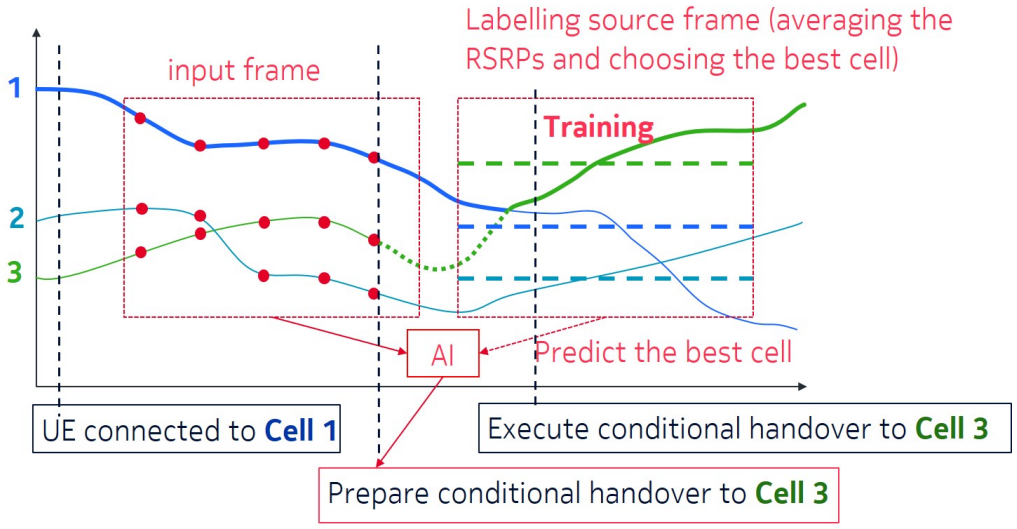


Figure 5.4: Data set generation for the CHO classifier. AI-related blocks are shown in red.

with 5 samples, each positioned $25 \times t_s$ sec apart. The CHO classifier is trained with data collected from various network users and hence, is a global model used for all UEs.

Since most of ML based units are developed independently of communications system design, the CHO classifier has been trained offline with full-precision data. Afterwards, it remains unchanged while functioning as an inference unit. In network, the CHO classifier receives $\hat{\mathbf{X}}_{N_c \times K}$ which is a compressed version of $\mathbf{X}_{N_c \times K}$ with uniform scalar quantization.

The mobility outcome is analyzed with the following commonly used KPIs. These KPIs are all normalized to show values per user per minute in the rest of this chapter.

CHO Preparations: Total number of CHO preparations per UE per minute that are performed successfully in the network. It is assumed that if the Signal-to-Noise-plus-Interference Ratio (SINR) between serving cell and the UE is not above Q_{out}, either measurement report or CHO command cannot be delivered leading to unsuccessful CHO preparation process.

Successful CHO (SCHO): Total number of CHOs per UE per minute from a serving cell to a target cell that are completed in the network. For a CHO execution to be completed successfully, the SINR on a link between UE and target cell should be above Q_{out} so that UE can perform synchronization and random access procedure. If CHO execution procedure is not completed within the time interval of $T304$ sec, it fails. The $T304$ timer starts when CHO execution condition is met. In case of failure, re-establishment procedure is initiated. More details about this 3rd Generation Partnership Project (3GPP) timer can be found in [105]. Note that SCHO targets successful CHO executions, and not preparations which is considered by the first introduced KPI.

Ping-pong (PP): PP shows the number of PP events occurring per UE per minute. PP handovers refer to cases when a UE is handed over from one cell to another, and it is quickly handed back to the original cell. Here, a PP event is detected when a successful CHO from cell A to B for a given UE is followed by a second CHO from cell B to cell A within a predefined T_{pp} sec.

Radio Link Failure (RLF): The number of RLF events per UE per minute is shown with this important KPI that reveals the mobility performance. If the link quality between a UE and its serving cell falls below a predefined level for a certain time, UE experiences service interruption. In this study, the link quality is assessed with SINR. Herein, when the SINR on a link between a UE and its serving cell is below SINR threshold Q_{out} continuously for $T310$ sec, UE declares RLF and initiates re-establishment procedure. The timer $T310$ starts when the SINR falls below Q_{out}, and the UE has a chance to recover during the constant time defined by this timer. This is a simplified version of the $T310$ timer for 5^{th} Generation of Mobile Network (5G) New Radio as described in [105].

Outage: Outage represents the time period, in seconds per UE per minutes, during which the UE experiences service interruption, i.e., cannot transmit or receive data due to weak link quality, CHO or re-establishment procedure after RLF. As discussed, it is assumed that if the SINR between serving cell and a UE falls below Q_{out}, UE experiences outage. Besides, when UE initiates a CHO execution procedure, it has to disconnect from its serving cell and attempts random access until CHO procedure is completed. During this time, UE experiences service interruption, since it is neither served by a source cell nor by target cell. In addition, re-establishment procedure after UE experiences RLF also requires additional time, i.e., 0.1 sec, for UE to find a suitable cell for connection.

5.2.2 Benchmarks

The focus of this dissertation is on determining and delivering relevant information to the given MLUs in network. Therefore, we firstly compare outcome of the CHO classifier after applying the proposed SOR with that of the CHO classifier receiving non-quantized data. A comparison between AI-CHO and conventional CHO is out of scope of this work. In the second benchmark each RSRP value is quantized with 7 bits which is in accordance with RSRP quantization in 3GPP specifications. For bit allocation, both KLD and MSE selections are studied. More details about the benchmarks are provided in Subsection 5.3.2.

5.3 The Proposed Signal Overhead Reduction

***Problem Statement* 2:** In this chapter, we aim at finding an operating point for the given AI-CHO, considering relevance of data for the CHO classifier, at which signal overhead is kept to a minimum, while mobility KPIs remain approximately unchanged.

To tackle the signal overhead dilemma, we avoid transmission of some RSRP matrices that belong to no CHO preparation events using a linear SOR classifier at UE side. For the rest of RSRP reports, a bit allocation is chosen that provides enough information for the CHO classifier to make accurate decisions.

5.3.1 SOR Classifier

As the first step, a SOR classifier with a simple hypothesis is trained to function at UEs. For the rest of this script, CHO and no CHO are used to refer to preparation part of CHO. This binary classifier decides if a RSRP matrix belongs to a CHO or no CHO event, which are respectively referred to as positive and negative labels. Here, output of the CHO classifier is our ground truth to decide if a RSRP matrix is related to a CHO event or not.

For training this classifier, we propose to asses the confusion matrix instead of the commonly used classification accuracy. For a binary classifier, a confusion matrix is a 2×2 matrix that includes number of true and false positives, and true and false negatives. The reason for the proposal of using confusion matrix instead of accuracy is that a false negative results in no transmission of RSRP data while a CHO preparation is necessary. This affects mobility KPIs like RLF. Therefore, the SOR classifier requires a significantly low FNR. On the other hand, false positives lead to unnecessary transmission of data but cause no harm to KPIs considering the upcoming process in the CHO classifier. Clearly, lower number of false positives yields more reduction in signal overhead but improving it should not cost the FNR to enlarge considerably. An analysis of these effects is presented in Section 5.5. The FNR is

$$\mathrm{FNR} = 1 - \frac{N_{\mathrm{TP}}}{N_{\mathrm{P}}}, \tag{5.2}$$

where N_{TP} stands for the number of true positives, i.e., CHO related samples predicted correctly as positive by the classifier. N_{P} is the number of positive samples in training set. Similarly, False Positive Rate (FPR) is calculated by

$$\mathrm{FPR} = 1 - \frac{N_{\mathrm{TN}}}{N_{\mathrm{N}}}, \tag{5.3}$$

where N_{TN} and N_{N} denote the number of true negatives and negative samples in training set. These two metrics are later used for SOR classifier model selection.

Performing simple calculations at the UE to avoid draining its power and computational resources is an important design aspect to take into account. That is why we use logistic regression as our hypothesis. This linear model has the same number of parameters as dimensionality of the input feature space. Here, input attributes of SOR classifier are the same as those of the CHO classifier. Therefore, probability of having a CHO case given $\mathbf{X}_{N_c \times K}$ becomes

$$p(\mathrm{CHO}|\mathbf{X}_{N_c \times K}) = \frac{1}{1 + \exp^{-\sum_{c,\kappa} x_{c\kappa} \times w_{c\kappa}}}, \tag{5.4}$$

where $x_{c\kappa}$ is the element in cth row and κth column of $\mathbf{X}_{N_c \times K}$ and $w_{c\kappa}$ denotes SOR classifier parameters to be learned. In our simulations, the conditional probability with larger value determines a negative or positive label. Since SOR classifier operates at UE, it works with non-quantized values $x_{c\kappa}$.

To train the SOR classifier, training and test sets are generated with the CHO classifier. In our simulations, more than 90% of samples point to no CHO events. This indicates a highly imbalanced data set. To deal with this issue, undersampling of no CHO cases is carried out. This imbalance exists in all mobility scenarios, implying that even a relatively low rate of true negatives can noticeably reduce signaling.

It is worth mentioning that in addition to crossing cell borders, other factors such as shadowing enforce CHOs. Hence, only considering the UE location information is not sufficient to make accurate predictions. This validates using a pre-processor like the SOR classifier at UE side rather than making a decision for transmission of RSRP data only based on the distance of a UE to the border of its serving cell.

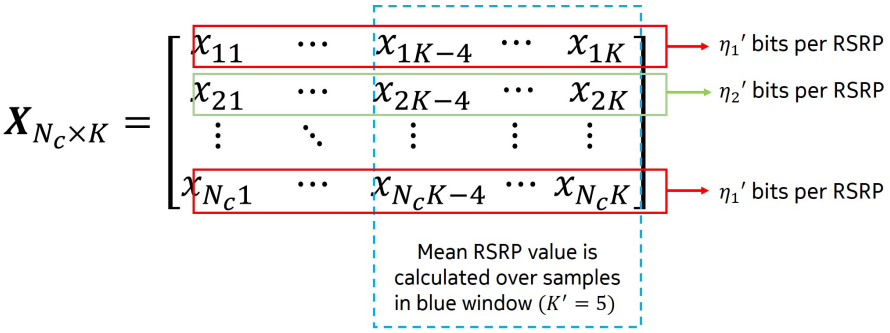


Figure 5.5: Overview of the bit allocation concept with $\eta_1' > \eta_2'$, where η_1' is used for RSRP quantization of N_c' stronger cell marked with red box.

5.3.2 Quantization Bit Allocation

In this subsection, we search for a bit allocation to quantize RSRP matrices categorized for transmission by the SOR classifier. For this purpose, KLD as presented in (4.7) is used, where $\mathbf{X}$ in (4.7) is the vector of input attributes for the CHO classifier, i.e., all elements of $\mathbf{X}_{N_c \times K}$. In order to make comparisons, MSE is additionally utilized to measure loss as follows

$$\boldsymbol{\eta}^* = \underset{\boldsymbol{\eta} \in \mathcal{H}}{\operatorname{argmin}} \sum_{\mathbf{X}_{N_c \times K} \in \mathcal{T}_{\text{train}}} \mathbb{E}_{x_{c\kappa}}\{(x_{c\kappa} - \hat{x}_{c\kappa})^2\}, \tag{5.5}$$

where $\boldsymbol{\eta}^*$, $\boldsymbol{\eta}$ and $\mathcal{H}$ represent the selected bit allocation, a feasible bit allocation and the set of all feasible bit allocations. And, $\mathcal{T}_{\text{train}}$ and $\mathbb{E}\{\cdot\}$ stand for training set and expectation operation. Here, $x_{c\kappa}$ is quantized with $\eta_{c\kappa}$ bits, and in compliance with RSRP quantization in 3GPP specifications $\eta_{c\kappa} \leq 7$ is assumed [106]. The number of feasible allocations $\boldsymbol{\eta}$ for scalar quantization and a matrix with $N_c \times K$ elements is $7^{N_c \times K}$ which is potentially large. So, we limit the search space with a heuristic approach.

As discussed earlier, ML based models extract the knowledge in data and are able to handle noise at their input. In HO management, a preparation decision is more dependent on RSRP values of cells with stronger links over last time steps of the observation window. To capture this more meaningful information, a lower quantization noise on data of these cells is expected. Thus, for each cell, we calculate its mean value of RSRP over last K' columns of $\mathbf{X}_{N_c \times K}$ as shown in Fig. 5.5. In this case, η_1' and η_2' bits are assumed for RSRP quantization of N_c' stronger and $N_c - N_c'$ weaker cells, and $\eta_1' > \eta_2'$. This heuristic shrinks the search space, allowing for a full-search to find $\boldsymbol{\eta}^*$. This search is performed once and offline on $\mathcal{T}_{\text{train}}$ and does not enforce extra computations at UEs.

In the input matrix of the CHO classifier, each row carries RSRP information of a specific cell based on a predefined numbering. Hence, a given row in $\mathbf{X}_{N_c \times K}$ can be once quantized with η_1' if it is among N_c' stronger cells, or once with η_2' if it belongs to the group of weaker cells. In this case, to decode the received bits into RSRP values $\log_2 \binom{N_c}{N_c'} = \log_2 \frac{N_c!}{(N_c - N_c')!\ N_c'!}$ side information bits should be transmitted with each $\mathbf{X}_{N_c \times K}$ to specify the subset of stronger cells.

One essential difference between this case study and the ones from Chapter 3 and 4 is that, considering the discussed heuristic, the more relevant attributes for the classifier alternate during time. For instance at a given time step and for a specific UE, cell 1-4 are the stronger cells providing more relevant attributes and at another time step cells 3-6. Therefore, the grouping of cells into stronger and weaker ones not only limits the search space but also overcomes this issue.

Table 5.1: Simulation parameters of the network. TX stands for transmitter.

Parameters	Value
Carrier frequency	2 GHz
Cell layout	7-site hexagon
Inter-site distance	500 m
TX antenna height	30.5 m
UE height	1.5 m
TX antenna element gain	14 dBi
TX azimuth beamwidth	70°
TX elevation beamwidth	10°
TX maximum backwards attenuation	25 dB
Downlink transmit power	29 dBm/PRB
Noise power	-97 dB
Frequency dependent path-loss component	128.1 dB
Distance dependent path-loss exponent	3.76
Penetration loss	20 dB
Shadow fading	Log-normal $\sigma = 8$ dB
Shadow fading correlation distance	50 m
Fast fading	According to [107]
Total number of pedestrians	105
Total number of street UEs	500
Pedestrians' speed	3 km h^{-1}
Street UE speed	30 km h^{-1}
TX system bandwidth	100 MHz
Physical resource block (PRB) bandwidth	10 MHz
Outage threshold Q_{out}	-8 dB
T310 timer	1 sec
T304 timer	0.2 sec
L3 filter time constant	0.1 sec
CHO preparation offset $o^{\text{exec}}_{c_s,c_t}$	3 dB
CHO time to trigger T_{TTT}	0.12 sec
Ping-pong timer T_{pp}	5 sec

Note that $\boldsymbol{\eta}$ is determined, once the values for η'_1 and η'_2 are selected. And, η'_1 and η'_2 are picked for all users, they are not chosen separately for each UE. In other words, $\mathcal{T}_{\text{train}}$ for estimating MSE in (5.5) and KLD in (4.7) contains $\mathbf{X}_{N_c \times K}$ of all users.

5.4 Simulation Setup

5.4.1 Network Layout and the CHO Classifier

The network under study consists of a layout with 21 cells and serves $N_u = 605$ users as shown in Fig. 5.6. It accounts for path-loss, fast and slow fading. Our simulation parameters are described in Table 5.1 in which PRB stands for physical resource block. With the CHO classifier performing preparations, the CHO preparation parameters are no longer relevant.

The input layer of the CHO classifier normalizes its input data which are matrices with $N_c = 21$, $K = 30$ and $t_s = 0.01$ sec. This classifier includes convolutional, rectified linear

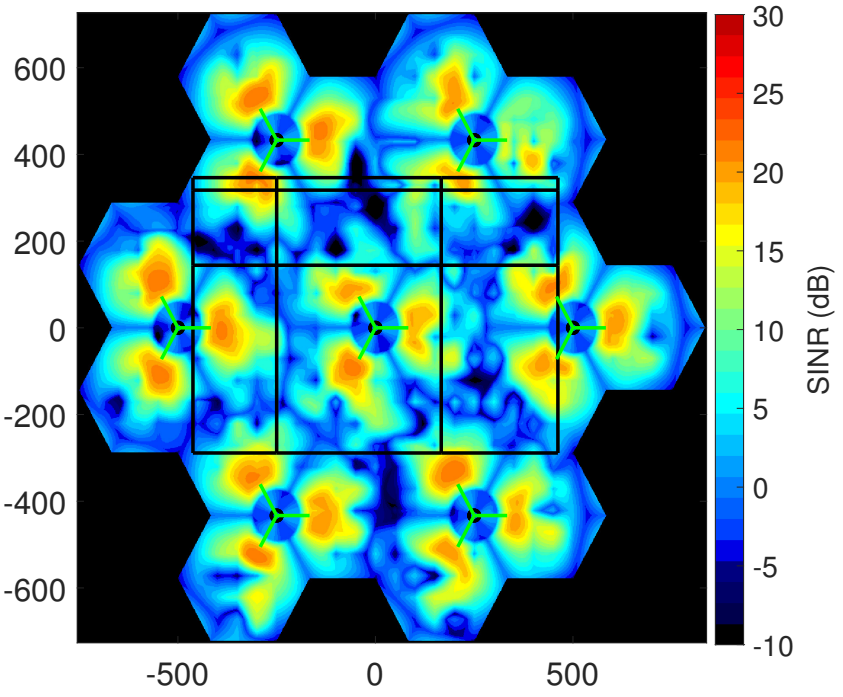


Figure 5.6: Network layout covering an area of 1400 m × 1600 m. Streets are marked by black lines.

unit, fully connected and softmax layers. The 2D convolutional layer consists of 50 filters with size 21 × 10. Both stride and dilation factor are set to $[1, 1]$, and no padding is performed. The fully connected layer has 1050 neurons, and cross entropy is used as loss function. Furthermore, only one target cell can be prepared for each user and $L = 0.7$. In case of determining a better target cell for any user, the old target cell is replaced with the new decision. Note that the focus of this paper is on SOR. Working on ML aspects, e.g., feature selection and finding the best CHO classifier is not our intention, particularly since AI based designs are not always rigorous, and if provided from a different vendor cannot be adjusted.

