IEEE Transactions on
Machine Learning in
Communications and Networking

Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

Digital Object Identifier 10.1109/TMLCN.2022.1234567

Explainable AI for Enhancing Efficiency of DL-based Channel Estimation

Abdul Karim Gizzini[1], Member, IEEE, Yahia Medjahdi[2], Member, IEEE, Ali J. Ghandour[3] and Laurent Clavier[2], Senior Member, IEEE

[1] SogetiLabs Research and Innovation (part of Capgemini), F-92130, Issy Les Moulineaux, France
[2] Centre for Digital Systems, IMT Nord Europe, Institut Mines-Télécom, University of Lille, France
[3] National Center for Remote Sensing, CNRS, Lebanon

Corresponding author: Abdul Karim Gizzini (email: abdul.gizzini@sogeti.com).

This work was supported by the framework of the TRAVEL Project, through the French National Research Agency (ANR) under Grant ANR-24-IAS1-0003.

ABSTRACT The support of artificial intelligence (AI) based decision-making is a key element in future 6G networks. Moreover, AI is widely employed in critical applications such as autonomous driving and medical diagnosis. In such applications, using AI as black-box models is risky and challenging. Hence, it is crucial to understand and trust the decisions taken by these models. Tackling this issue can be achieved by developing explainable AI (XAI) schemes that aim to explain the logic behind the black-box model behavior, and thus, ensure its efficient and safe deployment. Highlighting the relevant inputs the black-box model uses to accomplish the desired prediction is essential towards ensuring its interpretability. Recently, we proposed a novel perturbation-based feature selection framework called XAI-CHEST and oriented toward channel estimation in wireless communications. This manuscript provides the detailed theoretical foundations of the XAI-CHEST framework. In particular, we derive the analytical expressions of the XAI-CHEST loss functions and the noise threshold fine-tuning optimization problem. Hence the designed XAI-CHEST delivers a smart low-complex one-shot input feature selection methodology for high-dimensional model input that can further improve the overall performance while optimizing the architecture of the employed model. Simulation results show that the XAI-CHEST framework outperforms the classical feature selection XAI schemes such as local interpretable model-agnostic explanations (LIME) and shapley additive explanations (SHAP), mainly in terms of interpretability resolution as well as providing better performance-complexity trade-off.

INDEX TERMS 6G, AI, XAI, perturbation-based, feature selection, channel estimation

I. INTRODUCTION

ARTIFICIAL intelligence (AI) is expected to play a crucial role in the overall design of future networks. In particular, 6G will transform the classical Internet of Things (IoT) to “connected intelligence”, by leveraging the power of AI to connect billions of devices and systems worldwide [1]. This concept is defined as native (AI) which is a key element that differentiates 6G networks from the previous wireless networks. In native AI, distributed AI will be embedded within the functionality of all layers [2] to support demands for high data rates and low latency-critical applications.

Generally speaking, the AI-enabled intelligent architecture for 6G networks defines several layers including the intelligent sensing layer [3]. It is worth mentioning that robust environment monitoring and data detection are of great interest in 6G smart applications like autonomous driving [4]. Note that ensuring the reliability of the intelligent sensing layer is highly impacted by the channel estimation accuracy since a precisely estimated channel response influences the follow-up equalization and decoding operations at the receiver, therefore, it affects the sensing accuracy [5]. In this context, channel estimation is one of the major physical

(PHY) layer issues due to the doubly-selective nature of the channel in mobile applications. Conventional channel estimation schemes such as least squares (LS) ignores the presence of noise in the estimation process and requires the transmission of a large number of pilots which decreases the transmission data rate. Whereas, the linear minimum mean square error (LMMSE) channel estimator provides good performance assuming the prior knowledge of the channel and noise statistics in addition to its high computational complexity.

A. DL-BASED CHANNEL ESTIMATION

Recently, deep learning (DL) has been employed in the PHY layer of wireless communications [6], including channel estimation [7]–[10], due to its ability in providing good performance-complexity trade-offs. Among different DL networks, feed-forward neural networks (FNNs) have been widely used as a post-processing unit following conventional channel estimators. In [11], the authors proposed an end-to-end FNN-based scheme for channel estimation and signal detection, where it directly detects the received bits from the received signal. The proposed FNN model consists of 3 hidden layers with 500, 250, and 120 neurons, respectively. We note that in this scheme the FNN model is trained to predict 16 bits only, hence, several concatenated models are needed according to the total number of transmitted bits. Using the same FNN model proposed in [11], the authors in [12] proposed an FNN-based channel estimation scheme that is used to predict the channel response using the received signal, received pilots, and previously estimated channel. Simulation results show that using the previously estimated channel as an FNN input improves the channel estimation accuracy. Another FNN-based channel estimation scheme has been proposed in [13], where LS channel estimation is first applied and combined with the previously estimated channel to be fed as an input to a 3 hidden layer FNN consisting of 512, 256, and 128 neurons, respectively. As reported in [13], employing LS as an FNN input improves the channel estimation accuracy and provides a comparable performance to the LMMSE channel estimation scheme.

To further improve the performance while preserving low computational complexity, the authors in [7]–[9] tried a different strategy that is based on improving the conventional channel estimation accuracy and employing low complex FNN models as post-processing units. In [7], the authors proposed an FNN-based channel estimation scheme that applies data-pilot aided (DPA) channel estimation on top of LS channel estimation. After that a 3 hidden layer FNN consisting of 40, 20, and 40 neurons, respectively is utilized. Simulation results reveal that improving the initial channel estimation allows the use of a low-complex FNN model while recording a significant performance improvement in comparison to the conventional channel estimation schemes. Similarly in [8] and [9] the authors proposed two FNN-based channel estimation schemes that employ spectral temporal averaging (STA) [14] and time-domain reliable test frequency domain interpolation (TRFI) [15] channel estimation before the FNN model which consist of 3 hidden layers with 15 neurons each. STA-FNN and TRFI-FNN outperform the DPA-FNN [7] while recording a substantial computational complexity decrease.

In addition to FNN models, recurrent neural network (RNN) and convolutional neural network (CNN) models have been also used within the channel estimation by also trying to combine several inputs such as the received signal, pilots, and initially estimated channel. RNN-based channel estimation schemes [16]–[18] can provide better channel tracking capability in comparison to the FNN-based channel estimation. Whereas, CNN-based channel estimation [19], [20] is used in the frame-by-frame channel estimation, where the previous, current, and future pilots are employed in the channel estimation for each received signal. Thus, improving the channel estimation accuracy with the cost of a higher computational complexity as well as inducing high processing time in comparison to the RNN and FNN-based channel estimation. We note that in this work we focus on the FNN-based channel estimation since we are targeting low-complex low-latency DL-based solutions.

There exist three main issues concerning the discussed channel estimation schemes which can be defined as follows: (*i*) Identifying the relevant inputs: as previously discussed, the majority of DL-based channel estimation schemes use a combination of information as an input to the utilized DL model without any clear criteria. We note that the classical input selection XAI schemes such as shapley additive explanations (SHAP) and local interpretable model-agnostic explanations (LIME) can not be efficiently used in channel estimation due to the high dimensionality of the DL model input vector in addition to partially consider the correlations between the inputs. Hence, **is there a way to better select the DL black-box model high-dimensional model inputs?** (*ii*) High computational complexity: to guarantee good performance, highly complex DL architectures are employed. However, motivated by the fact that low-complex architectures are required in low-latency applications, so **is there a real need for such high-complex architectures?** (*iii*) Trustworthiness: Despite the good generalization and performance abilities offered by different DL-based channel estimation schemes, they lack trustworthiness since they are considered as "black box" models. Consequently, researchers and industrial leaders are not able to trust the employment of these models in real-case sensitive applications [21]. Therefore, **is there a way to provide interpretability to the decision-making strategy employed DL black box models?**

The mentioned issues can be tackled by developing explainable artificial intelligence (XAI) schemes that provide a reasonable explanation of the decisions taken by black-box models. Thus, ensuring the transparency of the employed models by transforming them from black-box into white-