5.4.2 The SOR Classifier and Bit Allocation

To train the SOR classifier, we randomly selected samples from simulation data and undersampled the negative cases. The random selection provides less correlated samples for training and undersampling improves achieved FNRs during the learning process. As mentioned earlier, achieving a low FNR or equivalently, finding positive labels with near one accuracy is essential in this problem. And, it can be reached by facilitating the training with large enough CHO samples. Therefore, the training set contains approximately 4000 positive and 1000 negative events. Split ratio for validation is 0.3 and test set has ≈2500 and 500 CHO and no CHO cases.

We employed stochastic gradient descent as optimizer with learning rate of 0.5, decay factor of 0.01 and batch size of 64. The loss function during training was cross entropy. Regularization factor was set to zero since it was unable of improving results over the test set. Maximum number of epochs was 500. The final selection of model and hyperparameters are done according to FNR and FPR. These metrics for the first SOR classifier are ≈2% and 67% over the test set, respectively. Therefore, we expect SOR classifier 1 to avoid transmission of $0.9 \times (1 - 0.67) \approx 30\%$ of RSRP matrices considering the ≈90% no CHO events in simulations and the FPR value. This was confirmed by our simulations showing a SOR gain of ≈28.5%.

Table 5.2: Simulation results. Outage is in sec per UE per min. Other KPIs show number of events per UE per min. SOR is computed with benchmark 1 or 2 as reference depending on quantization type. Here, both KLD and MSE select the same bit allocation.

	CHO Preparations	SCHO	PP	RLF	Outage	SOR (%)
Non-quantized (Benchmark 1)	2.67	1.99	0.040	0.19	1.30	-
7 bits per RSRP (Benchmark 2)	2.91	1.99	0.040	0.19	1.29	-
KLD and MSE Selections	3.54	1.97	0.035	0.19	1.29	35
SOR Classifier 1 Non-quantized	2.68	1.97	0.039	0.21	1.32	28.5
SOR Classifier 1 7 bits per RSRP	2.91	1.97	0.039	0.21	1.33	28.5
SOR Classifier 1 KLD and MSE Selections	3.51	1.95	0.035	0.21	1.32	53
SOR Classifier 2 Non-quantized	2.68	1.92	0.030	0.25	1.40	44
SOR Classifier 2 7 bits per RSRP	2.86	1.92	0.030	0.25	1.40	44
SOR Classifier 2 KLD and MSE Selections	3.39	1.89	0.029	0.25	1.40	63

In order to emphasize the importance of having low FNR, a second SOR classifier is also trained and studied with decay factor of 0.05 and shuffled data. It delivers FNR and FPR of $\approx 6\%$ and 49% over the test set, and its classification accuracy of $\approx 87\%$ is similar to that of the first SOR classifier. A SOR of $\approx 44\%$ is expected by only using this classifier.

For the data compression, $K' = 5$ and $N_c' = 4$. Hence, 13 side information bits are required. This amount is negligible compared to number of bits for RSRP matrix quantization. KLD is estimated using k-Nearest Neighbors (k-NN) as in Chapter 4 with $k = 10$. The KPIs are calculated over a simulation period of 300 sec accounting for 30000 time steps t_s and decisions.

5.5 Numerical Results

When solving (4.7) and (5.5), imposing different levels of SOR to define the set of feasible allocations $\mathcal{H}$ leads to getting various $\boldsymbol{\eta}^*$. Depending on the SOR constraints, KLD and MSE opt for same bit allocation or different ones. The operating point that we choose for the AI-CHO is one of the points with same bit allocation selected by both approaches. Thus, the corresponding results individually and in combination with the SOR classifier are investigated in Subsection 5.5.1. Afterwards in Subsection 5.5.2, we discuss some other feasible operating points at which KLD and MSE pick different bit allocations, and the KLD selection provides a higher gain in terms of SOR. For instance, a point at which the resources are very limited and the SOR constraint on η_{sum} is restricted, and another case when the SOR constraint is loose.

5.5.1 Numerical Results of the Proposed Method

In this subsection, mobility KPIs are investigated for various case studies. The first benchmark assumes full-precision data is fed to the CHO classifier. This scheme provides the best mobility performance but is impractical for implementation. As the second benchmark, a practical conventional scenario is considered in which each RSRP value is quantized with 7 bits. To evaluate our proposed method, we apply the KLD compression, which is similar to that of MSE, to the first benchmark instead of a 7-bit quantization. Then, SOR classifier 1 is added to both benchmarks 1 and 2. These studies analyze independent impact of each individual SOR step on mobility KPIs. The next case combines both steps of the proposed approach using SOR classifier 1. The simulations with SOR classifier 1 are then repeated using SOR classifier 2 which has a worse FNR. In the rest of this subsection, we first discuss elaborate numerical results presented in Table 5.2. Then, a summary is provided in Fig 5.7 and 5.8.

In our simulations, SOR classifier 1 avoids transmission of 28.5% of data. The selected bit allocation achieves overhead reduction of 35% over remaining data comparing with benchmark 2. Therefore, the SOR of 53% is achieved in total while KPIs degrade only slightly as shown in Table 5.2. On the other hand, SOR classifier 2 prevents 44% of transmissions. This leads to 63% total overhead reduction when combined with KLD based quantization; however, RLF and outage become non-negligibly worse, i.e., they are 0.06 and 0.11 more than those of benchmark 2 because of the higher FNR. The SOR rates in Table 5.2 are calculated considering benchmark 2 for comparison. The exceptions are scenarios in 4th and 6th row of Table 5.2, assuming benchmark 1 as the reference, since they all work with non-quantized data.

As it can be seen in Table 5.2, by applying the quantization of 2nd benchmark using a large number of bits for RSRP compression, the only non-negligible degradation is the increase of 0.24 in number of CHO preparations. Since other KPIs remain similar, these extra CHOs can be accounted as unnecessary CHOs. However, occurrence of such performance losses cannot be prevented in our system, since transmission of exact non-quantized data is infeasible. Hence, benchmark 1 is investigated as the best case but the main point of reference for evaluating our results is outcome of benchmark 2.

In comparison with the second benchmark, employing the bit allocation picked by the KLD, which is the same as the MSE selection, results in an increase of 0.63 in number of CHO preparations. And, while number of PP events per user per minute is improved by 0.005 with the KLD selection, SCHO is 0.02 lower than that of benchmark 2. However, this SCHO decrease does not necessarily imply a performance degradation. Here, the number of SCHOs and PPs are counted independently and thus, a PP event indicates occurrence of 2 SCHOs. In this case, a lower SCHO along with lower PP value points to reduction of some undesired SCHOs which would have resulted in PP events. As a result, the main loss caused by the KLD allocation can be assumed to be the increased number of unnecessary CHO preparations considering that parameters like RLF and outage remain unchanged. In comparison with benchmark 1, again the essential degradation is the 0.87 extra CHO preparations.

The KPIs presented in fourth and fifth row of the table demonstrate impact of using only SOR classifier 1 on our benchmarks. Firstly, we consider the forth row and compare KPIs of the first benchmark with the same case plus this SOR classification. It can be observed that SCHO, RLF and outage degrade by 0.02 in case of pre-processing data at the SOR classifier 1. In this scenario, the other KPIs remain approximately the same as the first benchmark. Similarly, with respect to the quantization with $\eta_{c\kappa} = 7$ in benchmark 2, addition of the SOR classifier to system as presented in 5th row affects SCHO, RLF and outage by 0.02, 0.02 and 0.03 per UE per minute, respectively. Hence, the SOR classifier 1

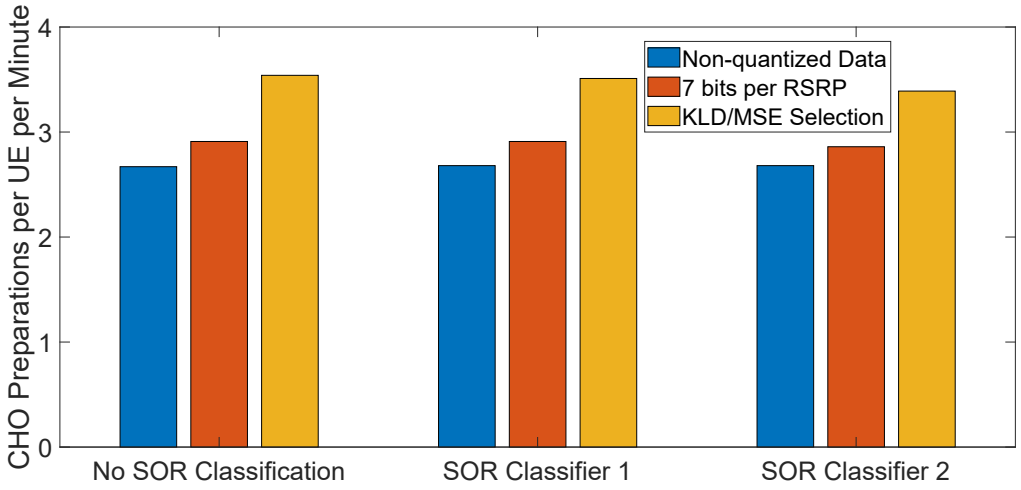


Figure 5.7: CHO preparations per UE per minute for different scenarios.

slightly affects these three KPIs. Moreover, observing outcome of 4th and 5th scenarios indicates that adding data compression even with $\eta_{c\kappa} = 7$ brings a degradation to number of CHO preparations per UE per minutes, i.e., from 2.68 to 2.91.

A combination of both data pre-processing at UEs and compression further affects the KPIs. For instance, the KLD quantization in addition to using the SOR classifier 1 yields 0.6, 0.02 and 0.03 more CHO preparations, RLF and outages considering the main benchmark. The average number of SCHO per UE per minute is also reduced by 0.05 while PP is enhanced by 0.005. These undesired effects are a small penalty for achieving the remarkable SOR of $\approx 53\%$ in this case.

The last three cases use the 2nd SOR classifier at UEs. Table 5.2 shows that all these studies have the same RLF and outage performance of 0.25 and 1.4. These numbers imply a loss of 0.04 and 0.06 on RLF comparing with the similar case using the SOR classifier 1 and benchmark 2, respectively. This non-negligible loss shows the importance of observing FNR for SOR classifier design. Here, while FPR of SOR classifier 2 is better than that of SOR classifier 1, its FNR is worse resulting in this performance loss. Utilizing the second SOR classifier also reduces the SCHO from 1.99 in benchmark 2 and 1.97 in 5th case study to 1.92 considering 7-bits quantization. Part of this loss is compensated by enhancement of 0.01 in PP events, i.e., 0.02, but the SCHO of this scenario remains lower than that of our main benchmark.

The number of CHO preparations with the SOR classifier 2 is smaller than that of the first SOR classifier when quantization is applied. For example with the KLD compression, the network with SOR classifier 2 experiences 3.39 preparations per UE per minute which is 0.12 less than that of the similar network with SOR classifier 1. This difference is not visible, if we compare the case studies employing full-precision data with the two SOR classifiers. Thus, this reduction is mainly achieved by removing unnecessary CHO preparations caused by data compression, considering the assumption that the CHO classifier provides the ground truth. And since SOR classifier 2 prevents more transmissions comparing with SOR classifier 1, it can reach this reduction. However, avoiding transmission of CHO events regarding ground truth in all studies with SOR classifier 2 results in performance loss on RLF and outage.

To summarize our numerical results, number of CHO preparations for different scenarios are depicted in Fig. 5.7. This KPI has shown largest variations which are mainly caused by data compression. All cases with non-quantized data have similar number of CHO preparations, while the small differences occur when 7-bits quantization is applied. The

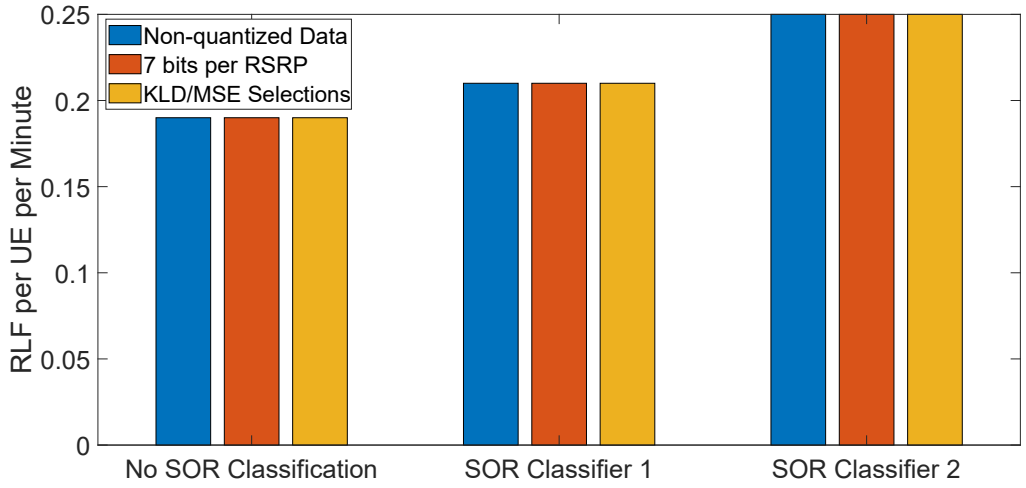


Figure 5.8: RLF per UE per minute for different scenarios. The RLF of 2nd classifier is 0.06, i.e., approximately 30% worse than that of the benchmark without SOR classification. Such degradation is not negligible in mobile networks and implies the importance of training SOR classifiers with low FNR. Thus, we generally recommend utilization of the 1st SOR classifier with lower FNR and RLF loss.

larger jumps comparing with full-precision cases are related to the KLD compression. For instance in case of using the KLD bit allocation instead of 7 bits per RSRP with SOR classifier 1, 0.6 more preparations take place in the network.

Since RLF is one of the most important indicators for evaluating quality of service experienced by UEs, we investigate it in Fig. 5.8. This figure shows the number of RLF events per UE per minutes for different SOR scenarios. As discussed earlier in details, implementation of the SOR classifier worsens RLF depending on FNR. As it can be seen, the largest RLF is caused by the SOR classifier 2 with worst FNR. Note that this loss is 0.06 compared with benchmark 2, however, even a small number of service interruptions can considerably affect the UE experience. Hence, depending on network requirements, a compromise between FNR and FPR needs to be reached which directly influences mobility performance and SOR gain.

5.5.2 Additional Investigation on KLD and MSE Bit Allocations

Heretofore, the AI-CHO used a SOR constraint, defined by η_{sum}, that leads to obtaining the same bit allocation from both MSE and KLD approaches. However, there are some SOR constraints that result in MSE and KLD opting for different bit allocations. Providing an explanation to answer where the selections are different and why is far from trivial because of many contributing factors. As discussed in Chapter 3 and 4, estimation accuracy, nature of the problem, the MLU and codebook design are some of these parameters. In addition, for the mobility problem under study, an unbalanced distribution between the number of street UEs and pedestrians, the heuristic for shrinking search space including the grouping and assuming $\eta'_1 > \eta'_2$ can influence the selections. In this subsection, we focus only on the compression step of the proposed method and provide additional numerical results to review two cases when MSE and KLD bit allocations are not identical. At these points, the KLD technique delivers a higher SOR gain.

In the first case that we present, SOR constraint is restricted such that at least 45% overhead reduction, comparing with benchmark 2, is achieved. In other words, $\eta_{\text{sum}} \leq 81 \times K$ for determining $\mathcal{H}$. And, the assumption on $\eta'_1 > \eta'_2$ is removed. In this scenario, the KLD and MSE methods select two different bit allocations using 71 and 80 bits in total

Table 5.3: Additional simulation results with a restricted SOR constraint. Outage is in sec per UE per min. Other KPIs show number of events per UE per min. SOR is computed with benchmark 2 as reference.

	CHO Preparations	SCHO	PP	RLF	Outage	SOR (%)
7 bits per RSRP (Benchmark 2)	2.91	1.99	0.040	0.19	1.29	-
KLD and MSE Selections at the operating point	3.54	1.97	0.035	0.19	1.29	35
KLD Selection with restricted constraint	4.82	2.01	0.039	0.20	1.30	52
MSE Selection with restricted constraint	4.65	1.95	0.035	0.20	1.30	46

for quantizing each column of $\mathbf{X}_{N_c \times K}$, respectively. In Table 5.3, the KPIs for each of these selections are studied and compared with those of benchmark 2 and the KLD bit allocation from Subsection 5.5.1. RLF and outage are the same for both of the bit allocations with restricted SOR constraint. The number of CHO preparations and PP events per UE per minute are slightly more with the KLD selection, i.e., ≈ 0.17 and 0.004 comparing with MSE assignment. However, SCHO and SOR of the divergence based method are 0.06 and 6% better than those of MSE approach.

Comparing KPIs of the two bit allocations with the restricted SOR constraint and benchmark 2 shows that RLF, outage and PP of all allocations are similar. However, with either KLD or MSE compression with the new SOR constraint, number of CHO preparations are 4.82 and 4.65 which are considerably higher than that of benchmark 2, 2.91 preparations per UE per minute. That is why, this point is not fit to be the operating point of the system. Nevertheless, it can still be utilized in particular scenarios, if the resources are very limited. For instance in case of having a much lower channel quality, this point can be utilized as a backup for the main operating point of AI-CHO, temporarily.