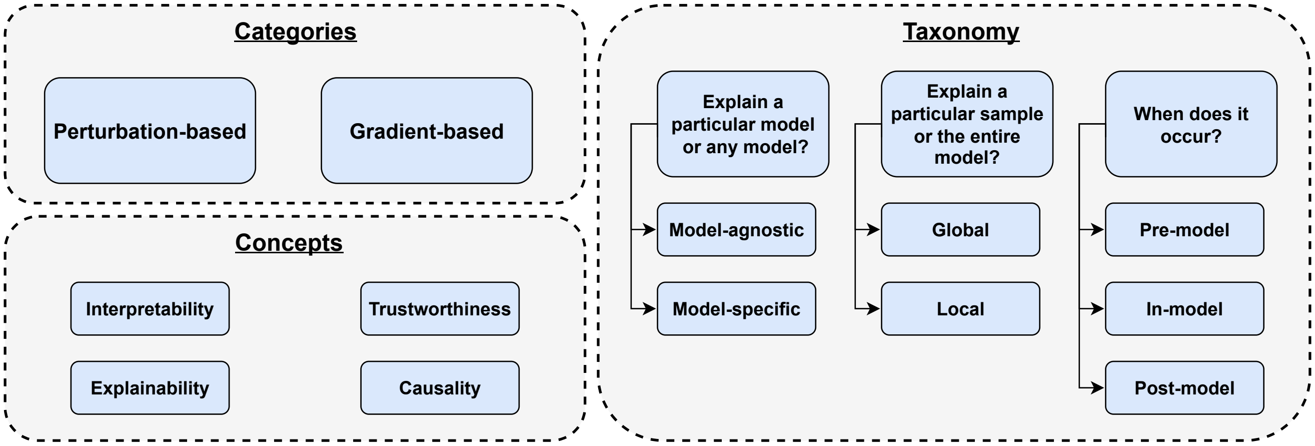


Figure 1. **XAI categories, concepts, and taxonomy.**

box models that can be safely employed in practice. In the following paragraphs, we will discuss the main XAI concepts, taxonomy, and deployment within the wireless communication research domain.

B. XAI MAIN CONCEPTS, CATEGORIES, AND TAXONOMY

XAI defines four main concepts as shown in Figure 1: (*i*) Interpretability: is based on the model design and it refers to how much the black-box model can be understood by humans. For example, decision tree models have good interpretability since a human can easily understand their logic. *ii*) Explainability: is the ability to clarify and justify a particular prediction performed by the black-box model. Hence, it aims to clarify the internal functioning of the employed model. (*iii*) Trustworthiness: is the ability to make professionals feel confident in the decisions taken by the black-box model. (*iv*) Causality: is related to the generalization ability of the black-box model, where models should be able to detect cause-effect relations and adapt to environmental changes.

Generally speaking, XAI methods can be divided into two main categories [22]: (*i*) Perturbation-based or gradient-free methods, where the concept is to perturb input features by masking or altering their values and record the effect of these changes on the model performance. (*ii*) Gradient-based methods where the gradients of the output are calculated with respect to the input via back-propagation and used to estimate importance scores of the input features. Moreover, in terms of the provided explanations, the XAI methods can be further classified into [23]:

- Model-agnostic vs model-specific: Model-agnostic XAI schemes are independent of the internal architecture of the black-box model including the weights and the hidden layers. Whereas, model-specific schemes depend on a specific model like FNN or CNN and can not be generalized to any other model. Therefore, model-agnostic schemes are characterized by their high flexibility and can be used despite the type of the considered model.
- Local vs global: Local XAI schemes are those that generate explanations for a group of samples, thus, they are highly dependent on the utilized dataset. In contrast, global XAI schemes generate explanations that are related more to the model behavior.
- Pre-model, in-model, and post-model strategies: Knowing that the XAI schemes can be applied throughout the entire development pipeline. Hence, interpretability can be acquired in three main phases. Pre-modeling explainability is used to define the useful features of the dataset for a better representation. Hence, pre-modeling aims to perform exploratory data analysis, explainable feature engineering, and dataset description. In contrast, in-model explainability is to develop inherently explainable models instead of generating black box models. Finally, the post-model explainability method extracts explanations that are dependent on the model predictions.

C. XAI FOR WIRELESS COMMUNICATIONS

Wireless communications are still in the early stages of using XAI. The majority of related works in the literature are surveys and reviews about the guidelines and importance of using XAI in wireless communications. In [24] the authors provided a review of the core concepts of XAI including definitions and possible performance vs. explainability tradeoffs. They mainly focused on reviewing the recent DL-based schemes for the PHY and MAC layers and specified the explainability level of the studied schemes which is in general low. In [25] the authors proposed a novel XAI knowledge-powered framework for network automation that effectively adapts to the dynamic changes of complex communication systems as well as provides a human-understandable explanation. The proposed XAI scheme aims to explain the decision-making for automated path selection within the network.

The deployment of XAI in the open radio access network (O-RAN) was recently surveyed in [26], where the authors performed a comprehensive survey on the use of XAI to design a trustworthy and explainable O-RAN architecture. Moreover, an explainable machine learning operations (MLOps) for streamlined automation-native 6G networks has been proposed in [27], [28], where SHAP XAI scheme is employed to assign the features importance [29]. We note that SHAP XAI scheme has been also employed for short-term resource reservation in 5G networks [30] and energy-efficient resource allocation, where the problem becomes more challenging [31]–[33]. It is worth mentioning that, the majority of DL-based resource allocation schemes are based on deep reinforcement learning (DRL) where SHAP assigns importance to the features used by the DRL agent at each state. These features could be the number of active antennas, utilized bandwidth, number of connected users, and the average data rate.

In addition to network optimization and resource allocation, XAI has been employed also in internet-of-things (IoT) networks. In [34] the authors presented a comprehensive survey on XAI solutions for IoT systems including the state-of-the-art past and ongoing research activities. In particular, they focused on the XAI for IoT adaptive solutions using several architectures based on 5G services, cloud services, and big data management. In [35] the authors proposed a novel model-agnostic XAI scheme denoted as transparency relying upon statistical theory (TRUST) for numerical applications. They further tested the proposed TRUST scheme on cybersecurity of the industrial IoT (IIoT). Simulation results show that TRUST scheme outperforms the local interpretable model-agnostic explanations (LIME) scheme [36] in terms of performance, speed, and explainability.

D. MOTIVATION AND CONTRIBUTIONS

To the best of our knowledge, the methodology of deploying XAI schemes in PHY layer applications, specifically, channel estimation is still unclear. Noting that the proposed XAI-based schemes for network optimization [27], resource allocation [30], and secured IoT [34] can not be adapted to the PHY layer applications because in such applications there are no clear discriminative features within the model inputs. In this context, this paper aims to design a novel XAI framework for FNN-based channel estimation, denoted as XAI-CHEST. This framework is based on a perturbation-based model-agnostic global pre-model methodology that jointly performs the channel estimation task and provides the corresponding interpretability. We precisely note that the objective of the XAI-CHEST framework is to provide the interpretability of any black-box model by highlighting the relevant model inputs and analyzing their impact on the overall performance of the considered black-box model. In other words, the provided interpretations can show if the decision making methodology of the black-box model is reliable in case it is focusing on really the most relevant model inputs in achieving the desired estimation task. The XAI-CHEST concept has been partially proposed in [37], where the key idea is to provide the interpretability of black box models by inducing noise on the model input while preserving accuracy. The model inputs are then classified into relevant and irrelevant sets based on the induced noise. It is worth mentioning that the proposed XAI-CHEST framework works according to a low-complex one-shot mechanism, where it could be easily adapted to other DL-based applications including real-time radio resource management. To sum up, the contributions of this work can be summarized as follows:

- The loss function employed to optimize the performance of the proposed interpretability noise model is theoretically detailed, where a custom induced noise control term, $\mathcal{L}_X$, is introduced.
- Deriving the analytical expression and the corresponding simulations of the noise threshold optimization to select the best threshold used in filtering the relevant model inputs.
- Benchmarking the proposed XAI-CHEST framework with LIME and SHAP schemes, where we demonstrate its superiority and efficiency in terms of mechanism, interpretability resolution, performance, and computational complexity.
- Showing that using only relevant inputs instead of the full set improves the performance of the considered DL-based channel estimators.
- Optimizing the architecture of the considered DL-based channel estimator where minimizing the relevant model inputs resulted in a reduction of the model's hidden layers while preserving performance levels.

The remainder of this paper is organized as follows: Section II presents the system model in addition to the DL-based channel estimators to be interpreted. Section III shows the detailed overview of the classical XAI features selection schemes such as LIME and SHAP, in addition to highlighting their limitations and how the proposed XAI-CHEST framework tackle them. Section IV illustrates the designed XAI-CHEST framework as well as the noise threshold fine-tuning optimization problem. In Section V, the performance of the designed XAI-CHEST framework in terms of bit error rate (BER) is analyzed considering several criteria. Finally, Section VI concludes the manuscript.

Notations: Throughout the paper, vectors are defined with lowercase bold symbols $\boldsymbol{s}$, where $\boldsymbol{s}$ and $\bar{\boldsymbol{s}}$ denote the frequency-domain and the time-domain OFDM symbol, respectively. The (i,k) element of $\boldsymbol{s}$ is represented by $\boldsymbol{s}_i[k]$, where i and k stand for the OFDM symbol and the subcarrier indices, respectively. Moreover, we note that the full OFDM symbol $\boldsymbol{s}_i \in \mathbb{C}^{K\times 1}$ includes $\boldsymbol{s}_{i,d} \in \mathbb{C}^{K_d\times 1}$ data symbols and $\boldsymbol{s}_{i,p} \in \mathbb{C}^{K_p\times 1}$ pilots.

II. SYSTEM MODEL

This section illustrates the considered generic system model in addition to the considered DL-based channel estimation scheme to be interpreted.

A. OFDM TRANSCEIVER

In this work, we consider single-input and single-output (SISO) orthogonal frequency division multiplexing (OFDM)-based transmission with non-linear radio frequency (RF) represented by the high power amplifier (HPA) at the OFDM transmitter. As shown in Figure 2, the first operation on the transmitter side is the binary bits generation. Generated bits are scrambled in order to randomize the bits pattern, which may contain long streams of 1s or 0s. The scrambled bits are then passed to the encoder, which introduces some redundancy into the bits stream. This redundancy is used for error correction that allows the receiver to combat the effects of the channel, hence reliable communications can be achieved.

Bits interleaving is used to cope with the channel noise such as burst errors or fading. The interleaver rearranges input bits such that consecutive bits are split among different blocks. This can be done using a permutation process that ensures that adjacent bits are modulated onto non-adjacent subcarriers and thus allows better error correction at the receiver. After that, the interleaved bits are mapped according to the employed modulation technique, i.e., BPSK, QPSK, 16QAM, 64QAM, etc. Bits mapping operation is followed by constructing the OFDM symbols to be transmitted. The data symbols and pilots are mapped to the active subcarriers and passed to the IFFT block to generate the time-domain OFDM symbols and followed by inserting the cyclic prefix (CP). Finally, the CP-OFDM symbol is subjected to the impacts of HPA non-linear distortion as well as the channel and the additive white Gaussian noise (AWGN) noise.

At the receiver side, the CP is removed and the FFT applied to the received symbol. Channel estimation and equalization are performed where the equalized data are demapped to obtain the encoded bits. Afterwards, deinterleaving, decoding, and descrambling are performed to obtain the detected bits. We note that the employed system model is based on the IEEE 802.11p standard [38].

B. SIGNAL MODEL

Consider a frame consisting of I OFDM symbols. The i-th transmitted frequency-domain OFDM symbol $\boldsymbol{s}_i \in \mathbb{C}^{K \times 1}$, can be expressed as:

$$\boldsymbol{s}_i[k] = \begin{cases} \boldsymbol{s}_{i,d}[k], & k \in \mathcal{K}_{\text{d}} \\ \boldsymbol{s}_{i,p}[k], & k \in \mathcal{K}_{\text{p}} \\ 0, & k \in \mathcal{K}_{\text{n}} \end{cases} \tag{1}$$

where $0 \leq k \leq K-1$ denotes the subcarrier index. We note that K_{on} useful subcarriers are used where $K_{\text{on}} = K_p + K_d$. $\boldsymbol{s}_{i,p} \in \mathbb{C}^{K_p \times 1}$ and $\boldsymbol{s}_{i,d} \in \mathbb{C}^{K_d \times 1}$ represent the allocated pilot symbols and the modulated data symbols at a set of subcarriers denoted $\mathcal{K}_{\text{p}}$ and $\mathcal{K}_{\text{d}}$, respectively. $K_n = K - K_{\text{on}}$ denotes the null guard band subcarriers. K_{cp} samples are added to the time-domain OFDM symbol resulting in $\boldsymbol{x}_i \in \mathbb{C}^{K+K_{\text{cp}} \times 1}$ which is then passed to the HPA. According to the Bussgang theorem [39], the HPA output $\boldsymbol{u}'_i \in \mathbb{C}^{K+K_{\text{cp}} \times 1}$ can be expressed as follows:

$$\boldsymbol{u}'_i = \rho \boldsymbol{x}_i + \boldsymbol{z}'_i, \tag{2}$$

where ρ and $\boldsymbol{z}'_i$ refer to the complex gain and the non-linear distortion (NLD), respectively. After that ρ is compensated at the transmitter and $\boldsymbol{u}'_i$ can be rewritten as:

$$\boldsymbol{u}_i = \frac{\boldsymbol{u}'_i}{\rho} = \boldsymbol{x}_i + \boldsymbol{z}_i, \tag{3}$$

where $\boldsymbol{z}_i = \frac{\boldsymbol{z}'_i}{\rho}$ denotes the remaining NLD of the HPA.

After passing via the doubly-dispersive channel and removing the CP, the received time-domain OFDM symbol $\bar{\boldsymbol{y}}_i[n]$ can be expressed as follows:

$$\begin{aligned} \bar{\boldsymbol{y}}_i[n] &= \sum_{l=0}^{L-1} \bar{\boldsymbol{h}}_i[l,n]\boldsymbol{u}_i[n-l] + \bar{\boldsymbol{v}}_i[n] \\ &= \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} \boldsymbol{h}_i[k,n]\tilde{\boldsymbol{u}}_i[k]e^{j2\pi\frac{nk}{K}} + \bar{\boldsymbol{v}}_i[n]. \end{aligned} \tag{4}$$

$\bar{\boldsymbol{h}}_i[l,n]$ denotes the delay-time response of the discrete linear time-variant (LTV) channel of L taps at the i-th OFDM symbol, whereas $\boldsymbol{h}_i[k,n] = \sum_{l=0}^{L-1} \bar{\boldsymbol{h}}_i[l,n]e^{-j2\pi\frac{lk}{K}}$ refers to the frequency-time response. Moreover, $\bar{\boldsymbol{v}}_i$ signifies the AWGN of variance σ^2.

The i-th received frequency-domain OFDM symbol is derived from (4) via discrete Fourier transform (DFT), and thus

$$\boldsymbol{y}_i[k] = \frac{1}{K} \sum_{q=0}^{K-1} \tilde{\boldsymbol{u}}_i[q] \sum_{n=0}^{K-1} \boldsymbol{h}_i[q,n]e^{-j2\pi\frac{n(k-q)}{K}} + \tilde{\boldsymbol{v}}_i[k]. \tag{5}$$

It is noteworthy that index k is used in (4) to express the channel delay-time response in terms of the channel frequency-time response. While the change of index into q in (5) is used to express the i-th received symbol in frequency domain. This, in turn, better illustrates the DFT transform. Moreover, $\boldsymbol{h}_i[q,n]$ refers to time-variant at the scale of the OFDM symbol duration (the index i) and within the symbol itself (the index n).

The time selectivity of the channel depends on the mobility. In very low mobility, where $f_{\text{d}} \approx 0$, $\boldsymbol{h}_i[q,n] = \boldsymbol{h}[q]$ is constant during the whole frame. For moderate to high mobility, the channel variation within the duration of one OFDM symbol is negligible, and therefore, $\boldsymbol{h}_i[q,n] = \boldsymbol{h}_i[q]$. At very high mobility, the channel becomes variant within a single OFDM symbol. In this instance, $\boldsymbol{h}_i[q,n] = \boldsymbol{h}_i[q] + \boldsymbol{\epsilon}_i[q,n]$, where

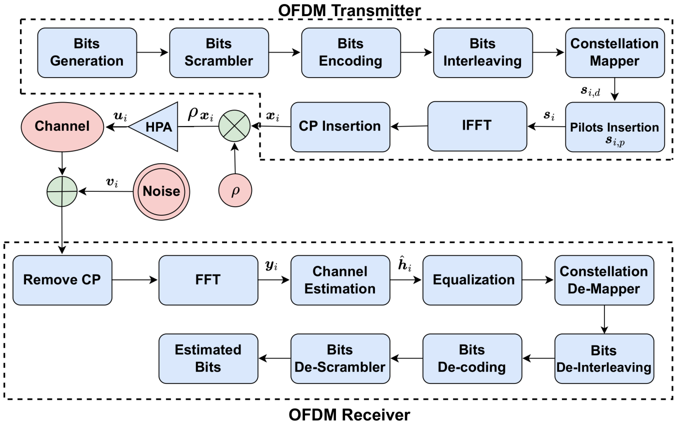


Figure 2. **OFDM transmitter-receiver block diagram.**

$$\boldsymbol{h}_i[q] = \frac{1}{K}\sum_{n=0}^{K-1}\boldsymbol{h}_i[q,n], \text{ and } \boldsymbol{\epsilon}_i[q,n] = \boldsymbol{h}_i[q,n] - \boldsymbol{h}_i[q]. \tag{6}$$