RLF and outage at the operating point are 0.19 and 1.29. These KPIs become 0.20 and 1.30 with both KLD and MSE methods, when the more restricted SOR constraint is applied. Furthermore, the number of PP events per UE per minute for these studies are 0.035, 0.039 and 0.035, while the worst PP performance with 0.004 difference belongs to the KLD selection with restricted constraint. As it can be seen, the differences between provided RLF, outage and PP of these three cases are negligible. On the other hand, the SCHO of the KLD in third row of the table is 2.01. Even after considering the effect of worse PP into account, this KPI is ≈ 2 which is 0.03 and 0.05 higher than SCHO of the operating point and MSE compression with restricted constraint, respectively. This improvement is achieved at the cost of having more CHO preparations which is 1.28 and 0.17 more than the number of preparations occurring with the proposed compression and MSE allocation with the restricted constraint. Consequently, the restricted constraint imposes many unnecessary CHO preparations on the system regardless of the bit allocation method. However, when comparing outcome of the allocations suggested by KLD and MSE, the KLD assignment provides a similar performance on KPIs in general, while delivering 6% more overhead reduction.

Table 5.4: Additional simulation results with loose SOR constraints. Outage is in sec per UE per min. Other KPIs show number of events per UE per min. SOR is computed with benchmark 2 as reference.

	CHO Preparations	SCHO	PP	RLF	Outage	SOR (%)
7 bits per RSRP (Benchmark 2)	2.91	1.99	0.040	0.19	1.29	-
KLD Selection with $\eta_{\text{sum}} \leq 121, 141 \times K$	3.54	1.97	0.035	0.19	1.29	35
MSE Selection with $\eta_{\text{sum}} \leq 121 \times K$	3.25	1.99	0.039	0.19	1.29	23
MSE Selection with $\eta_{\text{sum}} \leq 141 \times K$	3.02	1.99	0.040	0.19	1.29	11.5

In the second case which we briefly discuss, once again we assume $\eta_1' > \eta_2'$ and set the SOR constraint to loose values, $\eta_{\text{sum}} \leq 121 \times K, 141 \times K$, respectively. For all the SOR constraints, the KLD approach still sets the bit allocation to $\eta_1' = 7$ and $\eta_2' = 4$ bits. However, the MSE objective goes for $\eta_1' = 7$ and $\eta_2' = 5, 6$, respectively. Therefore, although all the bit allocations reach similar KPIs as presented in Table 5.4, the KLD assignment achieves 35% overhead reduction and MSE allocations achieve 23% and 11.5% reductions, respectively. Note that lower SCHO of the KLD is partly compensated by its lower PP. And, the SORs imply gains of 12% and 23.5% when using the divergence based method. More importantly, as in all previous case studies, the KLD based approach provides the best outcome in all scenarios and with all given conditions.

5.6 Summary and Conclusion

In this chapter, we investigated a network employing CHO execution process and a classifier for CHO preparations. The classifier is deployed at the network and delivering its data imposes a large signaling overhead on system. Therefore, we suggest to run a simple classifier at UEs for partially detecting no CHO preparation cases. To this end, two classifiers with various FNR were trained and studied. Furthermore, the compression of data with a bit pattern allocating more bits for cells with stronger RSRPs and the KLD approach is proposed. The AI-CHO scenario is then evaluated with and without applying the proposed SOR approach. The SOR classifier with lower FNR has only insignificant impact on RLF, outage and SCHO. The compression mostly affects the number of CHO preparations. Hence, any or both of these steps can be utilized depending on system requirements. The combination of these steps using SOR classifier 1 delivers 53% overhead reduction, a huge gain at cost of inconsiderable loss for mobility KPIs.

It is additionally shown that there exists some points including the operating point that we selected for the AI-CHO, where both KLD and MSE methods opt for the same bit allocations. However, in case of having dissimilar allocations selected by these techniques, the KLD assignments always provide the best SOR results, mainly at the cost of a slight degradation on number of CHO preparations, when compared with MSE allocations. Therefore, utilizing the divergence based method remains promising and is capable of delivering gains depending on the system requirements and conditions.

As a future line of research, improving the SOR classifier design can be taken into account, for instance by using Support Vector Machine (SVM) or the principle component analysis in combination with a simple k-NN classifier with a low value for k, e.g., $k = 1$, to keep the hypothesis and computations simple.

6. Relevance Based Wireless Resource Allocation

6.1 Overview

In Chapter 5, the divergence based bit allocation is employed in combination with a Signal Overhead Reduction (SOR) classifier to reduce signaling overhead of the AI-Assisted Conditional Handover (AI-CHO). In this chapter, the bit allocation framework is expanded to take the effect of dynamic channel states into account and provide a scheme for wireless resource allocation.

Due to vast capability of Machine Learning (ML), many applications are foreseen to deploy it in near future including internet of things use cases. In many of these cases, communications system has only one shot, namely a restricted time frame, to deliver input data of Machine Learning Based Unit (MLU) before data becomes out-of-date. Since MLU input components have different levels of impact and relevancy on output prediction accuracy, considering these aspects in resource management for networks of MLUs enhances system performance and resource utilization. However, the concept of relevancy is not addressed for such cases including a ML based centralized control system in literature, which is our case study in this chapter.

6.1.1 State of the Art

Most of the instantaneous best effort resource allocations for uplink, e.g., [108–111], aim at maximizing sum of utilities, where utility is a function of conventional Quality of Service (QoS) metrics such as data rate. These utilities and their optimization constraints are tailored for the network characteristics under study. For instance for delay-sensitive services, effective capacity is the utility introduced in [109] with constraints on power and interference. In [110], throughput and delay are inputs of the exponential utility for a heterogeneous network guaranteeing a minimum level of effectiveness in resource utilization. And, [111] maximizes a weighted sum rate with time-varying weights to deal with fairness and ensure stability of queues. As it can be seen, these techniques are not particularly developed for supporting a network of MLUs, and seek to provide high utility and fairness for all users. However, in a network of MLUs with sources containing redundant MLU input information, reducing utility of one device to increase utility of another terminal can be beneficial. This also implies the advantage of defining novel QoS metrics focused on MLU outcome.

The resource allocations attempting to assign different priority and importance levels for various data types or streams explore particular use cases. As an example, [112] suggests a utility maximization while users and each of their multimedia applications get a priority and preference factor, respectively. The focus is however on user priority and system preference factors are constant. Additionally, these factors are not systematically determined and not applicable to a network of MLUs. The same circumstance holds for the context information and priority levels introduced in [113]. Furthermore, [114] investigates sports data analytics and proposes an approach selecting important data streams to be transmitted to the cloud. For wireless virtual reality as another special case, utility depends on metrics such as position and orientation tracking accuracy in [115]. Thus, relevance of the information plays a role in resource allocation.

For taking relevancy into account without imposing many limiting assumptions, we suggest the use of a lookup table per MLU with different sets of quantization bit requirements, assuming that attributes of a MLU are transmitted from multiple terminals. Each lookup table is built using the divergence based approach as described either in (3.6) or (4.7) depending on problem setup, and resources are allocated with a proposed greedy algorithm such that for each MLU, at least one payload requirement from its table is satisfied.

6.1.2 Main Contributions of the Chapter

In this chapter, we revisit the resource allocation problem for a ML based centralized control system. For such systems with limited resources, we propose a resource allocation approach to deliver as much relevant information as possible to MLUs at each time instance and provide best effort performance. Thus, the QoS is defined to be an indicator of MLU outcome. Furthermore, each source has more than one quantization codebook. This novel aspect of having several payload requirements for groups of terminals introduces a new degree of freedom reducing packet drops, which is so far not considered in literature.

The codebooks are designed using a relevance based Kullback-Leibler Divergence (KLD) approach, which imposes different patterns on payload or bit allocation requirements summarized in a lookup table. Given channel coefficients and KLD lookup tables, a heuristic Greedy Resource Allocation Algorithm for KLD Based Lookup Table (GKLD) is proposed to share the available bandwidth among terminals and fulfill payload requirements from lookup tables. The proposed method is examined for a network of inverted pendulums on carts with MLUs as controllers. The conventional Greedy Maximum Sum Rate (GMSR) is our benchmark, studied once with KLD and once with equal sharing lookup tables. This way, the impact of both lookup tables and the proposed algorithm are investigated, separately and jointly. Numerical results show significant gains in MLU performance and resource utilization achieved by our approach. The main contributions of this chapter are published in [116], and summarized here.

1. The framework of our relevance based bit allocation is broadened to be used in terms of wireless resource allocation and in a more dynamic system, where fast fading is taken into account.

2. A novel QoS metric is defined, which directly targets performance of MLUs in the network.

3. A new aspect, i.e., utilization of lookup tables to capture the relevancy of information is proposed. The lookup tables provide various payload requirements for groups of terminals to be satisfied depending on the channel quality and available resources in the network.

4. A greedy algorithm is introduced in order to solve the resource allocation problem with the novel QoS measure, while taking the lookup tables into account.

5. The proposed method is applied to a network of cart inverted pendulums for various scenarios, while two benchmarks are investigated to evaluate the outcome. Numerical results show significant gains in performance of the controllers in terms of steady state error probability and resource utilization, in particular when the wireless resources are restricted. In other words reaching a target on the Key Performance Indicator (KPI) requires either less bandwidth or signal-to-noise power ratio, or with a given amount of these resources, the proposed approach can deliver better steady state error probabilities.

This chapter is organized as follows. The system model, formulating the KLD based bit allocation for a network with multiple MLUs, the optimization problem for resource allocation and the network of cart inverted pendulums are presented in Section 6.2. In Section 6.3, the proposed greedy bandwidth allocation algorithm using KLD lookup tables is introduced and benchmarks are explained in details. The simulation setup is elaborated in Section 6.4, and numerical results are discussed in Section 6.5. Finally, conclusions are drawn in Section 6.6.

Important Notation: Vectors and sequences are typeset boldface. Capital script typefaces denote sets, and card$(\cdot)$ is their cardinality. $(\cdot)^*$, $|\cdot|$ and $\hat{(\cdot)}$ represent optimal values, euclidean norm and quantized version of a given message. Subscripts n and m refer to the terminal and MLU index, e.g., $\mathbf{x}_{m,n}$ presents the input components of mth MLU which are transmitted from nth terminal, while $\mathbf{x}_m$ is the sequence of all MLU input components. To facilitate readability of this chapter, Table 6.1 summarizes the most relevant notations.

6.2 System Model

6.2.1 General Description

As shown in Fig. 6.1, we study a multiple access channel scenario with N memoryless stationary sources measuring input components of M MLUs. Similar to previous cases, we assume that learning process is carried out with full or high-precision data because MLUs are mostly trained independently of the communications system design. Afterwards, MLUs remain unchanged while working as inference units with quantized data in our network. Here, $\{1,\dots,N\}$ is the set of terminal indices and $\mathcal{N}_m$ is a subset of it with source indices providing input of $(\cdot)_m$th MLU. The full-precision input of this MLU is the sequence of attribute vectors $\mathbf{x}_m = (\mathbf{x}_{m,n})_{n\in\mathcal{N}_m}$, and $\mathbf{y}_m$ is its output. $\mathbf{x}_{m,n}$ is the data of nth device, and its quantized version $\hat{\mathbf{x}}_{m,n}$ should be delivered to mth MLU during T to make a timely decision. Otherwise, data becomes obsolete. Assuming a stringent constraint, resource allocation to carry available data has to be done in a single attempt. T is assumed to be smaller than channel coherence time.

Let us consider a limited available bandwidth B divided into N_{RB} similar Resource Block (RB)s of length $\tau < T$, each shown with an index $(\cdot)_r \in \{1,\dots,N_{\mathrm{RB}}\}$. The set of RB indices allocated for nth source is denoted by $\mathcal{C}_n$. Hence, the maximum payload that can be provided for nth device while $n \in \mathcal{N}_m$ is

$$\lambda_{m,n}^{\text{achievable}}(\mathcal{C}_n) = \sum_{r\in\mathcal{C}_n} \left\lfloor \frac{B}{N_{\mathrm{RB}}} \times \log_2(1+\gamma_{\max}|h_{n,r}|^2) \times \tau \right\rfloor, \tag{6.1}$$

where $h_{n,r} \overset{\text{i.i.d}}{\sim} \mathcal{CN}(0,1)$ is the Rayleigh fading channel coefficient for nth device on rth RB, assuming mutual independence. $\gamma_{\max}$ stands for the maximum signal-to-noise power

Table 6.1: Summary of the important notations.

N, n	Number of terminals, terminal index
M, m	Number of MLUs, MLU index
$\mathcal{N}_m$	Set of terminal indices transmitting to MLU m
$\mathcal{L}_m$	Lookup table of MLU m
$\mathbf{x}_{m,n}$	Data vector of terminal n to be transmitted to MLU m
$\mathbf{x}_m, \mathbf{y}_m$	Sequence of full-precision input and output of MLU m
N_{RB}, r	Number of available RBs, RB index
$h_{n,r}$	Channel coefficient for terminal n on rth RB
$\gamma_{n,r}$	Signal-to-noise power ratio for terminal n on rth RB
$\gamma_{\max}$	Maximum signal-to-noise power ratio for $\gamma_{n,r}$
$\mathcal{C}_n$	Set of RB indices allocated to terminal n
N_{th}	Primary limit on number of RBs allocated for each MLU
$\lambda_{m,n}^{\mathrm{achievable}}(\mathcal{C}_n)$	Achievable payload for terminal n transmitting to MLU m
$\boldsymbol{\lambda}_m^{\mathrm{achievable}}$	Sequence of achievable payloads for input of MLU m
$\boldsymbol{\eta}_i^*$	ith entry of a lookup table and ith payload requirement
$\mathcal{K}$	A feasible resource allocation for all terminals
$e_m(\mathcal{K})$	Performance indicator of mth MLU as a function of $\mathcal{K}$
$\mathcal{A}$,$\mathcal{U}$	Set of allocated and unassigned RBs
$\mathcal{F}$	Set of source indices Alg. 6.1 fails to meet their requirements

ratio, and it is assumed to be similar for all terminals on all RBs. The operator $\lfloor \cdot \rfloor$ returns the greatest integer equal or less than its input to consider the discrete rate and payload requirements introduced by lookup tables as explained in next subsection. For the mth MLU, $\boldsymbol{\lambda}_m^{\mathrm{achievable}} = \left(\lambda_{m,n}^{\mathrm{achievable}}(\mathcal{C}_n) \right)_{n \in \mathcal{N}_m}$. This term is a sequence containing maximum amount of payload that can be delivered to mth MLU from nth terminal given RB resources determined by $\mathcal{C}_n$, while $n \in \mathcal{N}_m$ shows the indices of terminals which are supposed to transmit data to mth MLU.

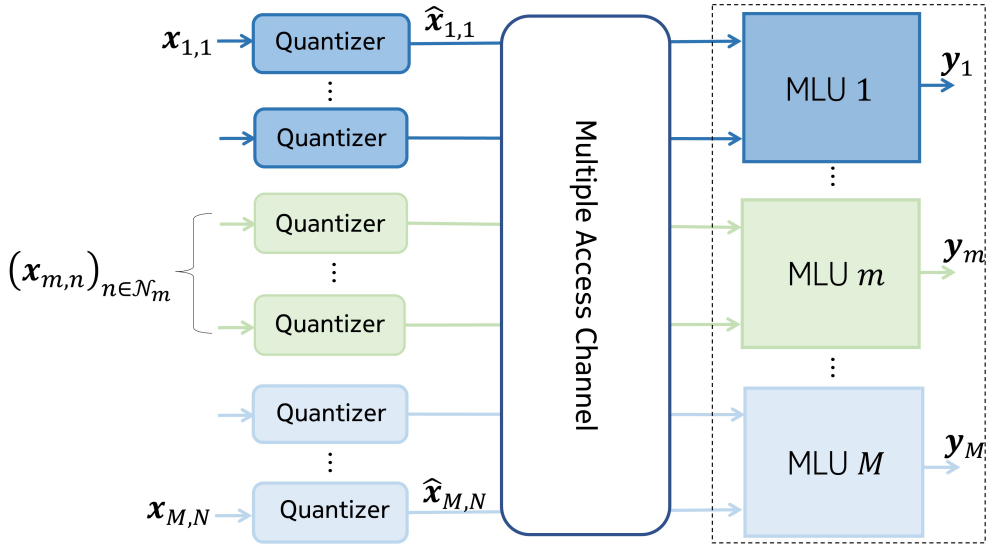


Figure 6.1: Block diagram of the system model.

Table 6.2: Lookup table example, 4 sources transmit to the mth MLU.

i	$\eta_{i,\text{sum}}$	$\boldsymbol{\eta}_i^*$ [bits]	e_i
1	46	$\boldsymbol{\eta}_1^* = (\eta_{1,1}^*, \ldots, \eta_{1,4}^*)$	0
...	...	...	...
5	42	$\boldsymbol{\eta}_5^* = (\eta_{5,1}^*, \ldots, \eta_{5,4}^*)$	0.1

6.2.2 KLD Based Lookup Table of Payload Requirements

Table 6.2 shows an example of a lookup table for a given MLU m consisting of multiple options as required payloads to be delivered during T, where $\mathcal{N}_m$ has 4 members. The bit allocation in ith row of this table is shown as $\boldsymbol{\eta}_i^* = (\ \eta_{i,n}^*\ )_{n\in\mathcal{N}_m}$, and it is picked given a constraint i on total number of quantization bits $\eta_{i,\text{sum}}$. The proposed resource allocation takes current channel gain information into account and intuitively, it aims at finding assignments such that at least one requirement $\boldsymbol{\eta}_i^*$ is satisfied for each MLU, i.e., $\forall m, \exists \boldsymbol{\eta}_i^* \in \mathcal{L}_m, \boldsymbol{\lambda}_m^{\text{achievable}} \geq \boldsymbol{\eta}_i^*$, where $\exists$ and $\mathcal{L}_m$ are existential quantifier and the lookup table of mth MLU.