By replacing this in (5), the received frequency-domain OFDM symbol $\boldsymbol{y}_i \in \mathbb{C}^{K_{\text{on}}\times 1}$ is expressed as follows:

$$\boldsymbol{y}_i[k] = \boldsymbol{h}_i[k]\tilde{\boldsymbol{u}}_i[k] + \boldsymbol{e}_i[k] + \tilde{\boldsymbol{v}}_i[k], \tag{7}$$

where

$$\tilde{\boldsymbol{u}}_i[k] = \begin{cases} \boldsymbol{s}_i[k], & \text{linear RF} \\ \boldsymbol{s}_i[k] + \tilde{\boldsymbol{z}}_i[k], & \text{non-linear RF} \end{cases} \tag{8}$$

$\boldsymbol{h}_i \in \mathbb{C}^{K_{\text{on}}\times 1}$, $\tilde{\boldsymbol{v}}_i \in \mathbb{C}^{K_{\text{on}}\times 1}$, and $\boldsymbol{e}_i \in \mathbb{C}^{K_{\text{on}}\times 1}$ refer to the frequency-time response of the doubly-selective channel, the AWGN at the i-th OFDM, and the Doppler-induced intercarrier interference, respectively. $\boldsymbol{e}_i$ can be expressed as:

$$\boldsymbol{e}_i[k] = \frac{1}{K}\sum_{\substack{q=0\\q\neq k}}^{K-1}\sum_{n=0}^{K-1}\boldsymbol{h}_i[q,n]e^{-j2\pi\frac{n(k-q)}{K}}\boldsymbol{s}_i[q]. \tag{9}$$

The Doppler interference destroys the orthogonality of the subcarriers within the received OFDM symbol, leading to a significant degradation in the overall system performance [40], [41]. We note that the objective is to obtain the channel frequency response estimate denoted by $\hat{\boldsymbol{h}}_i \in \mathbb{C}^{K_{\text{on}}\times 1}$.

C. DL-BASED CHANNEL ESTIMATION

Conventional channel estimation depends highly on environment conditions. In frequency-selective slow fading channels, the preamble-based channel estimation is sufficient, since the communication system encounters only muti-path fading and the channel is not changing over time. However, in double selective channels, the impact of Doppler interference is added to the communication system. Thus, the estimated channel at the beginning of the frame, i.e., the preambles, becomes outdated and channel tracking becomes more challenging, especially, in high mobility scenarios. To cope with this challenge, pilot subcarriers are allocated within a transmitted OFDM symbol to allow better channel tracking over time, where several conventional channel estimation and tracking schemes are proposed in the literature. In order to further improve the conventional channel estimation accuracy, DL models are applied as post-processing on top of conventional channel estimators. In this work, we considered the STA-FNN channel estimator [8] as a case study, where we used the optimized XAI-CHEST framework to provide the corresponding reasonable interpretations.

Conventional STA channel estimation scheme [14] is based on the DPA estimation where the demapped data subcarriers of the previously received OFDM symbol are used to estimate the channel for the current OFDM symbol such that:

$$\boldsymbol{d}_i = \mathfrak{D}\big(\frac{\boldsymbol{y}_i}{\hat{\boldsymbol{h}}_{\text{DPA}_{i-1}}}\big),\ \hat{\boldsymbol{h}}_{\text{DPA}_0} = \hat{\boldsymbol{h}}_{\text{LS}}, \tag{10}$$

where $\mathfrak{D}(.)$ refers to the demapping operation to the nearest constellation point according to the employed modulation order. $\hat{\boldsymbol{h}}_{\text{LS}}$ stands for the LS estimated channel at the received preambles. Thereafter, the DPA channel estimates are updated in the following manner:

$$\hat{\boldsymbol{h}}_{\text{DPA}_i} = \frac{\boldsymbol{y}_i}{\boldsymbol{d}_i}. \tag{11}$$

After that, frequency-domain averaging is applied where the DPA estimated channel at each subcarrier is updated as follows:

$$\hat{\boldsymbol{h}}_{\text{FD}_i}[k] = \sum_{\lambda=-\beta}^{\lambda=\beta}\omega_\lambda\hat{\boldsymbol{h}}_{\text{DPA}_i}[k+\lambda],\ \omega_\lambda = \frac{1}{2\beta+1}. \tag{12}$$

Finally, time-domain averaging is employed to reduce the AWGN noise impact such that:

$$\hat{\boldsymbol{h}}_{\text{STA}_i} = (1-\frac{1}{\alpha})\hat{\boldsymbol{h}}_{\text{STA}_{i-1}} + \frac{1}{\alpha}\hat{\boldsymbol{h}}_{\text{FD}_i}. \tag{13}$$

We note that conventional STA channel estimation performs well in the low signal-to-noise ratio (SNR) region. However, it suffers from a considerable error floor in high SNR regions due to the large DPA demapping error resulting from (10) and the fixed frequency and time averaging coefficients $\alpha = \beta = 2$ in (12) and (13), respectively. Therefore, the conventional STA channel estimation scheme is not practical in real-case scenarios due to the high doubly-selective channel variations. As a workaround, a 3 hidden layer FNN model denoted as utility model U consisting of 15 neurons per layer is utilized as a nonlinear post-processing unit following STA. As shown in [8], STA-FNN can better capture the frequency correlations of the channel samples, in addition to correcting the conventional STA estimation error.

We note that through the following sections, the notations are expressed in the context of the DL-based channel estimation as follows:

- Let U be the black-box model denoted as the utility model with parameters θ_U. In general, the U model refers to the channel estimation model that consists of I' inputs, L' hidden layers, and J neurons per layer. Hence, the computational complexity[1] of the U model can be expressed as follows:

$$\begin{aligned}\mathcal{C}_U &= \mathcal{O}(I'L'J)\\ &= \mathcal{O}(I').\end{aligned} \tag{14}$$

- The input and output of the U model are denoted as $\hat{\boldsymbol{h}}'_{\Phi_i} \in \mathbb{R}^{2K_{\text{on}}\times 1}$ and $\hat{\boldsymbol{h}}^{(\text{U})}_{\Phi_i} \in \mathbb{R}^{2K_{\text{on}}\times 1}$, respectively. $\hat{\boldsymbol{h}}'_{\Phi_i}$ corresponds to the conventional estimated channel that is applied prior to the U model. We note that the size of $\hat{\boldsymbol{h}}'_{\Phi_i}$ is $2K_{\text{on}}$ since the conventional estimated channel is converted from complex to real domain before further processing by stacking the real and imaginary values vertically in one vector.
- Φ refers to the employed conventional channel estimation scheme.

The objective is to provide a reasonable interpretation of the behavior of the U model by selecting the most relevant inputs contributing to its prediction.

III. Literature Review

Several methods have been explored in the field of XAI to interpret the machine learning models, such as LIME [36] and SHAP [29]. Both methods aim to assign a relevance score for the input features and interpret how each feature affects the final decision. LIME offers local explanations for individual predictions, allowing users to understand the specific outcomes by applying a pertubation-based iterative mechanism. On the other hand, SHAP employes a permutation-based iterative approach to provide a global perspective by quantifying the feature contributions across the whole dataset. In this section, we explain both XAI methods thoroughly, illustrating their mathematical derivations as well as their limitations. Finally, we show the main criteria that make our proposed XAI-CHEST framework better than the LIME and SHAP XAI methods.

A. Classical XAI methods

LIME aims to replace the original black-box model by a simple interpretable model that approximates the behavior of the original to explain its predictions. Given the instance vector $\hat{\boldsymbol{h}}'_{\Phi_i}$ to be explained, the first step is to generate new instances by perturbing $\hat{\boldsymbol{h}}'_{\Phi_i}$ several times. These perturbed instances are generated from the same distribution of $\hat{\boldsymbol{h}}'_{\Phi_i}$ and then fed to the utility model U to get the corresponding predictions. Each perturbed instance is assigned a weight relevant to its proximity to the original instance $\hat{\boldsymbol{h}}'_{\Phi_i}$. The new weighted dataset is used to train an interpretable model that best describes the behavior of the utility model U. Thus, the learned interpretable model gives coefficients indicating the importance of each feature in $\hat{\boldsymbol{h}}'_{\Phi_i}$.

Given the interpretable models set $\mathcal{G}$, the objective is to find the best interpretable model $g_{\Phi_i} \in \mathcal{G}$ that approximates the behavior of the utility model U. D_{LIME} denotes the number of generated perturbed samples with a proximity function π_{Φ_i} that describes the difference between the sample to be interpreted $\hat{\boldsymbol{h}}'_{\Phi_i}$ and the generated perturbated samples. Hence, the LIME XAI method aims to minimize the following loss function:

$$\mathcal{L}_{\text{LIME}}(\hat{\boldsymbol{h}}'_{\Phi_i}) = \underset{g_{\Phi_i} \in \ \mathcal{G}}{\arg\min}\, \mathcal{L}(U, g_{\Phi_i}, \pi_{\Phi_i}) + \mathcal{C}(g_{\Phi_i}), \tag{15}$$

where $\mathcal{L}(U, g_{\Phi_i}, \pi_{\Phi_i})$ is a loss function that gets smaller as g_{Φ_i} becomes better approximation of U. $\mathcal{C}(g_{\Phi_i})$ denotes the complexity measure of the interpretable model g_{Φ_i}. We note that the complexity is opposed to interpretability. Typically, g_{Φ_i} would belong to the family of linear functions with low complexity. However, it is not always possible to replace the utility U model by a simple interpretable low complex g_{Φ_i} model since non-linear functions could also be required according to the studied problem. In this last case, the complexity $\mathcal{C}(g_{\Phi_i})$ becomes higher, and interpretability decreases. We note that in this work, the default weighted linear regression model is used as a surrogate model g_{Φ_i}. A detailed comprehensive review of the LIME XAI method can be found in [42].

We note that the overall computational complexity of the LIME methods arises from generating and predicting the output of the perturbed samples. This involves calling the utility model U D_{LIME} times. Moreover, training the local interpretable model increases the complexity cost that depends on the chosen interpretable model and the number of features. In this context, the computational complexity of LIME methods can be expressed as follows:

$$\begin{aligned}\mathcal{C}_{\text{LIME}} &= \mathcal{O}(D_{\text{LIME}}(\mathcal{C}_U + 4K_{\text{on}}^2))\\ &= \mathcal{O}(D_{\text{LIME}}K_{\text{on}}^2).\end{aligned} \tag{16}$$

In addition to LIME, SHAP is also used to explain the prediction of the $\hat{\boldsymbol{h}}'_{\Phi_i}$ based on a game-theoretic approach. SHAP estimates the shapley values [43] that fairly distribute the prediction among the input features by considering all possible coalitions of the features acting as players. All possible coalitions of input features are defined, where for each feature, the marginal contribution to the prediction is computed after being added to every possible coalition of other input features. Thus, the shapley value of a feature is calculated by averaging its marginal contribution among all possible coalitions.

Mathematically speaking, let $\mathcal{M}$ be the set containing all the players, i.e, in our case $\mathcal{M} = \hat{\boldsymbol{h}}'_{\Phi_i}$ with $|\mathcal{M}| = 2K_{\text{on}}$ total players. $\mathcal{S} \subseteq \mathcal{M}$ is a subset of participants of full coalition $\mathcal{M}$. $j \in \mathcal{M}$ denotes a specific player with the coalition.

[1] We note that we are using the simplified $\mathcal{O}(.)$. Hence, the complexity of the U model can be simplified to $\mathcal{O}(I')$.

Finally, val is a value function that maps subsets of players $\mathcal{S}$ to a real number, i.e, $val(\mathcal{S})$ represents the revenue of the coalition $\mathcal{S}$. Noting that when a player j joins a set of players $\mathcal{S}$, the marginal contribution of player j to $\mathcal{S}$ denoted as $C_j^{(\mathcal{S})}$can be expressed as follows:

$$C_j^{(\mathcal{S})} = val(\mathcal{S} \cup \{j\}) - val(S)). \tag{17}$$

In other words, the marginal contribution measures the value that player j added when he joined the group of players $\mathcal{S}$. The Shapley value ϕ of player j given the set $\mathcal{M}$ and the value function $val(.)$ can be defined by:

$$\phi_j(\mathcal{M}, val) = \frac{1}{M!} \sum_{\mathcal{S} \subseteq \mathcal{M} \setminus \{j\}} |\mathcal{S}|!(M - |\mathcal{S}| - 1)!\ [C_j^{(\mathcal{S})}]. \tag{18}$$

SHAP XAI method seeks an additive explanation model that is the sum of contributions of individual features, and it is defined as:

$$g(s') = \phi_0 + \sum_{j=1}^{|\mathcal{M}|} \phi_j s'_j, \tag{19}$$

where $g(s') \approx U(\hat{\boldsymbol{h}}'_{\Phi_i})$ is the explanation model, $s' = (s'_1, \dots, s'_{|\mathcal{M}|})^T \in \{0,1\}^M$ is the coalition vector, ϕ_0 is the expected output of the model, and ϕ_j is the Shapely value of the feature j defined in (18).

It is worth mentioning that the manipulation of the exact shapley values is computationally expensive since the iteration over the full possible coalitions is required. Hence, the computational complexity of the exact SHAP method is $\mathcal{O}(2^{K_{\text{on}}})$. Therefore, exact shapley value computation is feasible only for very small numbers of input features and not for high-dimensional ones. In this context, several SHAP approximation methods have been developed to overcome the high complexity issue of the exact SHAP method. Among different approximation methods, we focus on DeepSHAP method, which efficiently approximates shapley values for deep learning models. Instead of iterating over all the possible coalitions, DeepSHAP uses some samples called background samples, where the DeepLIFT mechanism is employed to produce results that adhere to the desirable characteristics of shapley values.

The approximation resolution of DeepSHAP depends mainly on the number of background samples D_{SHAP} used in the approximation. Moreover, these background samples should belong to the distribution of the overall datasets, i.e, chosen from the training dataset. Further details can be found in [44], [45]. We note that employing D_{SHAP} background samples instead of $2^{2K_{\text{on}}}$ possible coalitions reduces the computational complexity of Deep SHAP to:

$$\begin{aligned} \mathcal{C}_{\text{DeepSHAP}} &= \mathcal{O}(4K_{\text{on}} D_{\text{SHAP}} \mathcal{C}_U) \\ &= \mathcal{O}(D_{\text{SHAP}} K_{\text{on}}). \end{aligned} \tag{20}$$

where $\mathcal{C}_{\text{model}}$ denotes the computational complexity of a single forward and backward propagation through the considered black-box model.

Finally, we note that LIME and DeepSHAP methods are employed to provide corresponding relevance scores for the elements of $\hat{\boldsymbol{h}}'_{\Phi_i}$. Based on the provided relevance scores, their performance against the proposed XAI-CHEST framework is evaluated as discussed in Section V-D.

B. Towards XAI-CHEST

Both LIME and DeepSHAP aim to explain the predictions of DL models, thus, enhancing their interpretability. LIME employs local approximation by training a simpler interpretable model that can approximate the functionality of the U model. Therefore, giving insights into how features influence the prediction locally. However, the perturbation methodology employed by LIME is not efficient, since it is based on generating N synthetic perturbated samples which may not be suitable for correlated data in the instance to be explained, i.e, in our case the correlation between the estimated channel among different subcarriers as shown in (12). Moreover, LIME simplifies the original U model, which may not always mimic its original performance. In addition, this simplification is mainly dependent on the choice of interpretable model which can introduce bias. These factors leads to a considerable overall computational complexity of LIME which is $\mathcal{O}(D_{\text{LIME}} K_{\text{on}}^2)$.

On the other hand, DeepSHAP provides both local and global interpretations by Leveraging game theory concepts. Shapley values corresponds to the average marginal contribution of a feature value after considering all possible combinations of input features coalitions. Even though, the manipulation of the exact Shapley values is computational expensive, the approximation methods like the DeepSHAP still suffers from several issues. DeepSHAP approximation requires the backpropagation of contributions via the internal model architecture and is mainly dependent on the background dataset samples D_{SHAP} that are used to compute the approximated Shapely values iteratively, hence, adding another factor of complexity. Moreover, the choice of the background dataset is essential to currently manage the presence/absence of specific input fractures. Therefore, it considers partially the correlation between input features. These issues makes the computational complexity of DeepSHAP still substantial which is equivalent to $\mathcal{O}(D_{\text{SHAP}} K_{\text{on}})$.