Note that a bit allocation is designed for a group of sources transmitting to a MLU, and not for individual sources. In other words, it forms a pattern of quantization distortion that can be tolerated at MLU input.

To take the relevancy of input components into consideration, we utilize the KLD based approach as introduced in (3.6) for lookup table design, sine we deal with the inverted pendulum case study in this chapter. For problems with high dimensional MLU input, (4.7) can be employed. The KLD bit allocation selects the ith payload requirement of mth MLU based on the following criterion

$$\boldsymbol{\eta}_i^* = \underset{\boldsymbol{\eta}_i}{\operatorname{argmin}}\, D_{\text{KL}}(p_{\hat{\mathbf{X}}_m,\mathbf{Y}_m}(\hat{\mathbf{x}}_m,\mathbf{y}_m)||q_{\hat{\mathbf{X}}_m,\mathbf{Y}_m}(\hat{\mathbf{x}}_m,\mathbf{y}_m)), \tag{6.2}$$

where $\boldsymbol{\eta}_i = (\eta_{i,n})_{n\in\mathcal{N}_m}$ presents feasible bit allocations satisfying the ith constraint, i.e., $\sum_{n\in\mathcal{N}_m} \eta_{i,n} \leq \eta_{i,\text{sum}}$ while $\eta_{i,n} > 0$ is integer-valued as usual in practical systems. $D_{\text{KL}}(\cdot||\cdot)$

measures the distance between its input distributions, defined as $\mathbb{E}\{\log \frac{p_{\hat{\mathbf{X}}_m,\mathbf{Y}_m}(\hat{\mathbf{x}}_m,\mathbf{y}_m)}{q_{\hat{\mathbf{X}}_m,\mathbf{Y}_m}(\hat{\mathbf{x}}_m,\mathbf{y}_m)}\}$, where $p_{\hat{\mathbf{X}}_m,\mathbf{Y}_m}(\hat{\mathbf{x}}_m,\mathbf{y}_m)$ is the reference distribution over input and output of the mth MLU calculated with $\sum_{n\in\mathcal{N}_m}\eta_{i,n} \gg \max_i \eta_{i,\text{sum}}$. $q_{\hat{\mathbf{X}}_m,\mathbf{Y}_m}(\hat{\mathbf{x}}_m,\mathbf{y}_m)$ is the joint distribution over input and output of the MLU for a given bit allocation $\boldsymbol{\eta}_i$. The empirical estimation of these distributions is discussed in Chapter 3 and 4. A data set for the approximations can be generated by running the system and collecting data, since MLU is fixed and given.

The selected bit allocation $\boldsymbol{\eta}_i^*$ of (6.2) determines quantization noise levels for the terminals transmitting to mth MLU. In this distortion pattern, a lower quantization noise level for a given terminal implies higher relevancy of its data.

Values of $\eta_{i,\text{sum}}$ are selected such that the MLU performance indicator, e_i being the KLD, is better than a threshold and $\eta_{i+1,\text{sum}} - \eta_{i,\text{sum}} = 1$. The threshold can be selected considering the needs of problem under study and heuristics or using elbow method in general. The elbow method runs a clustering approach over a range of values for number of clusters and calculates their corresponding errors. Then, elbow point, i.e., the point at which decrease of error becomes sufficiently small, provides the threshold on error and a corresponding $\eta_{i,\text{sum}}$. The error values e_i can be determined during or after the bit allocations $\boldsymbol{\eta}_i^*$ are selected for all constraints, and they are later used for resource allocation as demonstrated in 6.2.3.

As discussed, the KLD approach requires no prior knowledge on statistics and treats MLU as a black box. It can be applied to all non-adaptive MLUs with fixed parameters, once trained and executing tasks online as inference units in network, independently of their learning paradigm and hypothesis. In this case, the lookup table is designed offline once and remains unchanged. Hence, dealing with its corresponding computations is expensive in theory but feasible in practice.

6.2.3 Resource Allocation Optimization Problem

***Problem Statement* 3:** In this chapter, we aim at defining the QoS such that it takes MLU output into account. Therefore, the resource allocation problem to be solved is formulated as follows.

$$\mathcal{K}^* = \underset{\mathcal{K}}{\operatorname{argmin}} \sum_{m=1}^{M} e_m(\mathcal{K}), \tag{6.3}$$

subject to

$$e_m(\mathcal{K}) \in \mathcal{L}_m, \tag{6.4}$$
$$\mathcal{C}_n \cap \mathcal{C}_{n'\neq n} = \emptyset, \forall n, \tag{6.5}$$
$$\cup_n \mathcal{C}_n \subseteq \{1,\ldots,N_{\text{RB}}\}, \tag{6.6}$$
$$\gamma_{n,r} \leq \gamma_{\max}, \forall n, r, \tag{6.7}$$

where $\mathcal{K}=\{\mathcal{C}_n, n=1,\ldots,N\}$ represents a feasible resource allocation. $e_m(\mathcal{K})$ is the error function outputting KLD values. Its output depends on the resource allocation and the satisfied payload requirement from mth lookup table. This condition is captured in (6.4), and defining $e_m(\mathcal{K})$ is elaborated in the rest of this section. A lookup table per each different MLU should be derived. Similar MLUs use the same lookup table. $\cup, \cap$ and $\emptyset$ indicate the union and intersection operator and empty set. Equation (6.5) implies that only one device is scheduled on each RB. Equation (6.6) ensures that union of allocated RBs is a subset of available RBs. And, the transmission power is selected such that (6.7) holds, where $\gamma_{n,r}$ is the signal-to-noise power ratio at nth source on rth RB.

The error function $e_m(\mathcal{K})$ is defined considering the following aspects. For a given resource allocation $\mathcal{K}$ and mth MLU, if $\boldsymbol{\eta}_i^* \in \mathcal{L}_m$ exists such that

$$\boldsymbol{\eta}_i^* \leq \boldsymbol{\lambda}_m^{\text{achievable}}, \tag{6.8}$$

$e_m(\mathcal{K})$ equals minimum error value of e_i from table rows with $\boldsymbol{\eta}_i^*$ satisfying (6.8). If this condition is not met, $e_m(\mathcal{K})$ is set to a number larger than maximum value of $e_i \in \mathcal{L}_m$. Clearly, output range of error functions should be normalized for all MLU tables to avoid inconsistency which is far from trivial. In particular, some KLD approximation methods like k-Nearest Neighbors (k-NN) do not provide true distribution models. Therefore, they can be used for comparisons in a single setup such as bit allocation of (6.2), but not as error measures when comparing different learning problems. To circumvent this challenge, a greedy algorithm is proposed for solving (6.3) in Subsection 6.3. The algorithm starts with minimizing $e_m(\mathcal{K})$ for a randomly selected MLU and then the same process is repeated for other MLUs. This approach does not guarantee to reach the optimal solution but reduces computations and eliminates the complexity of normalizing error values.

KLD values for $e_m(\mathcal{K})$ can be calculated by running the system with non-quantized data and quantized input of each payload requirement. Thus, no evaluations in application layer is necessary. Furthermore, KLD is used for $e_m(\mathcal{K})$ in this study due to its capability according to [88, 94], and to provide a common framework for comparing performance of KLD and equal sharing lookup tables. If only KLD lookup tables with the same distortion measure are given and employed for resource allocation, error values of lookup tables can be selected arbitrarily with a descending order when $\eta_{i,\text{sum}}$ increases.

Note that in case of having very limited resources, it is possible that no feasible resource allocation satisfying all constraints of (6.4) – (6.7) exists, or such an allocation exists but the heuristic greedy algorithm is incapable of finding it. In such cases, packet drops and missing values for MLU input are assumed. More details about dealing with missing values are provided in Section 6.4.

6.2.4 Case Study 4: Network of Inverted Pendulums on Carts and its KPI

The problem of a single cart inverted pendulum is fully described in Chapter 3. The resource allocation is performed every 0.01 sec based on a rule of thumb for reporting pendulum conditions to the controller. In a network of pendulums, mth MLU input becomes $\mathbf{x}_m = (\mu_p, l_p, \nu, \theta, \dot{\nu}, \dot{\theta})$, and the KPI for performance evaluation is the steady state error probability as elaborated in Subsection 3.2.2.

In our network, the same copy of a trained ML instance is used for all pendulums. Hence, only one lookup table is used and the subscript m can be dropped from $\mathcal{L}_m$ in (6.4). Although the error function of all pendulums are the same, we employ greedy algorithms to solve (6.3), since reducing computations is essential for resource allocation.

6.2.5 Benchmarks

The proposed resource allocation includes two main components to take into account for selecting benchmarks: the lookup table and greedy algorithm to assign resources. In order to study impact of the method generating lookup tables, in addition to the lookup table designed with KLD, a lookup table with equal sharing bit allocations from Chapter 3 is investigated. The reason is that except from the relevance based KLD bit assignment, this conventional bit allocation strategy provided the best outcome in terms of steady state error probability for cart inverted pendulum. For exploring the influence of using the GKLD algorithm, the well-known and practical GMSR is employed as elaborated in the following.

Algorithm 6.1: Step 1 of GKLD

Input: $\{h_{n,r}, \forall n, r\}$, B, $\gamma_{\max}$, $\mathcal{L}_m$
Output: $\mathcal{F}$, $\{\mathcal{C}_n^*, n \notin \mathcal{F}\}$ and $\mathcal{U}$
Initialization: N_{th}, $\mathcal{A} = \emptyset$, $\mathcal{C}_n = \emptyset, \mathcal{F} = \emptyset$

for each MLU or a copy m // MLU selection
do:
  $f_t = 1$, $f_s = 0$ // Keep trying and success flags
  $i = 1$
  $\boldsymbol{\eta}_{\text{target}} = \boldsymbol{\eta}_i^*, \boldsymbol{\eta}_i^* \in \mathcal{L}_m$
    // Starting from the payload requirement with best MLU performance, i.e., lowest $e_i \in \mathcal{L}_m$
  $\mathbf{n}' \leftarrow$ Sorted source indices of $\mathcal{N}_m$ in descending order of their requirement given $\boldsymbol{\eta}_{\text{target}} = (\eta_{\text{target},n})_{n \in \mathcal{N}_m}$
    // Starting the allocation with the most demanding terminal to utilize diversity.
  $j' = 1$
  $n = \mathbf{n}'[j']$ // $\mathbf{n}'[j']$ is the j'th element in $\mathbf{n}'$
  while $\sum_{n \in \mathcal{N}_m} \text{card}(\mathcal{C}_n) \leq N_{\text{th}} \wedge\ f_t == 1$ **do**
    $\mathcal{U}' = \{r, r \in \mathcal{U} \wedge \lambda_{m,n}^{\text{achievable}}(\mathcal{C}_n \cup r) \geq \eta_{\text{target},n}\}$
    if $\mathcal{U}' \neq \emptyset$ **then**
      $r^* = \text{argmin}_{r \in \mathcal{U}'} \lambda_{m,n}^{\text{achievable}}(\mathcal{C}_n \cup r) - \eta_{\text{target},n}$
      $\mathcal{C}_n = \mathcal{C}_n \cup r^*$, $\mathcal{U} = \mathcal{U} \setminus r^*$ // temporary update
      $j' = j' + 1$
      if $j' > \text{card}(\mathcal{N}_m)$ **then**
        $f_s = 1$, $f_t = 0$
        $\mathcal{C}_n^* = \mathcal{C}_n$ for $n \in \mathcal{N}_m$
        Update N_{th} with extra RBs.
      else
        $n = \mathbf{n}'[j']$
    else
      $r^* = \text{argmin}_{r \in \mathcal{U}} \eta_{\text{target},n} - \lambda_{m,n}^{\text{achievable}}(\mathcal{C}_n \cup r)$
      $\mathcal{C}_n = \mathcal{C}_n \cup r^*$, $\mathcal{U} = \mathcal{U} \setminus r^*$ // temporary update
    if $\sum_{n \in \mathcal{N}_m} \text{card}(\mathcal{C}_n) == N_{\text{th}} \wedge\ f_s == 0$ **then**
      $i = i + 1$ // Next best payload
      if $i \leq \max\{i\}$ **then**
        Do step 5–8
        $\mathcal{U} = \cup_{n \in \mathcal{N}_m} \mathcal{C}_n \cup \mathcal{U}$ // Undo temporary updates
        $\mathcal{C}_n = \emptyset$ for $n \in \mathcal{N}_m$
      else
        // Failure in meeting all payload requirements of $\mathcal{L}_m$
        $f_t = 0$
        $\mathcal{F} = \mathcal{F} \cup \mathcal{N}_m$ // Keep source indices of mth MLU in $\mathcal{F}$ for Step 2

The GMSR picks the RB with maximum $|h_{n,r}|^2$ and allocates it for its corresponding device. The process goes on with next RB having best channel quality, while the terminal with already assigned RB is not accounted for allocation until each source gets a RB. Then it is observed whether the selected resource allocation can satisfy any payload requirement from the lookup table. If not, it is individually considered for each terminal, whether any quantization based on available codebooks of the terminal can be performed. If non of

Algorithm 6.2: Step 2 of the GKLD

Input: $\mathcal{U}, \mathcal{F}, \{h_{n,r}, n \in \mathcal{F}, r \in \mathcal{U}\}$, B, $\gamma_{\max}$, $\mathcal{L}_m$
Output: $\{\mathcal{C}_n^*, n \in \mathcal{F}\}$
Initialization: N_{th}, $\mathcal{C}_n = \emptyset$ for $n \in \mathcal{F}$

```
Allocate RBs of U to remaining sources with GMSR to form C_n, n ∈ F
Calculate λ^achievable_{m,n}(C_n), ∀n ∈ F
for each MLU m with N_m ⊆ F do
    if ∃η*_i ∈ L_m, η*_i ≤ λ^achievable_m then
                                                    // On group of sources
        C*_n = C_n, n ∈ N_m
    else
        for n ∈ N_m do
            if ∃η*_{i,n} ∈ L_m, λ^achievable_{m,n}(C_n) ≥ η*_{i,n} then
                                                    // On individual source
                C*_n = C_n
            else
                C*_n = ∅                            // Missing Value
```

/* With more than one option satisfying if conditions, $\boldsymbol{\eta}_i^*$ with lowest $e_i \in \mathcal{L}_m$ and maximum $\eta_{i,n}^*$ are used for the 1st and 2nd condition. */

these conditions are met, missing values occur at input of the MLU which are handled as explained in 6.4. This procedure is elaborated in Algorithm 6.2 and in Section 6.3.

For evaluating our proposed method using the KLD based lookup table and GKLD algorithm, the first benchmark employs the equal sharing lookup table with the GMSR algorithm. The second benchmark uses the KLD lookup tables in combination with GMSR to clarify impact of the GKLD algorithm. To study only the influence of approaches deriving lookup tables, results of benchmark 1 and 2 can be compared.

6.3 The Proposed Resource Allocation Algorithm

In this section, we introduce a heuristic greedy algorithm to solve (6.3) – (6.7). A general overview of the whole proposed resource allocation procedure including the GKLD algorithm is shown in Fig. 6.2. The GKLD algorithm starts with enforcing a primary threshold on number of RBs that can be allocated for transmitting input components of mth MLU. This avoids allocation of all RBs for a few MLUs and further limits power consumption. In network of cart inverted pendulums, same threshold N_{th} is used for all m,

$$\sum_{n \in \mathcal{N}_m} \text{card}(\mathcal{C}_n) \leq N_{\text{th}}, \forall m. \tag{6.9}$$

In a system with different MLUs and given N_{RB}, the threshold can be chosen relative to the average number of required bits to satisfy requirements of lookup tables.

In general, the GKLD algorithm tries to find a resource allocation for all terminals in two steps. **Step 1:** The algorithm picks MLU instances in turn. For each MLU m, the heuristic GKLD starts with the most demanding payload requirement from $\mathcal{L}_m$ which leads to best QoS, and tries to find an allocation fulfilling it while (6.9) holds. In case of success and when less than N_{th} RBs are allocated, the extra RBs are also used for the next MLU. This process is elaborated in Algorithm 6.1. **Step 2:** In case of failing to achieve the goal of step 1 for some MLUs, the opportunistic GMSR allocation is performed over remaining RBs for them.

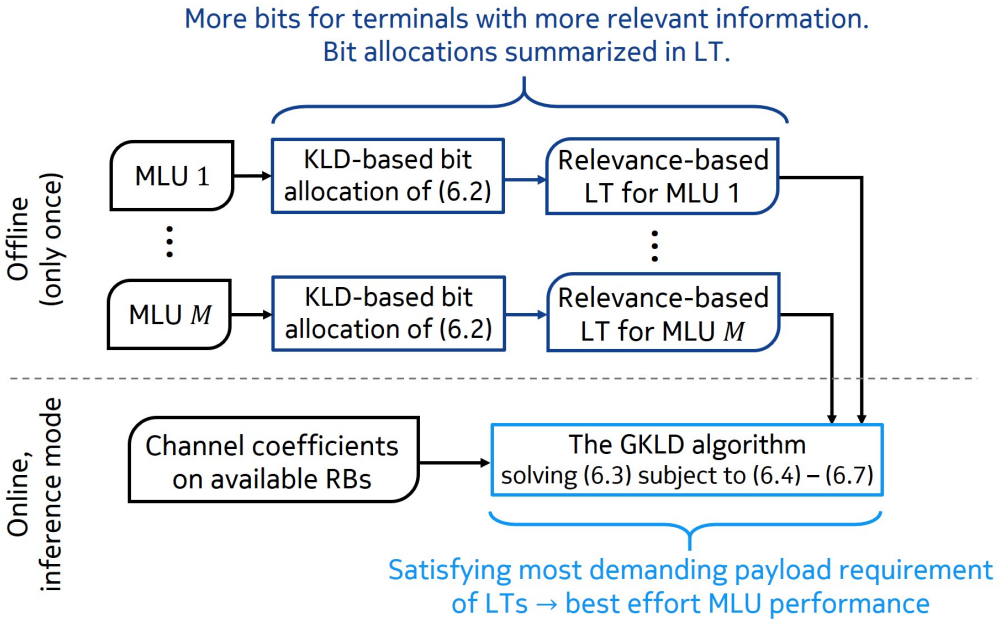


Figure 6.2: A general overview of the proposed resource allocation procedure. LT stands for lookup table.

Afterwards, if the resulting allocations satisfy any payload requirement from lookup tables, quantization and transmission is done for the most demanding one. Otherwise, the devices are treated as independent components and the grouping of terminals serving a particular MLU is not considered. In other words, if the resulting allocations can provide enough number of bits for quantization considering the available codebooks of each individual source, data packet of the terminal is transmitted with highest possible resolution. In case of violating all the aforementioned conditions, $\mathcal{C}_n$ remains empty and a packet drop occurs. In such situations, the missing input attributes are replaced with predefined values at the MLU. This process is presented in Algorithm 6.2.

We assume lookup tables are sorted such that their first entry with $i = 1$ is the one with most demanding bit allocation, i.e., highest $\eta_{i,\text{sum}}$. $\mathcal{A}$ is the set of allocated RBs and $\mathcal{U} = \{1, \ldots, N_{\text{RB}}\} \setminus \mathcal{A}$, where $\setminus$ stands for set subtraction operator. $\mathcal{F}$ is the set of terminal indices with empty $\mathcal{C}_n$ after Algorithm 6.1 is performed, and it is an input for Algorithm 6.2. In our benchmarks which employ GMSR resource allocation, the same process as explained in Algorithm 6.2 is followed for all MLUs and over the whole bandwidth to solve the optimization problem of (6.3) – (6.7).

6.4 Simulation Setup

In this chapter, the MLU trained for Chapter 3 is employed. Similar to the case study 1, since bar mass and length do not change frequently, we assume each of them are transmitted with 10 bits when needed. Hence, $\mathbf{x}_m = (\nu, \theta, \dot{\nu}, \dot{\theta})$ for resource allocation and we deal with four terminals per each MLU, i.e., $N = 4 \times M$. Uniform scalar quantization is performed on data of each terminal. Bit allocations of the KLD lookup table are selected based on calculations with histogram and smoothing presented in Chapter 3. This table has 7 different bit allocation patterns, equivalently rows, built from 10 codebooks in total for all terminals. In other words, for the given MLU, to make these 7 bit allocations, each of the first and second terminals need to be equipped with 2 codebooks, and each of the third and fourth sources are supplied with 3 codebooks. To get analogous QoS using similar number of codebooks, the lookup table with equal payload requirements contains bit allocations with 6, 7, 8 bits for each terminal, e.g., $\boldsymbol{\eta}_1^* = (8, 8, 8, 8)$, requiring 12 codebooks.

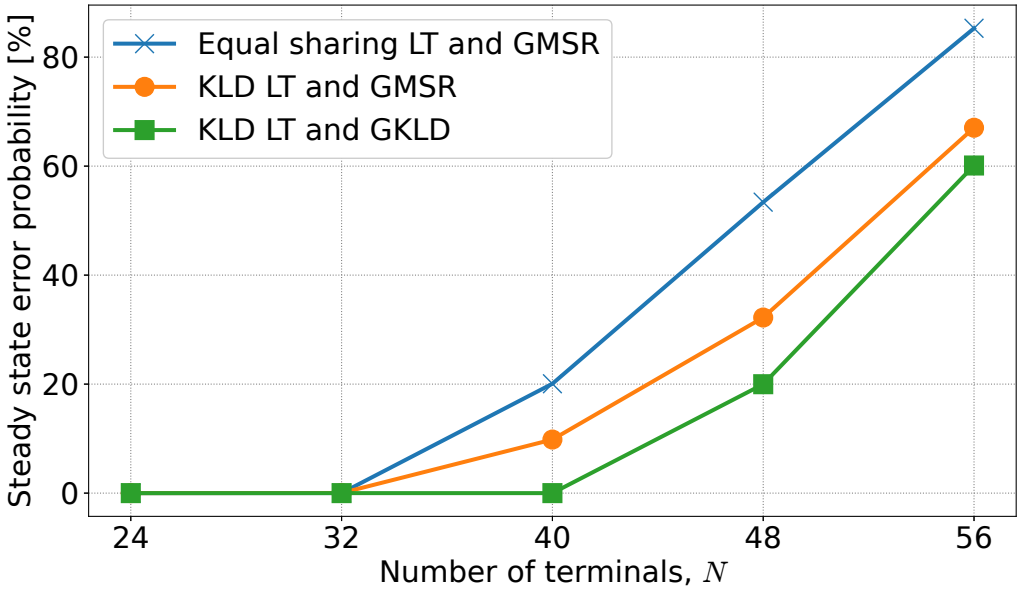


Figure 6.3: Steady state error probability in percentage vs. number of terminals N. $N_{\mathrm{RB}} = 64$ and $\gamma_{\max} = 0\,\mathrm{dB}$.

In GKLD, the order of resource allocation for MLUs changes with a circular shift. Thus, one MLU is not always scheduled first. Initially, $N_{\mathrm{th}} = \lfloor N_{\mathrm{RB}}/N \rfloor$ and increased by one for a fixed subset of MLUs in greedy algorithm until remainder of $\mathrm{N_{RB}}/N$ becomes zero. $\tau = 0.5\,\mathrm{ms}$ and bandwidth of each RB is 10 kHz in this conceptual study. Missing values are replaced with the middle point in interval of each variable in $\mathbf{x}_m$, as if quantization is performed with 0 bits. For evaluations, pendulums are monitored for 1000 iterations while each iteration simulates a period of 2 sec.

6.5 Numerical Results

In this section, steady state error probability is investigated for network of pendulums using three resource allocations. The most conventional benchmark performs the GMSR algorithm to fulfill requirements of the equal sharing lookup table. The second and third approaches employ the KLD lookup table, where allocation is done with the GMSR and proposed GKLD algorithm, respectively. In all the setups under study, the KLD based lookup table with GKLD algorithm outperforms other strategies.

In the first setup, $\gamma_{\max} = 0\,\mathrm{dB}$ and number of RBs is fixed, i.e., $N_{\mathrm{RB}} = 64$. The steady state error probability vs. number of terminals N is depicted in Fig. 6.3. For $M \leq 8$ or equivalently $N \leq 32$, all three resource allocations are capable of providing zero steady state error. However, by increasing the number of sources implying reduced number of RBs for each terminal and scarcer resources, shifting from conventional to relevance based allocations leads to achieving lower error probabilities. For instance with 40 active terminals, the GKLD algorithm using KLD lookup table can still stabilize the system in the predefined 2 sec with no errors. On the other hand, using the GMSR even in combination with KLD lookup table degrades the performance by 10%, since the main goal of the GMSR is to allocate resources opportunistically, and not to particularly satisfy lookup table requirements. Therefore, strong RBs could be assigned to terminals with low payload demands, resulting in more failure to fulfill the requirements on quantization noise patterns of KLD lookup table. This loss becomes 20% by employing the GMSR and equal sharing lookup table. Comparing with the KLD lookup table, this lookup table imposes higher payload requirements for some terminals which are unable to provide much relevant

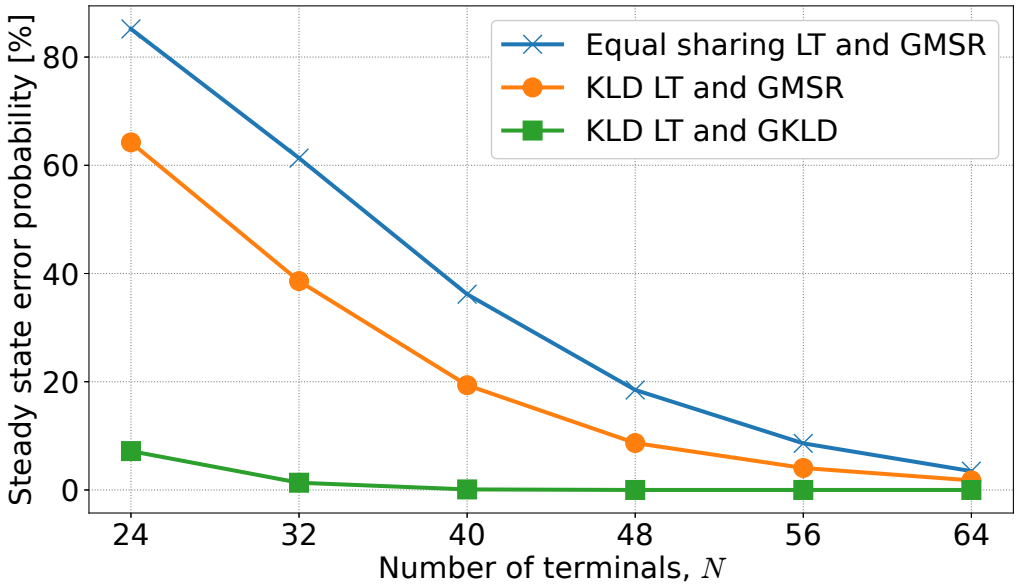


Figure 6.4: Steady state error probability in percentage vs. number of terminals N. $N_{\mathrm{RB}} = 1.5 \times N$ and $\gamma_{\max} = 0\,\mathrm{dB}$.

information for the MLU. Thus, less allocations can cope with requests of this lookup table which increases the occurrence of errors.

Furthermore, in case of targeting zero steady state error probability for the discussed network with 64 RBs, the proposed resource allocation can serve up to 40 terminals, i.e., 8 more terminals compared to other benchmarks. This indicates more efficient resource utilization of the proposed approach.

Fig. 6.4 shows the steady state error probability vs. number of terminals N for $N_{\mathrm{RB}} = 1.5 \times N$ and $\gamma_{\max} = 0\,\mathrm{dB}$. In this scenario, we assume the available bandwidth considered for the network is determined based on the number of terminals. In this case, N_{RB} is selected such that each MLU gets 6 RBs. In other words, 1.5 RBs are allocated for each terminal on average. Note that each terminal provides one input for a MLU. Here, increasing N improves the system performance considering existence of multiple terminals and diversity of channel coefficients, e.g., the proposed approach enhances steady state error probability from $\approx 7.2\%$ to 0 with increasing N from 24 to 40. Similar to the last case, worst results are given by the resource allocation with GMSR and equal sharing lookup table. It can deliver a steady state error probability of $\leq 10\%$ only for $M \geq 56$.

As shown by simulations, the GKLD results deliver significant gains in comparison with other benchmarks. With 32 sources, we gain ≈ 37 and 61% regarding the benchmark using the GMSR with KLD lookup table and the most conventional method, respectively. These gains are smaller in absence of resource scarcity. However, even for $N = 64$, error probability of the proposed approach, GMSR with KLD lookup table and GMSR with equal sharing lookup table are 0, 1.7 and 3.5%, indicating improvements of 1.7 and 3.5% with our scheme. Here, abundance of resources in form of diverse channel gains due to higher number of terminals and N_{RB} compensates for inefficiency of the GMSR and even larger payloads of the equal sharing lookup table.

In Fig. 6.5, steady state error probability vs. $\gamma_{\max}$ is investigated for two scenarios with 8 and 40 terminals, i.e., $M = 2$ and 10. Here, we study a system with even more limited resources assuming that number of terminals is known and fixed. Thus, $N_{\mathrm{RB}} = N$ stating that the available bandwidth is selected to contain 4 RBs per each MLU. In both scenarios, the proposed technique outperforms benchmarks with remarkable gains. For $\gamma_{\max} = 9\,\mathrm{dB}$

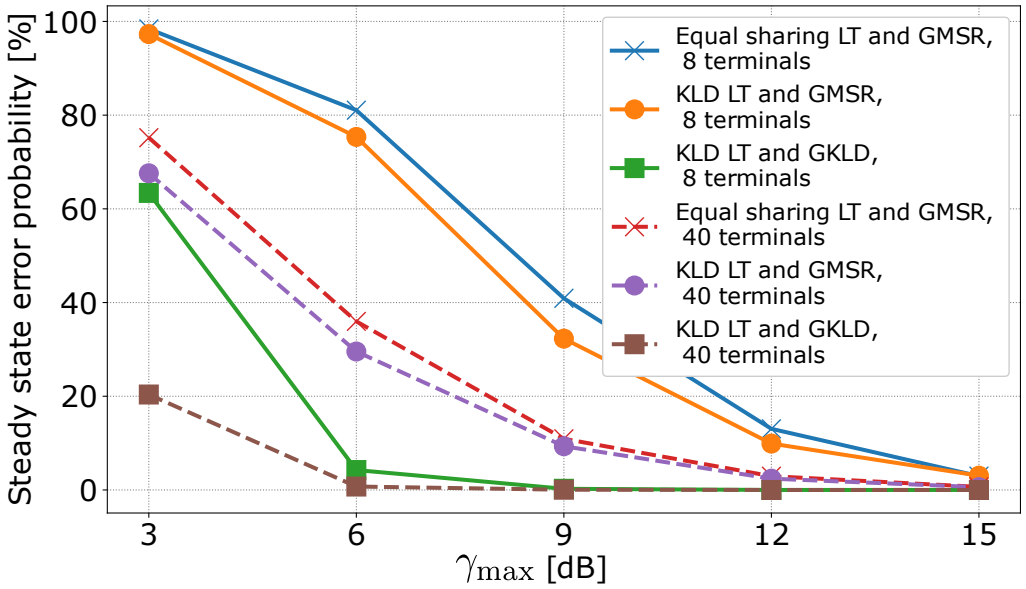


Figure 6.5: Steady state error probability in percentage vs. $\gamma_{\max}$ in dB for different number of terminals N, and $N_{\mathrm{RB}} = N$.

and $N = 8$, error probabilities of $\approx 0.25\%$, 32%, and 41% are achieved using our approach, GMSR and KLD lookup table and GMSR with equal sharing lookup table, respectively. With $N = 40$ assuming same $\gamma_{\max}$, these outcomes enhance to approximately 0.05%, 9.3% and 11%. Similar to last setups, the GMSR allocation with KLD lookup table performs better than the same algorithm using the equal sharing table. However, its gains are insignificant when comparing them with the outcome achieved by the introduced GKLD algorithm using KLD lookup table.

It can be concluded from the results of Fig. 6.5 that the proposed approach is a powerful framework regarding resource utilization. In order to reach a steady state error probability of less than 5%, the relevance based method requires $\gamma_{\max} = 6\,\mathrm{dB}$ to provide 4.2 and 0.73% error probabilities in case of having 8 and 40 terminals. However, reaching the same target calls for $\gamma_{\max} = 12$ and 15 dB by other benchmarks assuming $N = 40$ and 8, respectively. Furthermore, with $\gamma_{\max} = 12$ and $N = 40$, the GMSR plus KLD lookup table and equal sharing lookup table achieve 2.4 and 2.9% error probabilities instead of 0.73% of the GKLD approach. This shows a gain of 6 and 9 dB, depending on the number of devices, delivered by our method yet providing better MLU performance. These gains imply more efficient utilization of resources by our method and are obtained at the cost of extra computations to build lookup tables. However, these calculations are performed once and offline assuming MLUs with fixed parameters.

6.6 Summary and Conclusion

As discussed earlier, we capture the relevancy of MLU input attributes in terms of the bit allocation and the pattern of quantization noise that can be handled at the MLU to provide a general framework. In this chapter, the proposed divergence based foundation is expanded to be utilized in a network with integrated Artificial Intelligence (AI), in order to both deliver better outcome for the MLUs and employ radio resources efficiently.

For this purpose, a novel relevance based resource allocation for a ML based centralized control system is proposed. The scheme consists of designing KLD based lookup tables to account for relevancy of input information for each MLU, defining the optimization problem with new QoS metric and introducing a greedy algorithm to solve it. As shown

by simulations over network of cart inverted pendulums, using the proposed approach yields remarkable gains in terms of MLU performance and resource utilization comparing with conventional methods, particularly in case of having limited resources. In addition, the extra computation for deriving lookup tables remains feasible to be done in practice, since MLUs of the network are assumed to be non-adaptive while executing their tasks in inference mode.

These results and the significant gains motivate the study of relevance based methods for various scenarios. For instance, although the proposed approach can be applied to a wide range of problems, only a network of cart inverted pendulums is explored while employing this method. Hence, a future line of research is to apply the method to a network of heterogeneous MLUs.

chapterConclusion and Outlook

In this chapter, a brief summary of the studies provided in this dissertation, and the main conclusions regarding our results are presented in Section 6.7. Based on our observations and remarkable gains achieved by the proposed relevance based solutions, several fields and study items are brought to attention for future investigations in Section 6.8.

6.7 Summary and Conclusions

In this dissertation, the gap in providing efficient support for Machine Learning Based Units (MLUs) with communications resources is addressed. The main idea is to circumvent syntax and focus on the semantics of input data for given MLUs in inference mode. For this purpose, multiple terminals transmitting multivariate and correlated input data for MLUs are considered. The MLUs operating in the network are treated as black boxes, while the only assumption to hold is that MLU parameters such as weights are trained, fixed and do not change. This system model under study, along with avoiding any prior assumption on statistics, guarantees the applicability of our proposed framework to many learning related use cases. This framework quantifies the relevancy of MLU input components in terms of bit allocation aspect of data quantization.