It is clearly shown that LIME and DeepSHAP XAI schemes are not practical for high-dimensional correlated data, either because of their overall computational complexity, or due to their iterative mechanism that depends on considering the partial correlation between input features. All these factors limits the performance of LIME and SHAP in practical real-time scenarios. In this context, in this work we propose the perturbation-based XAI-CHEST XAI framework that tackles the limitations of LIME and DeepSHAP XAI schemes, thus, providing a fast one-shot low-complex

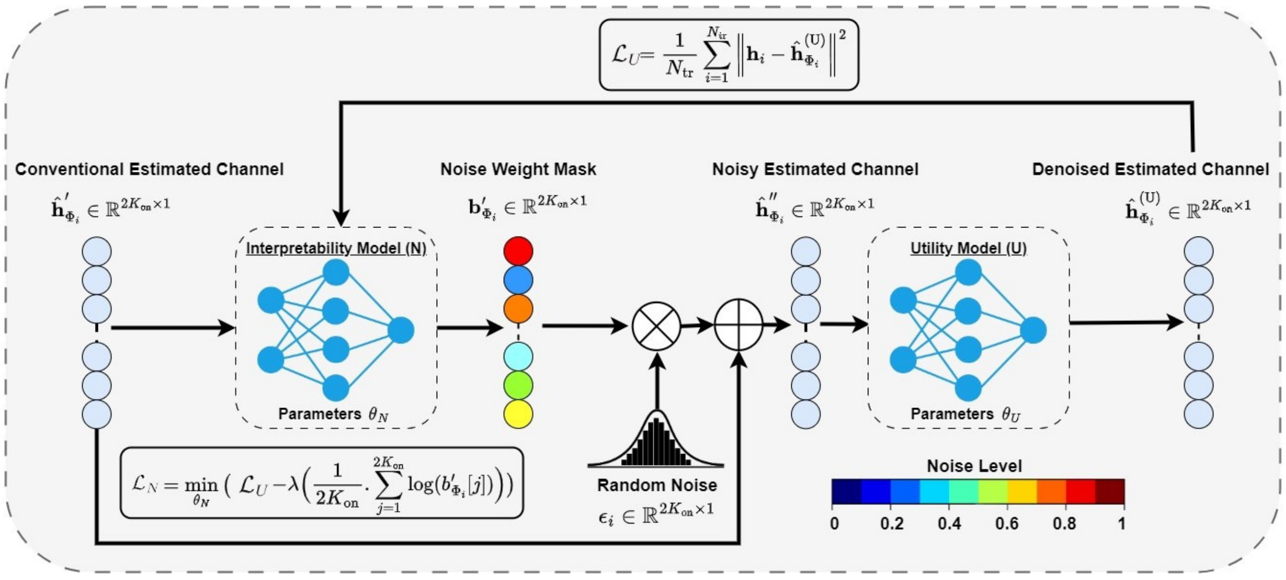

$\mathcal{L}_U = \frac{1}{N_{\text{tr}}} \sum_{i=1}^{N_{\text{tr}}} \left\| \mathbf{h}_i - \hat{\mathbf{h}}^{(\text{U})}_{\Phi_i} \right\|^2$
Conventional Estimated Channel
$\hat{\mathbf{h}}'_{\Phi_i} \in \mathbb{R}^{2K_{\text{on}} \times 1}$
Noise Weight Mask
$\mathbf{b}'_{\Phi_i} \in \mathbb{R}^{2K_{\text{on}} \times 1}$
Noisy Estimated Channel
$\hat{\mathbf{h}}''_{\Phi_i} \in \mathbb{R}^{2K_{\text{on}} \times 1}$
Denoised Estimated Channel
$\hat{\mathbf{h}}^{(\text{U})}_{\Phi_i} \in \mathbb{R}^{2K_{\text{on}} \times 1}$
Interpretability Model (N)
Parameters θ_N
Utility Model (U)
Parameters θ_U
$\mathcal{L}_N = \min_{\theta_N} \left(\mathcal{L}_U - \lambda \left(\frac{1}{2K_{\text{on}}} \cdot \sum_{j=1}^{2K_{\text{on}}} \log(b'_{\Phi_i}[j]) \right) \right)$
Random Noise
$\epsilon_i \in \mathbb{R}^{2K_{\text{on}} \times 1}$
Noise Level
0
0.2
0.4
0.6
0.8
1

Figure 3. Block diagram of the XAI-CHEST framework. The first step is to train the U model and freeze its parameters θ_U. After that, the N model is trained where the objective is to boost the growth of b'_{Φ_i} while preserving the same performance of the pre-trained U model (24). Finally, the subcarriers are filtered based on b'_{Φ_i}, where higher noise weight signifies that the corresponding subcarrier is irrelevant (red colors). In contrast, low noise weights mean that the subcarriers are relevant (blue colors). We recall that in this work we are considering the STA-FNN channel estimation scheme, hence, $\Phi =$ STA.

solution for classifying the U model correlated inputs into relevant and irrelevant. By employing the proposed XAI-CHEST framework, better performance-complexity trade-offs can be designed in addition to interpreting the relevance of the U model inputs. Therefore, produce a reasonable interpretation of the decision-making methodology of the U model by analyzing on which inputs it is focusing to predict the final output.

IV. XAI-CHEST FRAMEWORK DESIGN

Providing external interpretability of the black-box model used for channel estimation could be achieved through classifying the model's input into relevant and irrelevant by employing a perturbation-based methodology. The main intuition is that if a subcarrier is relevant for the decision-making of a trained black box model, then adding noise with high weight to this subcarrier would negatively impact the accuracy of the model. Whereas, if the subcarrier is not contributing to the decision-making of the model, then whatever the induced noise is, the channel estimation accuracy will be preserved. Therefore, it is expected that considering only the relevant subcarriers as model inputs would improve channel estimation performance in comparison to the case where the full subcarriers are given to the model. Moreover, by reducing the model input size, the employed architecture could be further optimized resulting in significantly decreasing the overall computational complexity. Hence, by using the XAI-CHEST framework, we can obtain a reasonable interpretation of the model decision-making methodology, improve the channel estimation performance as well as reduce the required computational complexity.

A. METHODOLOGY

The objective is to provide a reasonable interpretation of the behavior of the U model. Besides the U model, we define the interpretability noise model N, with parameters θ_N, whose purpose is to compute the weight of the noise induced to each subcarrier within the U input vector. The key idea is that the induced noise weights of the N model should not impact the accuracy of the U model. This could be achieved by customizing the loss function of the N model that will adjust the induced noise while simultaneously maximizing the performance of the U model. We note that the U model is trained before the XAI processing of the N model, i.e., the weights of the U model are frozen. Moreover, the U and N models have the same FNN architecture.

Let $\hat{\boldsymbol{h}}'_{\Phi_i}$ be the input of the interpretability N model. The role of the N model is to find a mask $\boldsymbol{b}'_{\Phi_i} \in \mathbb{R}^{2K_{\text{on}} \times 1}$ that can be represented as follows:

$$\boldsymbol{b}'_{\Phi_i} = N(\hat{\boldsymbol{h}}'_{\Phi_i}, \theta_N), \tag{21}$$

where $\boldsymbol{b}'_{\Phi_i} = (b'_{\Phi_i}[1], b'_{\Phi_i}[2], ..., b'_{\Phi_i}[2K_{\text{on}}]) \in [0,1]^{2K_{\text{on}}}$ determines the weight (standard deviation) of the noise applied to each element in $\hat{\boldsymbol{h}}'_{\Phi_i}$. We note that the scaling of $\boldsymbol{b}'_{\Phi_i}$ is achieved using the sigmoid activation function. After that, the generated noise weight mask $\boldsymbol{b}'_{\Phi_i}$ is first multiplied by a random noise $\epsilon \sim \mathcal{N}(0,1)$ sampled from the standard normal distribution, the resultant is added to the conventional estimated channel vector, such that:

$$\hat{\boldsymbol{h}}''_{\Phi_i} = \hat{\boldsymbol{h}}'_{\Phi_i} + \boldsymbol{b}'_{\Phi_i}\epsilon. \tag{22}$$

After that, $\hat{\boldsymbol{h}}''_{\Phi_i}$ is fed as input to the U model, such that:

$$\hat{\boldsymbol{h}}^{(\mathrm{U})}_{\Phi_i} = U(\hat{\boldsymbol{h}}''_{\Phi_i}, \theta_U). \tag{23}$$

The customized loss function of the N model can be expressed as follows:

$$\mathcal{L}_N = \min_{\theta_N} \big(\mathcal{L}_U - \lambda \mathcal{L}_X\big), \tag{24}$$

$\mathcal{L}_U$ denotes the loss unction of the U model when $\hat{\boldsymbol{h}}''_{\Phi_i}$ is fed as an input. Hence, $\mathcal{L}_U$ can be expressed as:

$$\mathcal{L}_U = \frac{1}{N_{tr}}. \sum_{i=1}^{N_{tr}} \left\| \boldsymbol{h}_i - \hat{\boldsymbol{h}}^{(\mathrm{U})}_{\Phi_i} \right\|^2, \tag{25}$$

where $\boldsymbol{h}_i$ refers to the true channel and N_{tr} is the number of training samples. Moreover, the induced noise is controlled by $\mathcal{L}_X$ that can be written as:

$$\mathcal{L}_X = \frac{1}{2K_{\text{on}}}. \sum_{j=1}^{2K_{\text{on}}} \log(b'_{\Phi_i}[j]) \tag{26}$$

We would like to mention that the objective of $\mathcal{L}_N$ is to keep the loss of the U model as low as possible. In other words, minimizing the added noise by the N model while maximizing the generated $\boldsymbol{b}'_{\Phi_i}$. It is worth mentioning that the interpretability measure aims to find a maximum number of low-significant elements. Hence, the term $-\mathcal{L}_X$ gives a negative value, which will get closer to zero when more weights are close to one, meaning that our N model finds more irrelevant elements and can better highlight the significant features. We note that λ is a parameter that allows to give more or less weight to the interpretability measure. The problem is similar to a minimization on $\mathcal{L}_U$ with a constraint on $\mathcal{L}_X$ (λ being a Lagrange multiplier) or simply a regularization term. We note that λ is considered as a hyperparameter that can be tuned using cross-validation in order to achieve the optimal balance between $\mathcal{L}_U$ and $\mathcal{L}_X$ during the training process.