The proposed divergence based bit allocation strategy, using any of the introduced distortion measures, finds the patterns of quantization noise that can be tolerated by given Machine Learning (ML) based entities, which are afterwards translated into gains in the utilization of network radio resources. Moreover, it offers the degree of freedom for further modifications and expansions that adjust it as a solution for other network related problems, e.g., signaling overhead. In order to evaluate the proposed relevance based framework, various case studies with or without feedback, employing real or simulator data, different ML models and codebook designs are investigated. Dependency of achieved gains with our approach on many parameters is discussed, while the complexity level of the problem, ML algorithm, Kullback-Leibler Divergence (KLD) estimation and codebook design are determined as contributing factors impacting these benefits.

In all of the studied scenarios, the proposed relevance based approaches deliver the best performance in terms of the corresponding Key Performance Indicator (KPI)s of each case study. In many of these cases, the best performance implies significant gains, e.g., more than 50% signaling overhead reduction for the Artificial Intelligence (AI) assisted conditional handover preparation. Note that having limited communications resources like bandwidth, or the goal of employing the least amount of them is a key point with respect to these gains. Based on our observations, more enhancements occur in case of having scarce resources in the system. Otherwise, even the less efficient benchmarks can provide a given target performance in presence of abundant resources. The advantages of using the divergence based method remain valid when Packet Drop (PD)s exist. This

indicates more compressed inputs chosen by the KLD allocation do not cause additional sensitivity to this imperfection.

Regarding the relevancy level of each input component, the KLD based bit allocation generally picks patterns with nonuniform distribution of quantization noise. However, an exclusive relevancy only for a subset of input attributes is not observed. In other words, we cope with low and high levels of relevancy rather than not relevant and relevant input components. This conclusion is expected, since meaningful attributes are typically chosen as input components of MLUs, and MLUs learn to extract their knowledge considering all of these features.

It is worth mentioning that although the proposed methods of this dissertation call for empirical distribution and KLD estimations, these extra calculations do not impose an obstacle to employing our framework in practice. Performing these computations is specifically simple when dealing with classification problems allowing for utilization of the k-Nearest Neighbors (k-NN) estimator. More importantly, these approximations need to be carried out only once and in offline mode, making them feasible to be done in real scenarios. In addition, computational complexity can be mitigated by using several techniques such as restricting the search space based on the requirements of the case under study. In the upcoming section, this domain and more future directions of research are discussed.

6.8 Outlook and Future Directions

In this section, several potential areas are introduced for future research on radio resource management for ML based entities. Some examples of the following guideline are directly related to the presented studies in this dissertation. And, some topics require further analysis and supplement of new aspects and ideas. Since this subject and its state of the art are at a primary stage, future directions are certainly not limited to this presented guideline and its instances.

As it can be seen from the summary of Section 6.7, one core objective fulfilled in this dissertation is to prevent studying oversimplified systems. Therefore, the introduced approaches are applicable in practice and construct a generic framework that can be utilized at least as a first step towards tackling many real-world problems. Since our proposals deal with a fundamental aspect, i.e., the number of bits in data compression and the rate-distortion tradeoff, they have the potential to be slightly reshaped or extended for more specific use cases, as it is done to the KLD based bit allocation in chapters 5 and 6. Considering the achieved outcome of our relevance based strategies, we motivate modifications leading to realization of specifically tailored solutions for various use cases. Such specialized designs are likely to deliver even higher gains and improved KPIs. Naturally, these extensions need keen observations and may not seem straightforward at a first glance for some use cases.

In our case studies, MLUs are not particularly trained to handle missing values at their input ports. Hence, considering such MLUs and exploring KLD based allocations for these entities can be a future study item, in order to find out whether a meaningful change in achieved gains occurs in these scenarios. In general, applying the proposed bit allocation strategy to various experiments is encouraged. This can lead to gaining more insight into the influence of different system components on relevance based gains.

Employing more efficient search algorithms and developing novel ones for finding bit allocations minimizing KLD, and investigating the impact of using more sophisticated methods for distribution and KLD estimations are among other areas to study regarding the proposed framework of this dissertation. These studies reduce computational complexity and

potentially assist the development of methods that are capable of working with non-fixed MLUs.

Another future line of research is studying and applying the proposed wireless resource allocation to heterogeneous networks, i.e., a network containing various use cases with different and asynchronous service requirements and MLUs. In addition, retraining of MLUs with selected bit allocations of the KLD based method can be taken into account. However, as already discussed, if MLUs are provided by third-party vendors, such modifications might be impossible. Nevertheless, the outcome of these explorations can be insightful, and making such changes can be allowed in specific situations. In these particular cases, joint optimization of bit allocation, codebook and MLU design can also be considered.

Extending the current framework or providing novel approaches to quantify the relevancy of data and using it for efficient utilization of communications resources for adaptive MLUs that adjust themselves to their environment is another interesting subject to be explored. Finding an efficient solution for such scenarios can also assist in modifying or replacing the lookup tables used in the proposed wireless resource allocation with more precise and up-to-date information about the MLU requirements. In a further step, an enhanced relevance based approach can adapt its decisions for picking a bit allocation that results in a best effort performance by taking instantaneous channel state information into account.

Finally, a shift from syntax to relevance or semantic based communications for networks of MLUs, however, non-trivial, shows considerable benefits. These advantages facilitate the implementation of future applications that function with connected devices. Therefore, the research on data relevancy from different perspectives, yielding gains in terms of wireless resources, provides us with novel and effective solutions for future radio resource and network management. A few related instances for further studying of this subject are quantifying the relevancy of data in the time domain, and partitioning of input space combined with applying the bit allocation for each cluster to equip the current strategy with more degrees of freedom.

Appendices

A. Extra Simulations on Impact of Bit Allocation Strategies on MLUs

In the following, we provide the numerical results for two more scenarios in addition to the cart inverted pendulum which is the main sandbox in Chapter 3. These cases are investigated to provide further proof on applicability and advantages of using the proposed relevance based method using KLD.

A.1 Moon Data Set

In this section, a toy data set presented in scikit-learn to perform classification tasks is studied. This data set consist of 100 samples, each with two attributes as shown in Fig. A.1. The range of input components are similar. Additive Gaussian noise with standard deviation of 0.3 is applied to input samples of the training and test sets. It is assumed that each input attribute is compressed with uniform scalar quantization and transmitted from a different terminal.

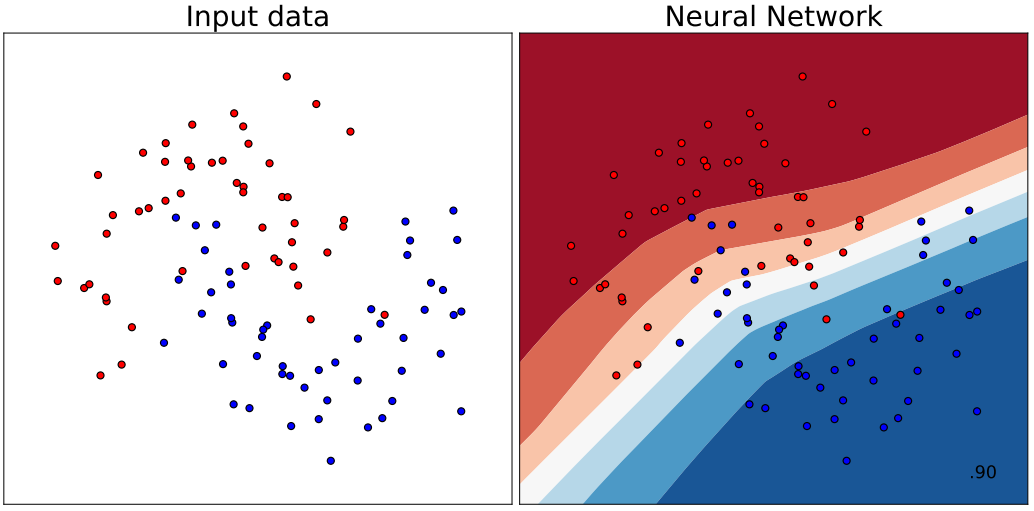


Figure A.1: Moon data set and Neural Network (NN) Classifier with non-quantized data.

For the classification, a shallow NN with 100 neurons is employed and its outcome is sketched in Fig. A.1. The NN is trained with L2 penalty parameter being 1 while maximum

	Selected bit allocation	Classification Accuracy (%)
The proposed KLD	4, 3 bits	92.5
MSE (benchmark)	7, 7 bits	90

Table A.1: Moon data set results.

	Selected bit allocation	Steady state error probability (%)
The proposed KLD	2, 3 bits	0
MSE (benchmark 1)	4, 1 bits	16.2
Equal Sharing (benchmark 2)	2, 2 bits	1.6

Table A.2: Results for simulations with different inverted pendulum setup, quantizing bar mass and length.

number of iterations is set to 1000. The data set and more details are available in [95,117]. In order to estimate KLD, k-NN for classification as described in 4.4.2 is utilized.

Here, the KPI for measuring system performance is the classification accuracy. We assume $2 \leq \eta_n \leq 7$ to determine the feasible set $\mathcal{H}$. Accordingly, the results shown in Table A.1 are achieved. The proposed KLD selects a bit allocation that results in both 2.5% gain in classification accuracy and 50% gain in number of used bits comparing with the bit allocation selected by Mean Squared Error (MSE). As it can be observed, since the range of values for both input attributes are similar, 7 bits is assigned for each of them by MSE approach. Clearly, MSE utilizes the maximum amount of available resources, i.e., bits in order to get the minimum distortion. However, this unnecessary over-utilization of resources does not even lead to an improvement in classification accuracy when compared with the bit allocation selected by the KLD approach.

In this scenario, the KLD is capable of finding a pattern of distortion for input components that works in compliance with MLU. This bit allocation delivers meaningful and relevant information that the MLU requires to make more correct decisions. Hence, the classification of 92.5% instead of 90% of the MSE selection is achieved while only 7 bits are used for data quantization of both input components. This implies a waste of resources in case of employing MSE based approach that uses all 14 available bits.

A.2 Inverted Pendulum with Different Setup

In Chapter 3, it was assumed that bar mass an length do no change frequently and are quantized with high precision. Therefore, the input components of the Machine Learning Based Controller (MLC) requiring quantization were angle, position and their derivatives. Here, the reverse scenario is investigated in which we quantize bar mass an length values, μ_p and l_p, while other input components are delivered with high precision to the MSE. To this end, $\mathcal{H}$ is defined for $1 \leq \eta_n \leq 7$ and firstly, $\eta_{\text{sum}} = 5$ is assumed. In order to estimate KLD, the histogram approach is used.

The simulation results for this different setting are shown in Table A.2. As it can be seen, the KLD approach picks a bit allocation which results in achieving zero steady state errors

$P_e = 0\%$. For the same case, MSE chooses a bit allocation to decrease the quantization noise on μ_p which has a larger interval; however, the controller sensitivity to changes in l_p is higher. Hence, the MSE selection results in a degradation of 16.2% in performance. Equal sharing allocates 2 bits instead of just 1 bit for l and thus, the performance loss becomes 1.6%.

For $\eta_{\text{sum}} \geq 5$, all selected bit allocations of the three methods can achieve steady state error probability of 0%. Similar to the main case study, the KLD approach is particularly beneficial when resources are restricted or we aim at increasing the efficiency with regard to resource usage.

List of Acronyms

k-NN	k-Nearest Neighbors.
3GPP	3rd Generation Partnership Project.
5G	5^{th} Generation of Mobile Network.
6G	6^{th} Generation of Mobile Network.
AI	Artificial Intelligence.
AI-CHO	AI-Assisted Conditional Handover.
CEO	Chief Executive Officer.
CHO	Conditional Handover.
CNN	Convolutional Neural Network.
CSI	Channel State Information.
CTF	Channel Transfer Function.
DNN	Deep Neural Network.
FCF	Frequency Coherence Function.
FNR	False Negative Rate.
FPR	False Positive Rate.
GKLD	Greedy Resource Allocation Algorithm for KLD Based Lookup Table.
GMSR	Greedy Maximum Sum Rate.
HO	Handover.
i.i.d.	Independent and Identically Distributed.
IIoT	Industrial Internet of Things.

IoT	Internet of Things.
KLD	Kullback-Leibler Divergence.
KPI	Key Performance Indicator.
LIDAR	Light Detection and Ranging.
LQR	Linear-Quadratic Regulator Controller.
MIMO	Multiple-Input Multiple-Output.
MISO	Multiple-Intput Single-Output.
ML	Machine Learning.
MLC	Machine Learning Based Controller.
MLU	Machine Learning Based Unit.
MMSE	Minimum Mean Square Error.
mmWave	Millimeter-Wave.
MSE	Mean Squared Error.
NN	Neural Network.
OFDM	Orthogonal Frequency-Division Multiplexing.
PD	Packet Drop.
PP	Ping Pong.
QoS	Quality of Service.
QPSK	Quadrature Phase Shift Keying.
RB	Resource Block.
ReLU	Rectified Linear Unit.
RIS	Reconfigurable Intelligent Surface.
RL	Reinforcement Learning.
RLF	Radio Link Failure.
RNN	Recurrent Neural Network.
RSRP	Reference Signal Received Power.

SCHO	Successful Conditional Handover.
SGD	Stochastic Gradient Descent.
SINR	Signal-to-Noise-plus-Interference Ratio.
SNR	Signal-to-Noise Ratio.
SOR	Signal Overhead Reduction.
SSE	Sum of Squared Errors.
SVM	Support Vector Machine.
UE	User Equipment.
VC	Vapnik-Chervonenkis.

List of Symbols

$(\cdot)_m$	MLU index, $m \in \{1, \cdots, M\}$.
$(\cdot)_n$	Source index, $n \in 1, \cdots, N$.
$(\cdot)_r$	RB index, $r \in \{1, \ldots, N_{\mathrm{RB}}\}$.
B	Available bandwidth.
B_n	Total bandwidth allocated to source n.
C_B	Capacity of bandlimited channel given B in bits/sec.
$E_{\mathrm{in}}(g)$	In-sample error for hypothesis g.
$E_{\mathrm{out}}(g)$	Out-of-sample error for hypothesis g.
E_{val}	An estimation of out-of-sample error using validation samples.
E_b	Energy per bit.
$H(f)$	Channel transfer function at frequency f.
I	The moment of inertia for bar mass.
$J_n(\eta_n)$	The objectibve of kmeans at the nth source.
K'	The window length for determining whether a cell belongs to the group of cells providing strong or weak RSRP values.
L	Predefined threshold for CHO classifier decisions.
N_0	Noise power spectral density.
N_{N}	Number of negative samples in training set.
N_{P}	Number of positive samples in training set.
N_{RB}	The number of available RBs in network.
N_{TN}	Number of true negatives.

N_{TP}	Number of true positives.
N_{th}	Primary threshold on number of RBs that can be allocated for transmitting input components of mth MLU.
N_c	Number of surronding cells that each UE collects their RSRP values, and number of rows in input matrix of CHO classifier.
N_u	Number of UEs collecting RSRP information in AI-CHO case study.
P_{e}	Steady state error probability in cart inverted pendulum problem.
P_n	Probability of packet drop for source n.
$P_s(t)$	RSRP values of serving cell at a given time t.
$P_t(t)$	RSRP values of target cell at a given time t.
Q_{out}	Outage threshhold in AI-CHO problem.
$R(f)$	Frequency coherence function at frequency f.
R_{LQR}	A parameter in LQR controller.
$R_p(\mathbf{z}_j)$	The euclidean distance between $\mathbf{z}_j$ and its kth neighbor in a given data set related to distribution p.
$R_q(\mathbf{z}_j)$	The euclidean distance between $\mathbf{z}_j$ and its kth neighbor in a given data set related to distribution q.
$S_{\mathcal{G}}(J)$	Shatter coefficient or growth function of $\mathcal{G}$ and for J data points.
S_{21}	Scattering parameters.
T	The time constraint for delivering MLU input attributes.
T_b	Bit interval.
T_s	Symbol interval.
T_{TTT}	Time to trigger during which the RSRP condition must hold before a CHO execution occurs.
T_{pp}	Ping-pong timer.
$\Gamma(\cdot)$	The gamma function.
α	Smoothing parameter.
$\mathbf{x}_m$	Input data of the mth MLU, when multiple MLUs are studied.

$\mathbf{x}_{LQR}$	Input vector of the LQR controller in cart inverted pendulum problem.
$\mathbf{x}_{m,n}$	Data of the nth terminal to be delivered to the mth MLU, when multiple MLUs are studied.
$\mathbf{z}_j = [\hat{\mathbf{x}}_j, \mathbf{y}_j]$	jth data set sample for estimation of KLD.
$\boldsymbol{\eta}$	A feasibe bit allocation with η_n bits to quantize data of nth source, when a single MLU is studied, $\boldsymbol{\eta} = \{\eta_n\}$.
$\boldsymbol{\eta}_i$	A feasible bit allocation satisfying the ith constraint of a lookup table.
$\boldsymbol{\eta}_i^*$	ith entry of a lookup table and ith payload requirement, $\boldsymbol{\eta}_i^* = (\eta_{i,n}^*)_{n\in\mathcal{N}_m}$.
$\boldsymbol{\lambda}_m^{\text{achievable}}$	Sequence of achievable payloads for input of MLU m, $\boldsymbol{\lambda}_m^{\text{achievable}} = \left(\lambda_{m,n}^{\text{achievable}}(\mathcal{C}_n)\right)_{n\in\mathcal{N}_m}$.
$\boldsymbol{\mu}_{i,n}$	The d_n dimensional codeword of cluster i at nth terminal.
$\boldsymbol{\omega}$	Set of all NN weights, $\boldsymbol{\omega} = \{\boldsymbol{\omega}^{(l_{\text{NN}})}\}$.
$\boldsymbol{\omega}^{(l_{\text{NN}})}$	NN wights of l_{NN}th layer.
η_{lr}	Learning rate in gradient descent and SGD algorithms.
η_{sum}	Total number of bits for quantizing input data of a given MLU.
η_n	The number of bits to quantize data of nth source.
$\eta_{i,\text{sum}}$	Total number of quantization bits used as the bit allocation constraint for ith row of a lookup table.
$\eta_{i,n}^*$	The number of bits from ith row of a lookup table for quantizing data of nth source.
γ	Signal-to-noise-ratio, i.e., $\gamma = \frac{E_b}{N_0 B T_b}$.
$\gamma_{\max}$	Maximum signal-to-noise power ratio for $\gamma_{n,r}$.
γ_n	Signal-to-noise-ratio of nth source, when a single MLU is studied.
$\gamma_{n,r}$	Signal-to-noise power ratio for terminal n on rth RB.
$\hat{\mathbf{x}}$	Quantized version of $\mathbf{x}$.
$\hat{p}(\mathbf{z}_j)$	Distribution estimation for p with data set samples $\mathbf{z}_j$.
$\hat{q}(\mathbf{z}_j)$	Distribution estimation for q with data set samples $\mathbf{z}_j$.

$\hat{x}_n$	Quantied version of x_n.
$\lambda_{m,n}^{\text{achievable}}(\mathcal{C}_n)$	Achievable payload for terminal n transmitting to MLU m.
$\mathbb{E}\{\cdot\}$	Expectation operator.
$\mathbf{K}$	Matrix of controller coefficients in cart inverted pendulum problem.
$\mathbf{Q}$	Matrix of controller parameters to balance tradeoff between error and control effort in cart inverted pendulum problem.
$\mathbf{S}_n$	A feasible clustering at source n, where $\mathbf{S}_n = \{S_{i,n}\}$ and $S_{i,n}$ is the ith cluster of the n source.
$\mathbf{X}_{N_c \times K}$	Input matrix of the CHO classifier.
$\mathbf{u}$	A parameter used in LQR cost function and defined as $-\mathbf{K}\mathbf{x}_{\text{LQR}}$.
$\mathbf{x}$	Vector of MLU input attributes, when a single MLU is studied.
$\mathbf{x}_n, n = 1, \cdots, N$	A vector of attributes transmitted from terminal n to the single MLU in network as part of its input components.
$\mathbf{y}$	Sequence of output components of a given MLU.
$\mathbf{y}_m$	Output of the mth MLU, when multiple MLUs are studied.
$\mathcal{A}$	Set of allocated RBs.
$\mathcal{C}_n$	The set of RB indices allocated for nth source.
$\mathcal{D}$	An available set of J nosie-free input and target samples $\big(\mathbf{x}_j, g_t(\mathbf{x}_j)\big)$ for supervised learning.
$\mathcal{F}$	Set of source indices Algorithm 6.1 fails to meet their requirements.
$\mathcal{G}(\mathbf{x}_1, \cdots, \mathbf{x}_J)$	Set of different label sequences or dichotomies that can be built by $\mathcal{G}$ given J data points in binary classification.
$\mathcal{G}$	A hypothesis set.
$\mathcal{H}$	Set of all feaisble bit allocations.
$\mathcal{K}$	A feasible resource allocation.
$\mathcal{L}_m$	The lookup table of mth MLU.
$\mathcal{N}_m$	A subset of $\{1, \cdots, N\}$ with source indices providing input of mth MLU.

$\mathcal{T}_n$	Data sets for distribution estimations.
$\mathcal{U}$	Set of unassigned RBs.
μ_{bin}	The number of bins in support of distribution p with zero samples from $\mathcal{T}_2$ representing distribution q.
μ_c	Cart mass.
μ_p	Pendulum mass.
ν	The position of a cart in cart inverted pendulum problem.
$\omega_{io}^{(l_{\text{NN}})}$	The NN weight associated to the link connecting ith neuron of layer $(l_{\text{NN}}-1)$ to oth node of (l_{NN})th layer.
ϕ	The activation function of the NN.
σ_n^2	Quantization error variance, the MSE between nth input feature and its quantized version.
τ	Length of each RB in time domain.
θ	The angle for a pendulum measured from inverted equilibrium position.
b	The coefficient of friction for cart in cart inverted pendulum problem.
c	A parameter used in steady state equations of cart inverted pendulum which equals $(\mu_c+\mu_p)I+\mu_p\mu_c l_p^2$.
d	Dimension of multivariate vectors over which a distribution is estimated.
$d^{(l_{\text{NN}})}$	Number of neurons in (l_{NN})th layer of a NN.
$d_{\text{rel}}(\cdot)$	A relevance based distortion measure.
$d_{\text{vc}}(\mathcal{G})$	VC dimension of $\mathcal{G}$.
$d_n, n=1,\cdots,N$	Dimension of nth source vector $\mathbf{x}_n$.
$e(\boldsymbol{\omega})$	Error on each sample of training set.
e_i	MLU performance indicator for ith row of a lookup table.
$e_m(\mathcal{K})$	Error function of the mth MLU.
f_c	The force applied to a cart in horizontal direction.

g	A hypothesis in $\mathcal{G}$.
g^*	The final hypothesis selected by the learning algorithm from a hypothesis set $\mathcal{G}$.
g_{G}	Standard gravity.
g_{t}	An unknown target function from $\mathcal{X}$ to $\mathcal{T}$ for MLU to learn.
$h_{n,r}$	Channel coefficient for terminal n on rth RB.
k	kNN parameter for the number of neighbors.
l_p	Pendulum length.
n_{bin}	The number of samples in the histogram bin corresponding to a given sample $\mathbf{z}_j$.
$o^{\mathrm{exec}}_{c_s,c_t}$	CHO execution offset defined between the serving and target cells.
p	$p_{\hat{\mathbf{X}},\mathbf{Y}}(\hat{\mathbf{x}},\mathbf{y})$.
$p_{\hat{X}_n\|X_n}(\hat{x}_n\|x_n)$	The quantization over data of nth source.
$p_{\hat{\mathbf{X}},\mathbf{Y}}(\hat{\mathbf{x}},\mathbf{y})$	Joint input-output distribution of the MLU assuming a highly accurate quantization, when a single MLU is studied. Also shown simply with $p_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})$.
$p_{\hat{\mathbf{X}}_m,\mathbf{Y}_m}(\hat{\mathbf{x}}_m,\mathbf{y}_m)$	Joint input-output distribution of the mth MLU assuming a highly accurate quantization, when multiple MLUs are studied.
q	$q_{\hat{\mathbf{X}},\mathbf{Y}}(\hat{\mathbf{x}},\mathbf{y})$.
$q_{\hat{\mathbf{X}},\mathbf{Y}}(\hat{\mathbf{x}},\mathbf{y})$	Joint MLU input-output distribution for a given bit allocation, when a single MLU is studied.
$q_{\hat{\mathbf{X}}_m,\mathbf{Y}_m}(\hat{\mathbf{x}}_m,\mathbf{y}_m)$	Joint input-output distribution for mth MLU and a given bit allocation, when multiple MLUs are studied.
$s_o^{(l_{\mathrm{NN}})}$	The input signal for oth activation function in (l_{NN})th layer.
t_{GD}	t_{GD}th step of gradient descent and SGD algorithms.
t_s	Time step for simulations in AI-CHO problem.
t_{exec}	CHO execution time.
$v(\mathbf{z}_j)$	The volume of a d-dimensional ball with radius $R_p(\mathbf{z}_j)$.
$w_{c\kappa}$	Weights of the regression model, equivalently, SOR classifier parameters to be learned.

$x_i^{(l_{\mathrm{NN}}-1)}$	The output of ith neuron in $(l_{\mathrm{NN}}-1)$th layer, which is also seen as input for the (l_{NN})th layer under study.
x_n	Univariate version of $\mathbf{x}_n$.
$x_o^{(l_{\mathrm{NN}})}$	The output of oth activation function in $(l_{\mathrm{NN}})$the layer under study.
$x_{c\kappa}$	The element in cth row and κth column of $\mathbf{X}_{N_c \times K}$.
y	Univariate version of $\mathbf{y}$.

List of Figures

List of Tables

Literature

[1] Wei Jiang, Bin Han, Mohammad Asif Habibi, and Hans Dieter Schotten. The road towards 6G: A comprehensive survey. *IEEE Open Journal of the Communications Society*, 2:334–366, 2021. doi:10.1109/OJCOMS.2021.3057679.

[2] Syed Junaid Nawaz, Shree Krishna Sharma, Shurjeel Wyne, Mohammad N. Patwary, and Md. Asaduzzaman. Quantum machine learning for 6G communication networks: State-of-the-art and vision for the future. *IEEE Access*, 7:46317–46350, 2019. doi:10.1109/ACCESS.2019.2909490.

[3] Khaled B. Letaief, Wei Chen, Yuanming Shi, Jun Zhang, and Ying-Jun Angela Zhang. The roadmap to 6G: Ai empowered wireless networks. *IEEE Communications Magazine*, 57(8):84–90, 2019. doi:10.1109/MCOM.2019.1900271.

[4] Baiqing Zong, Chen Fan, Xiyu Wang, Xiangyang Duan, Baojie Wang, and Jianwei Wang. 6G technologies: Key drivers, core requirements, system architectures, and enabling technologies. *IEEE Vehicular Technology Magazine*, 14(3):18–27, 2019. doi:10.1109/MVT.2019.2921398.

[5] Emilio Calvanese Strinati, Sergio Barbarossa, Jose Luis Gonzalez-Jimenez, Dimitri Ktenas, Nicolas Cassiau, Luc Maret, and Cedric Dehos. 6G: The next frontier: From holographic messaging to artificial intelligence using subterahertz and visible light communication. *IEEE Vehicular Technology Magazine*, 14(3):42–50, 2019. doi:10.1109/MVT.2019.2921162.

[6] Harish Viswanathan and Preben E. Mogensen. Communications in the 6G era. *IEEE Access*, 8:57063–57074, 2020. doi:10.1109/ACCESS.2020.2981745.

[7] Shunqing Zhang, Chenlu Xiang, and Shugong Xu. 6G: Connecting everything by 1000 times price reduction. *IEEE Open Journal of Vehicular Technology*, 1:107–115, 2020. doi:10.1109/OJVT.2020.2980003.

[8] Nei Kato, Bomin Mao, Fengxiao Tang, Yuichi Kawamoto, and Jiajia Liu. Ten challenges in advancing machine learning technologies toward 6G. *IEEE Wireless Communications*, 27(3):96–103, 2020. doi:10.1109/MWC.001.1900476.

[9] Gino Masini, Yin Gao, and Sasha Sirotkin. Artificial intelligence and machine learning. https://www.3gpp.org/news-events/2201-ai_ml_r3, 2021.

[10] https://www.3gpp.org/release18.

[11] Wei Jiang and Hans Dieter Schotten. Multi-antenna fading channel prediction empowered by artificial intelligence. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pages 1–6, 2018. doi:10.1109/VTCFall.2018.8690550.

[12] Wei Jiang and Hans Dieter Schotten. Deep learning for fading channel prediction. *IEEE Open Journal of the Communications Society*, 1:320–332, 2020. doi:10.1109/OJCOMS.2020.2982513.

[13] Wei Jiang and Hans D. Schotten. Neural network-based fading channel prediction: A comprehensive overview. *IEEE Access*, 7:118112–118124, 2019. doi:10.1109/ACCESS.2019.2937588.

[14] Wei Jiang and Hans D. Schotten. Recurrent neural networks with long short-term memory for fading channel prediction. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5, 2020. doi:10.1109/VTC2020-Spring48590.2020.9128426.

[15] Hongji Huang, Yiwei Song, Jie Yang, Guan Gui, and Fumiyuki Adachi. Deep-learning-based millimeter-wave massive MIMO for hybrid precoding. *IEEE Transactions on Vehicular Technology*, 68(3):3027–3032, 2019. doi:10.1109/TVT.2019.2893928.

[16] Jun Du, Chunxiao Jiang, Jian Wang, Yong Ren, and Merouane Debbah. Machine learning for 6G wireless networks: Carrying forward enhanced bandwidth, massive access, and ultrareliable/low-latency service. *IEEE Vehicular Technology Magazine*, 15(4):122–134, 2020. doi:10.1109/MVT.2020.3019650.

[17] Wei Jiang, Mathias Strufe, and Hans D. Schotten. Experimental results for artificial intelligence-based self-organized 5G networks. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6, 2017. doi:10.1109/PIMRC.2017.8292532.

[18] Wei Jiang, Mathias Strufe, and Hans D. Schotten. Intelligent network management for 5G systems: The SELFNET approach. In *2017 European Conference on Networks and Communications (EuCNC)*, pages 1–5, 2017. doi:10.1109/EuCNC.2017.7980672.

[19] Wei Jiang, Mathias Strufe, and Hans Schotten. Autonomic network management for software-defined and virtualized 5G systems. In *European Wireless 2017; 23th European Wireless Conference*, pages 1–6, 2017.

[20] Fengxiao Tang, Yuichi Kawamoto, Nei Kato, and Jiajia Liu. Future intelligent and secure vehicular network toward 6G: Machine-learning approaches. *Proceedings of the IEEE*, 108(2):292–307, 2020. doi:10.1109/JPROC.2019.2954595.

[21] Jing Zhang and Dacheng Tao. Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal*, 8(10):7789–7817, 2021. doi:10.1109/JIOT.2020.3039359.

[22] Firuz Kamalov. Sensitivity analysis for feature selection. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1466–1470, 2018. doi:10.1109/ICMLA.2018.00238.

[23] Toshimitsu Homma and Andrea Saltelli. Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering and System Safety*, 52(1):1–17, 1996. doi:https://doi.org/10.1016/0951-8320(96)00002-6.

[24] W.W.Y. Ng, D.S. Yeung, and I. Cloete. Input sample selection for RBF neural network classification problems using sensitivity measure. In *SMC'03 Conference*

Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483), volume 3, pages 2593–2598 vol.3, 2003. doi:10.1109/ICSMC.2003.1244274.

[25] W.W.Y. Ng, D.S. Yeung, Xi-Zhao Wang, and I. Cloete. A study of the difference between partial derivative and stochastic neural network sensitivity analysis for applications in supervised pattern classification problems. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, volume 7, pages 4283–4288 vol.7, 2004. doi:10.1109/ICMLC.2004.1384590.

[26] Jan Macdonald, Stephan Wäldchen, Sascha Hauch, and Gitta Kutyniok. A rate-distortion framework for explaining neural network decisions, 2019. URL: https://arxiv.org/abs/1905.11092, doi:10.48550/ARXIV.1905.11092.

[27] SIMON D'ALFONSO. *Towards a framework for semantic information.* PhD thesis, 2012.

[28] Shayan Hassanpour, Dirk Wubben, and Armin Dekorsy. A novel approach to distributed quantization via multivariate information bottleneck method. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019. doi:10.1109/GLOBECOM38437.2019.9014239.

[29] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. *Proc. 37th Annual Allerton Conference on Communications, Control, and Computing*, page 368–377, 1999.

[30] Ran Gilad-bachrach, Amir Navot, and Naftali Tishby. An information theoretic tradeoff between complexity and accuracy. In *In Proceedings of the COLT*, pages 595–609. Springer, 2003.

[31] Inaki Estella Aguerri and Abdellatif Zaidi. Distributed information bottleneck method for discrete and gaussian sources. *International Zurich Seminar on Information and Communication*, pages 35–39, Feb. 2018. URL: https://doi.org/10.3929/ethz-b-000245048, doi:10.3929/ethz-b-000242151.

[32] T. Berger, Zhen Zhang, and H. Viswanathan. The CEO problem [multiterminal source coding]. *IEEE Transactions on Information Theory*, 42(3), 1996.

[33] H. Viswanathan and T. Berger. The quadratic Gaussian CEO problem. In *Proceedings of IEEE International Symposium on Information Theory*, page 260, 1995.

[34] Y. Ugur, I. E. Aguerri, and A. Zaidi. Vector gaussian CEO problem under logarithmic loss. In *IEEE IT Workshop*, pages 1–5, 2018.

[35] Jin-Jun Xiao and Zhi-Quan Luo. Optimal rate allocation for the vector Gaussian CEO problem. In *1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, pages 56–59, 2005.

[36] Artur Ferreira and Mário Figueiredo. An incremental bit allocation strategy for supervised feature discretization. In *Pattern Recognition and Image Analysis*, pages 526–534. Springer Berlin Heidelberg, 2013.

[37] A. Ababneh. Low-complexity bit allocation for RSS target localization. *IEEE Sensors Journal*, 19(17):7733–7743, 2019.

[38] Yasera Abu-Mostaf. Learning from data - online course. https://home.work.caltech.edu/lectures.html#lectures, 2012.

[39] Miroslav Kubat. *An Introduction to Machine Learning.* Springer, 2015.

[40] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[41] https://onestore.nokia.com/asset/201489.

[42] Mohamed I. AlHajri, Nazar T. Ali, and Raed M. Shubair. Classification of indoor environments for IoT applications: A machine learning approach. *IEEE Antennas and Wireless Propagation Letters*, 2018.

[43] M. I. AlHajri, N. Alsindi, N. T. Ali, and R. M. Shubair. Classification of indoor environments based on spatial correlation of RF channel fingerprints. In *IEEE International Symposium on Antennas and Propagation (APSURSI)*, pages 1447–1448, 2016. doi:10.1109/APS.2016.7696430.

[44] Dario Bega, Marco Gramaglia, Marco Fiore, Albert Banchs, and Xavier Costa-Perez. Deepcog: Cognitive network management in sliced 5G networks with deep learning. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 280–288, 2019. doi:10.1109/INFOCOM.2019.8737488.