Finally, in the testing phase, $\boldsymbol{b}'_{\Phi_i}$ is scaled back to $\boldsymbol{b}_{\Phi_i} \in \mathbb{R}^{K_{\text{on}} \times 1}$, where the noise weight of the real and imaginary parts for each subcarrier are averaged. The motivation behind this averaging lies in the fact that it is noticed during the training phase that the interpretability model produces almost the same noise weight for the real and imaginary parts of each subcarrier within $\hat{\boldsymbol{h}}'_{\Phi_i}$. The block diagram of the XAI-CHEST framework and the N model training procedure are illustrated in Figure 3 and algorithm 1, respectively. We note that $\hat{\boldsymbol{H}}'_{\Phi} \in \mathbb{R}^{2K_{\text{on}} \times I_{\text{tr}}}$ and $\boldsymbol{H} \in \mathbb{R}^{2K_{\text{on}} \times I_{\text{tr}}}$ denote the training dataset pairs of the conventional estimated channels and the true ones, where I_{tr} is the size of the training dataset. To sum up, the overall functionality of the proposed XAI-CHEST framework can be summarized as follows:

1) **Train the Utility Model (U)**: We train the main utility model U on the original dataset. Hence, feeding it the estimated channels as inputs and the true channels as outputs.

Algorithm 1 N model training

```
Input: Conventional estimated channel: $\hat{\boldsymbol{H}}'_{\Phi}$, true channel: $\boldsymbol{H}$, learning rate: $\eta$, trained $U$ model with parameters $\theta_U$
Output: Trained N model with parameters $\theta_N$
  while not converged do
    for $\hat{\boldsymbol{h}}'_{\Phi_i} \in \hat{\boldsymbol{H}}'_{\Phi_i}$, $\boldsymbol{h}_i \in \boldsymbol{H}_i$ do
      $\boldsymbol{b}'_{\Phi_i} \leftarrow N(\hat{\boldsymbol{h}}'_{\Phi_i}, \theta_N)$
      $\epsilon \leftarrow \mathcal{N}(0, 1)$
      $\hat{\boldsymbol{h}}''_{\Phi_i} \leftarrow \hat{\boldsymbol{h}}'_{\Phi_i} + \boldsymbol{b}'_{\Phi_i}\epsilon$
      $\hat{\boldsymbol{h}}^{(\mathrm{U})}_{\Phi_i} \leftarrow U(\hat{\boldsymbol{h}}''_{\Phi_i}, \theta_U)$
      $\mathcal{L}_U \leftarrow \mathrm{MSE}\big(\boldsymbol{h}_i - \hat{\boldsymbol{h}}^{(\mathrm{U})}_{\Phi_i}\big)$
      $\mathcal{L}_X \leftarrow \mathrm{mean}\big(\log(\boldsymbol{b}'_{\Phi_i})\big)$
      $\mathcal{L}_N \leftarrow \mathcal{L}_U - \lambda\mathcal{L}_X$
      $\theta_N \leftarrow \theta_N + \eta \frac{\partial \mathcal{L}_N}{\partial \theta_N}$
    end for
  end while
```

2) **Train the Interpretability Model (N)**: Next, we train the interpretability model N using the pre-trained U model and the original dataset. In this step, the objective is to maintain the performance of the U model while introducing noise weights to its inputs, as shown in Algorithm 1.
3) **Interpret the U Model**: Here we feed the original testing dataset into the interpretability model N. This step aims to generate the averaged noise weights of the corresponding testing dataset.
4) **Test the U Model with Relevant Inputs**: Finally, we test the U model using only the relevant inputs. These inputs are filtered based on the averaged noise weights from step 3 and a specific threshold. After that, the U model is trained and tested on this modified dataset containing only the selected relevant inputs.

B. NOISE WEIGHT THRESHOLD OPTIMIZATION

After accomplishing the N model training, the fine-tuning of the noise weight threshold denoted by γ is essential for classifying the model inputs into relevant and irrelevant. This could be formulated as an optimization problem, where the objective is to select the best input combination that minimizes the mean squared error (MSE) between the corresponding estimated channel by the U model and the true channel.

Technically speaking, each element in $\boldsymbol{b}'_{\Phi_i}[k]$ denotes the assigned noise weight by the N model to the estimated channel at the corresponding subcarrier in $\hat{\boldsymbol{h}}'_{\Phi_i}[k]$. Hence, $\boldsymbol{b}'_{\Phi_i}[k]$ can be seen as a subcarrier relevance score in contributing to the desired channel estimation task. Therefore, all the subcarriers are grouped according to the assigned $\hat{\boldsymbol{h}}'_{\Phi_i}[k]$, where lower values indicate high relevance and vice versa. In this context, the optimal γ signifies selecting the optimized subcarriers set at a specified threshold, where training the U model with the selected set provides the optimal BER

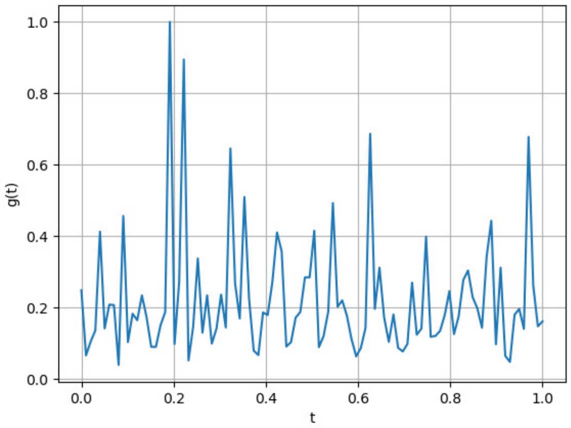


Figure 4. **Normalized restricted loss function $g(t)$. The numerous local minima show the non-convexity nature of the FNN's loss function.**

performance. The relevant and irrelevant subcarriers set are denoted as $\mathcal{R}_{\Phi_i}$ and $\mathcal{IR}_{\Phi_i}$, respectively, and defined as follows:

$$\Psi_\gamma = \begin{cases} \mathcal{R}_{\Phi_i} \leftarrow \mathcal{R}_{\Phi_i} + k, & \boldsymbol{b}'_{\Phi_i}[k] \leq \gamma \\ \mathcal{IR}_{\Phi_i} \leftarrow \mathcal{IR}_{\Phi_i} + k, & \boldsymbol{b}'_{\Phi_i}[k] > \gamma \end{cases} . \tag{27}$$

$\mathcal{R}$ and $\mathcal{IR}$ contain the indices of the relevant and irrelevant subcarriers selected according to the corresponding noise weight as shown in (27). We precisely note that the fine-tuning optimization problem is subjected to improving or preserving the BER in comparison to the case where the full subcarriers are given to the U model.

Let Ω be the generic input given to the U model and Ψ_γ be the optimized model input according to the selected γ, where $\Psi_\gamma \in \{\mathcal{R}_{\Phi_i}, \mathcal{IR}_{\Phi_i}\}$. The considered fine-tuning optimization problem can be mathematically expressed as:

$$\begin{aligned} \min_{\Psi_\gamma, \theta_U} \quad & \mathcal{L}_U = \frac{1}{N_{tr}} . \sum_{i=1}^{N_{tr}} \left(\tilde{\boldsymbol{h}}_i - U(\Omega = \Psi_\gamma, \theta_U) \right)^2 \\ \text{s.t.} \quad & \text{BER}(\Omega = \Psi_\gamma) \leq \text{BER}(\Omega = \hat{\hat{\boldsymbol{h}}}''_{\Phi_i}) \end{aligned} \tag{28}$$

We note that the defined optimization problem in (28) is not convex. The non-convexity can be shown by the line restriction method illustrated in Lemma 1 [46]. We note that the line restriction method is also referred as the 1D slice visualization is a common technique used to illustrate the non-convexity of the loss function of FNN models since the full parameter space is too high-dimensional to visualize directly.