[45] Anurag Thantharate, Rahul Paropkari, Vijay Walunj, and Cory Beard. Deepslice: A deep learning approach towards an efficient and reliable network slicing in 5G networks. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pages 0762–0767, 2019. doi:10.1109/UEMCON47517.2019.8993066.

[46] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. Deep learning in mobile and wireless networking: A survey. *IEEE Communications Surveys Tutorials*, 21(3):2224–2287, 2019. doi:10.1109/COMST.2019.2904897.

[47] Quoc-Viet Pham, Nhan Thanh Nguyen, Thien Huynh-The, Long Bao Le, Kyungchun Lee, and Won-Joo Hwang. Intelligent radio signal processing: A survey. *IEEE Access*, 9:83818–83850, 2021. doi:10.1109/ACCESS.2021.3087136.

[48] Rugui Yao, Shengyao Wang, Xiaoya Zuo, Juan Xu, and Nan Qi. Deep learning aided signal detection in OFDM systems with time-varying channels. In *2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pages 1–5, 2019. doi:10.1109/PACRIM47961.2019.8985060.

[49] Nhan Thanh Nguyen and Kyungchun Lee. Deep learning-aided Tabu search detection for large MIMO systems. *IEEE Transactions on Wireless Communications*, 19(6):4262–4275, 2020. doi:10.1109/TWC.2020.2981919.

[50] Neev Samuel, Tzvi Diskin, and Ami Wiesel. Deep MIMO detection. In *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, 2017. doi:10.1109/SPAWC.2017.8227772.

[51] Neev Samuel, Tzvi Diskin, and Ami Wiesel. Learning to detect. *IEEE Transactions on Signal Processing*, 67(10):2554–2564, 2019. doi:10.1109/TSP.2019.2899805.

[52] Mehrdad Khani, Mohammad Alizadeh, Jakob Hoydis, and Phil Fleming. Adaptive neural signal detection for massive MIMO. *IEEE Transactions on Wireless Communications*, 19(8):5635–5648, 2020. doi:10.1109/TWC.2020.2996144.

[53] Hao Huang, Wenchao Xia, Jian Xiong, Jie Yang, Gan Zheng, and Xiaomei Zhu. Unsupervised learning-based fast beamforming design for downlink MIMO. *IEEE Access*, 7:7599–7605, 2019. doi:10.1109/ACCESS.2018.2887308.

[54] Carles Antón-Haro and Xavier Mestre. Learning and data-driven beam selection for mmWave communications: An angle of arrival-based approach. *IEEE Access*, 7:20404–20415, 2019. doi:10.1109/ACCESS.2019.2895594.

[55] Ahmed Alkhateeb, Sam Alex, Paul Varkey, Ying Li, Qi Qu, and Djordje Tujkovic. Deep learning coordinated beamforming for highly-mobile millimeter wave systems. *IEEE Access*, 6:37328–37348, 2018. doi:10.1109/ACCESS.2018.2850226.

[56] Wenchao Xia, Gan Zheng, Yongxu Zhu, Jun Zhang, Jiangzhou Wang, and Athina P. Petropulu. A deep learning framework for optimization of MISO downlink beamforming. *IEEE Transactions on Communications*, 68(3):1866–1880, 2020. doi:10.1109/TCOMM.2019.2960361.

[57] Hao Huang, Yang Peng, Jie Yang, Wenchao Xia, and Guan Gui. Fast beamforming design via deep learning. *IEEE Transactions on Vehicular Technology*, 69(1):1065–1069, 2020. doi:10.1109/TVT.2019.2949122.

[58] Chongwen Huang, Ronghong Mo, and Chau Yuen. Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning. *IEEE Journal on Selected Areas in Communications*, 38(8):1839–1850, 2020. doi:10.1109/JSAC.2020.3000835.

[59] Ahmet M. Elbir and Sinem Coleri. Federated learning for hybrid beamforming in mm-wave massive mimo. *IEEE Communications Letters*, 24(12):2795–2799, 2020. doi:10.1109/LCOMM.2020.3019312.

[60] Muhammad Alrabeiah, Andrew Hredzak, Zhenhao Liu, and Ahmed Alkhateeb. Viwi: A deep learning dataset framework for vision-aided wireless communications. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5, 2020. doi:10.1109/VTC2020-Spring48590.2020.9128579.

[61] Muhammad Alrabeiah, Andrew Hredzak, Zhenhao Liu, and Ahmed Alkhateeb. Viwi data set framework. https://www.viwi-dataset.net/.

[62] Gouranga Charan, Muhammad Alrabeiah, and Ahmed Alkhateeb. Vision-aided 6G wireless communications: Blockage prediction and proactive handoff. *IEEE Transactions on Vehicular Technology*, 70(10):10193–10208, 2021. doi:10.1109/TVT.2021.3104219.

[63] Aldebaro Klautau, Nuria González-Prelcic, and Robert W. Heath. Lidar data for deep learning-based mmwave beam-selection. *IEEE Wireless Communications Letters*, 8(3):909–912, 2019. doi:10.1109/LWC.2019.2899571.

[64] Marcus Dias, Aldebaro Klautau, Nuria González-Prelcic, and Robert W. Heath. Position and lidar-aided mmwave beam selection using deep learning. In *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, 2019. doi:10.1109/SPAWC.2019.8815569.

[65] Li Sun, Jing Hou, and Tao Shu. Optimal handover policy for mmwave cellular networks: A multi-armed bandit approach. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019. doi:10.1109/GLOBECOM38437.2019.9014079.

[66] Delin Guo, Lan Tang, Xinggan Zhang, and Ying-Chang Liang. Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 69(11):13124–13138, 2020. doi:10.1109/TVT.2020.3020400.

[67] Mohamed Sana, Antonio De Domenico, Emilio Calvanese Strinati, and Antonio Clemente. Multi-agent deep reinforcement learning for distributed handover management in dense mmwave networks. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8976–8980, 2020. doi:10.1109/ICASSP40776.2020.9052936.

[68] Mohamed Sana, Antonio De Domenico, Wei Yu, Yves Lostanlen, and Emilio Calvanese Strinati. Multi-agent reinforcement learning for adaptive user association in dynamic mmwave networks. *IEEE Transactions on Wireless Communications*, 19(10):6520–6534, 2020. doi:10.1109/TWC.2020.3003719.

[69] Sara Khosravi, Hossein Shokri-Ghadikolaei, and Marina Petrova. Learning-based handover in mobile millimeter-wave networks. *IEEE Transactions on Cognitive Communications and Networking*, 7(2):663–674, 2021. doi:10.1109/TCCN.2020.3030964.

[70] M. Mollel et al. A survey of machine learning applications to handover management in 5G and beyond. *IEEE Access*, 9, 2021. doi:10.1109/ACCESS.2021.3067503.

[71] Data competition contest. https://ieee-dataport.org/data-competition-contest, 2022.

[72] https://mlc.committees.comsoc.org/datasets/.

[73] Gregor Cerar, Halil Yetgin, Mihael Mohorčič, and Carolina Fortuna. Machine learning for wireless link quality estimation: A survey. *IEEE Communications Surveys Tutorials*, 23(2):696–728, 2021. doi:10.1109/COMST.2021.3053615.

[74] S.K. Kaul, M. Gruteser, and I. Seskar. Creating wireless multi-hop topologies on space-constrained indoor testbeds through noise injection. In *2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006.*, pages 10 pp.–521, 2006. doi:10.1109/TRIDNT.2006.1649191.

[75] Chong Zhou and Randy C. Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 665–674, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3097983.3098052.

[76] Albert Reuther, Peter Michaleas, Michael Jones, Vijay Gadepally, Siddharth Samsi, and Jeremy Kepner. Survey of machine learning accelerators. In *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–12, 2020. doi:10.1109/HPEC43674.2020.9286149.

[77] Zhi Wang, Lihua Li, Yue Xu, Hui Tian, and Shuguang Cui. Handover control in wireless systems via asynchronous multiuser deep reinforcement learning. *IEEE Internet of Things Journal*, 5(6):4296–4307, 2018. doi:10.1109/JIOT.2018.2848295.

[78] Jani Suomalainen, Arto Juhola, Shahriar Shahabuddin, Aarne Mämmelä, and Ijaz Ahmad. Machine learning threatens 5G security. *IEEE Access*, 8:190822–190842, 2020. doi:10.1109/ACCESS.2020.3031966.

[79] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.

[80] A. Wyner and J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1):1–10, January 1976. doi:10.1109/TIT.1976.1055508.

[81] M. Gastpar. On Wyner-Ziv networks. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers*, volume 1, pages 855–859, Nov 2003. doi:10.1109/ACSSC.2003.1292034.

[82] A. B. Wagner, S. Tavildar, and P. Viswanath. Rate region of the quadratic Gaussian two-encoder source-coding problem. *IEEE Transactions on Information Theory*, 54(5):1938–1961, 2008.

[83] T. A. Courtade and T. Weissman. Multiterminal source coding under logarithmic loss. *IEEE Transactions on IT*, 60(1), 2014.

[84] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7):1294–1309, 2009. doi:10.1109/TPAMI.2008.138.

[85] Noam Slonim, Nir Friedman, and Naftali Tishby. Multivariate information bottleneck. *Neural computation*, 18:1739–89, Sep. 2006. doi:10.1162/neco.2006.18.8.1739.

[86] C. Huang, G. C. Alexandropoulos, A. Zappone, C. Yuen, and M. Debbah. Deep learning for UL/DL channel calibration in generic massive MIMO systems. In *IEEE International Conference on Communications (ICC)*, pages 1–6, 2019. doi:10.1109/ICC.2019.8761962.

[87] B. Farber and K. Zeger. Quantization of multiple sources using integer bit allocation. In *Data Compression Conference*, pages 368–377, 2005.

[88] A. Gharouni, P. Rost, A. Maeder, and H. Schotten. Impact of bit allocation strategies on machine learning performance in rate limited systems. *IEEE Wireless Communications Letters*, pages 1–1, 2021. doi:10.1109/LWC.2021.3058893.

[89] Control tutorials, inverted pendulum: State-space methods for controller design. https://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=ControlStateSpace.

[90] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Efficient estimation of mutual information for strongly dependent variables. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, volume 38, pages 277–286, San Diego, California, USA, May 2015. PMLR.

[91] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. ACL '96, page 310–318, USA, 1996. Association for Computational Linguistics. doi:10.3115/981863.981904.

[92] W. S. Levine. *The Control Handbook*. Electrical Engineering Handbook. Taylor & Francis, 1996. URL: https://books.google.de/books?id=2WQP5JGaJOgC.

[93] Mohamed I. AlHajri, Nazar T. Ali, and Raed M. Shubair. 2.4 GHZ indoor channel measurements data set. URL: https://archive.ics.uci.edu/ml/datasets/2.4+GHZ+Indoor+Channel+Measurements.

[94] Afsaneh Gharouni, Peter Rost, Andreas Maeder, and Hans Schotten. Divergence-based bit allocation for indoor environment classificaion. In *IEEE 7th World Forum on Internet of Things*, 2021.

[95] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[96] Ahmed Alkhateeb, Iz Beltagy, and Sam Alex. Machine learning for reliable mmWave systems: Blockage prediction and proactive handoff. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1055–1059, 2018. doi:10.1109/GlobalSIP.2018.8646438.

[97] Roman Klus, Lucie Klus, Dmitrii Solomitckii, Mikko Valkama, and Jukka Talvitie. Deep learning based localization and HO optimization in 5G NR networks. In *International Conference on Localization and GNSS*, pages 1–6, 2020. doi:10.1109/ICL-GNSS49876.2020.9115530.

[98] Ursula Challita, Henrik Ryden, and Hugo Tullberg. When machine learning meets wireless cellular networks: Deployment, challenges, and applications. *IEEE Communications Magazine*, 58(6):12–18, 2020. doi:10.1109/MCOM.001.1900664.

[99] Neil Sinclair, David Harle, Ian A. Glover, James Irvine, and Robert C. Atkinson. Parameter optimization for LTE handover using an advanced SOM algorithm. In *IEEE 77th Vehicular Technology Conference*, 2013. doi:10.1109/VTCSpring.2013.6692692.

[100] Neil Sinclair, David Harle, Ian A. Glover, and Robert C. Atkinson. A kernel methods approach to reducing handover occurrences within LTE. In *18th European Wireless Conference*, pages 1–8, 2012.

[101] Vikash Mishra, Debabrata Das, and Namo Narayan Singh. Novel algorithm to reduce handover failure rate in 5G networks. In *IEEE 3rd 5G World Forum (5GWF)*, pages 524–529, 2020. doi:10.1109/5GWF49715.2020.9221410.

[102] Changsung Lee, Hyoungjun Cho, Sooeun Song, and Jong-Moon Chung. Prediction-based conditional handover for 5G mm-wave networks: A deep-learning approach. *IEEE Vehicular Technology Magazine*, 15(1):54–62, 2020. doi:10.1109/MVT.2019.2959065.

[103] Henrik Martikainen et al. On the basics of conditional handover for 5G mobility. In *29th Annual International Symposium on PIMRC*, 2018. doi:10.1109/PIMRC.2018.8580946.

[104] Afsaneh Gharouni, Umur Karabulut, Anton Enqvist, Peter Rost, Andreas Maeder, and Hans Schotten. Signal overhead reduction for AI-assisted conditional handover preparation. In *Mobile Communication - Technologies and Applications; 25th ITG-Symposium*, pages 1–6, 2021.

[105] 3GPP. NR; Radio Resource Control (RRC); Protocol specification, 2022. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3197.

[106] 3GPP TS38.214. Physical layer procedures for data. 2018. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3216.

[107] Umur Karabulut, Ahmad Awada, Ingo Viering, Andre Noll Barreto, and Gerhard P. Fettweis. Low complexity channel model for mobility investigations in 5G networks. In *IEEE Wireless Comm. and Networking Conference*, 2020. doi:10.1109/WCNC45663.2020.9120625.

[108] E. Yaacoub and Z. Dawy. A survey on uplink resource allocation in OFDMA wireless networks. *IEEE Communications Surveys Tutorials*, 14(2):322–337, 2012. doi:10.1109/SURV.2011.051111.00121.

[109] D. Xu and Q. Li. Resource allocation in delay-QoS constrained multiuser cognitive radio networks. In *6th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6, 2014. doi:10.1109/WCSP.2014.6992100.

[110] R. Ruby, V. C. M. Leung, and D. G. Michelson. Uplink scheduler for SC-FDMA-based heterogeneous traffic networks with QoS assurance and guaranteed resource utilization. *IEEE Transactions on Vehicular Technology*, 64(10):4780–4796, 2015. doi:10.1109/TVT.2014.2367007.

[111] J. Huang, V. G. Subramanian, R. Agrawal, and R. Berry. Joint scheduling and resource allocation in uplink OFDM systems for broadband wireless access networks. *IEEE Journal on Selected Areas in Communications*, 27(2):226–234, 2009. doi:10.1109/JSAC.2009.090213.

[112] K. M. Koumadi and Y. Han. Bandwidth allocation for multimedia applications at mobile stations. *IEEE Communications Letters*, 12(5):359–361, 2008. doi:10.1109/LCOMM.2008.071946.

[113] Lianghai Ji, Andreas Klein, Nandish Kuruvatti, Raja Sattiraju, and Hans D. Schotten. Dynamic context-aware optimization of D2D communications. In *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, pages 1–5, 2014. doi:10.1109/VTCSpring.2014.7023161.

[114] L. Toka, B. Lajtha, É. Hosszu, B. Formanek, D. Géhberger, and J. Tapolcai. A resource-aware and time-critical IoT framework. In *IEEE INFOCOM - IEEE Conerence on Computer Communications*, pages 1–9, 2017. doi:10.1109/INFOCOM.2017.8057143.

[115] M. Chen, W. Saad, and C. Yin. Resource management for wireless virtual reality: Machine learning meets multi-attribute utility. In *GLOBECOM - IEEE Global Communications Conference*, pages 1–7, 2017. doi:10.1109/GLOCOM.2017.8254650.

[116] Afsaneh Gharouni, Peter Rost, Andreas Maeder, and Hans Schotten. Relevance-based wireless resource allocation for a machine learning-based centralized control system. In *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–7, 2021. doi:10.1109/PIMRC50174.2021.9569302.

[117] Classifier comparison with scikit-learn. URL: https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html#sphx-glr-auto-examples-classification-plot-classifier-comparison-py.

Afsaneh Gharouni

Senior Research Specialist

Research Experience

Senior Research Specialist **06/2022 - Present**

Machine Learning in Telecommunications; Research and Standardization

Nokia, Munich, Germany.

Internal PhD Candidate **06/2018 - 05/2022**

Relevance Based Radio Resource Management for Machine Learning Units

Nokia Bell Labs, Munich, Germany.

Research assistant (HiWi) **06/2015 - 10/2016**

Performance Comparison of Equalization Techniques for Indoor THz Communications

Friedrich Alexander University (FAU) of Erlangen-Nuremberg, Erlangen, Germany.

Education

PhD Candidate **06/2018 - 05/2022**

University of Kaiserslautern-Landau, Kaiserslautern, Germany.

Thesis: Relevance Based Radio Resource Management for Machine Learning Units

M.Sc. in Communications and Multimedia Engineering **10/2014 - 07/2017**

Friedrich Alexander University (FAU) of Erlangen-Nuremberg, Erlangen, Germany

Thesis: Sum Rate and Outage Probability Analysis of Non-orthogonal Multiple Access (NOMA) Systems with Residual Hardware Impairments

B.Sc. in Electrical Engineering, Telecommunications **10/2009 - 06/2014**

Ferdowsi Univrsity of Mashhad, Mashhad, Iran

Thesis: Peak-to-Average Power Ratio (PAPR) Reduction of Orthogonal Frequency-Division Multiplexing (OFDM) using Information Theoretic Learning Perspective

Munich, 29th January 2024