Lemma 1. ***Restriction of a convex function to a line***
A function f: $\mathbb{R}^n \to \mathbb{R}$ is convex, if and only if $\forall x \in \operatorname{dom} f$ and $\forall v \in \mathbb{R}^n$, the function $g = f(x + tv)$ is convex on $\operatorname{dom} g = \{t \in \mathbb{R} \mid x + tv \in \operatorname{dom} f\}$

Table 1. **Parameters of the studied STA-FNN channel estimation scheme.**

Parameter	Values
STA-FNN (Hidden layers; Neurons per layer)	(3;15-15-15)
Activation function	ReLU
Number of epochs	500
Training samples	800000
Testing samples	200000
Batch size	128
Optimizer	ADAM
Loss function	MSE
Learning rate	0.001
Training SNR	40 dB

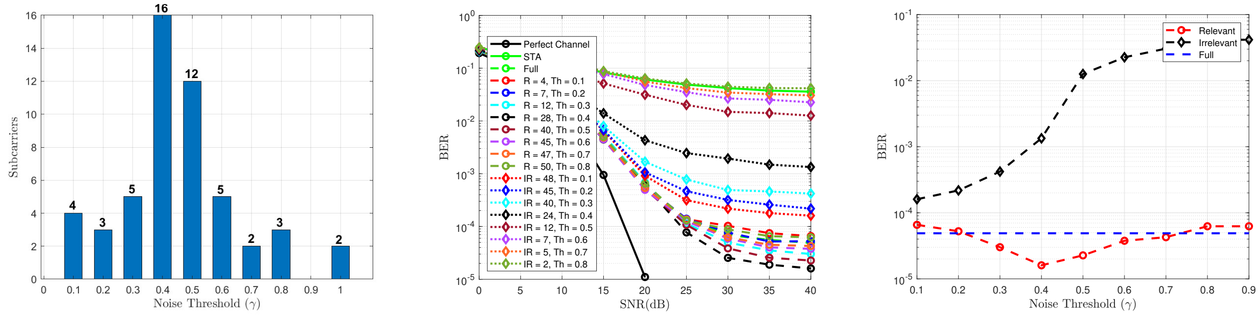
Lemma 1 is based on the line restriction method to prove the convexity of the considered function. In this context, the initial loss function $\mathcal{L}_U$ is reduced to the restricted loss function denoted as $g(t) = \mathcal{L}_U(\theta_U + tv)$, where v and t denote the randomly selected slice and the step size, respectively. Figure 4 shows the $g(t)$ where we can see numerous local minima signifying visually the non-convexity nature of the FNN's loss function as well as the optimization function expressed in (28). We note that in the next section, we provided a heuristic solution of (28), where the BER vs noise weight threshold is analyzed and the best threshold is selected according to the lowest recorded BER among all the considered thresholds.

V. SIMULATION RESULTS

This section illustrates the performance evaluation of the proposed XAI-CHEST framework, where BER performance of STA-FNN channel estimation scheme is analyzed taking into consideration full, relevant, and irrelevant subcarriers. First of all, we start with the noise weight threshold fine-tuning, where the simulation-based solution of (28) is provided. After that, the performance evaluation is performed according to several criteria including the (*i*) modulation order, (*ii*) frequency selectivity of the channel, (*iii*) training SNR, and (*iv*) conventional channel estimation accuracy. Finally, a detailed computational complexity analysis is discussed where we show that further significant reduction in the overall computational complexity can be achieved by employing only the relevant subcarriers identified by the proposed XAI-CHEST framework. We note that the considered channel models [47] are simulated using the comm.RayleighChannel Matlab function that allows the realistic modeling of doubly-dispersive channels. This function takes the required power-delay profile (PDP) in addition to the Doppler sift in order to simulate the effect of relative motion between the transmitter, receiver, and scatterers. We note that the employed Doppler

Table 2. Characteristics of the employed channel models following Jake's Doppler spectrum.

Channel model	Average path gains [dB]	Path delays [ns]
VTV-EX	[0, 0, 0, -6.3, -6.3, -25.1, -25.1, -25.1, -22.7, -2.27, -22.7]	[0, 1, 2, 100, 101, 200, 201, 202, 300, 301, 302]
VTV-SDWW	[0, 0, -11.2, -11.2, -19, -21.9, -25.3, -25.3, -24.4,-28, -26.1, -26.1]	[0, 1, 100, 101, 200, 300, 400, 401, 500, 600, 700, 701]



(a) Noise weight distribution

(b) HFS channel model - QPSK modulation

(c) BER performance vs. noise threshold

Figure 5. Fine-tuning of the noise threshold γ considering the HFS channel model and QPSK modulation. (a) Noise weight distribution for all subcarriers. (b) BER results considering all possible relevant and irrelevant subcarriers combinations. (c) BER vs. noise thresholds considering SNR = 40 dB.

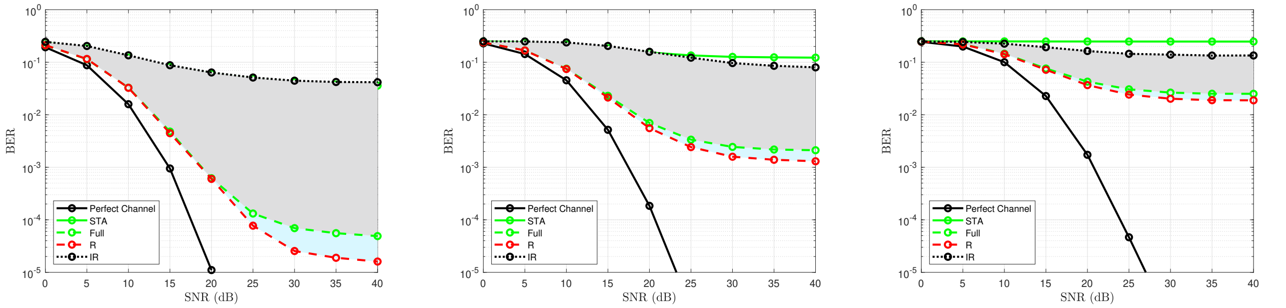
spectrum follows the Jake's model. In this context, the generated perfect channels for each PDP are used in the training of the U and N models. The employed channel models are shown in Table 2: (i) Low-frequency selectivity (LFS), where VTV Expressway (VTV-EX) scenario is employed. (ii) High-frequency selectivity (HFS), where VTV Expressway Same Direction with Wall (VTV-SDWW) scenario is considered. In both scenarios, Doppler frequency $f_d = 1000$ Hz is considered, i.e, the considered channels are doubly-dispersive. However, in this work, we focus on analyzing the impact of frequency selectivity since the FNN models capture the frequency correlation among subcarriers and not the channel tracking over time compared to the RNN models. More precisely, we note that the channel tracking is performed by the conventional channel estimation scheme as shown in (13), where the employed FNN model captures the frequency correlations in addition to denoising the conventional estimated channel. Both the U and N models are trained using a $100,000$ OFDM symbols dataset, splitted into 80% training, and 20% testing. ADAM optimizer is used with a learning rate $lr = 0.001$ with batch size equals 128 for 500 epoch. Simulation parameters are based on the IEEE 802.11p standard [5], where the comb pilot allocation is used so that $K_p = 4$, $K_d = 48$, $K_n = 12$, and $I = 50$. We note that the allocation of the pilot, data, and guard band subcarriers is detailed in [5]. Table 1 summarizes the simulation parameters considered in this work. Finally, we note that the STA channel estimation is considered as an initial estimation prior to the FNN processing. Hence, $\Phi = \text{STA}$, unless stated otherwise.

A. NOISE WEIGHT THRESHOLD ANALYSIS

Selecting the optimal noise weight threshold γ is essential in order to optimize the BER performance of the studied DL-based channel estimator, as shown in (28). To this end we simulated the BER considering all possible values, i.e, $\gamma = [0.1, 0.2, 0.3, ..., 0.8]$. In each case, we trained the U model considering both the relevant and irrelevant subcarriers sets. In this section both the U and N models are trained using the HFS channel model with QPSK modulation and 40 dB training SNR. We note that we train the models on SNR = 40 dB since the models can learn the channel better when the training is performed at a high SNR value because the impact of the channel is higher than the impact of the AWGN noise in this SNR range [48]. Owing to the robust generalization properties of DL, trained networks can still estimate the channel even if the AWGN noise increases, i.e., at low SNR values.

Figure 5(a) shows the distribution of $\boldsymbol{b}'_{\text{STA}_i}$. We notice that the majority of subscribers are distributed more towards zero. This signifies that the model is not sure if the subcarriers can be neglected or not. It is worth mentioning that the pilot subcarriers are assigned the lowest noise weight, i.e., 0.1 which reveals that the U model is not able to neglect the estimated channels at the pilots, and considering them is crucial for high estimation accuracy. This is consistent with the channel estimation rules, where the channel estimates at the pilots are very close to the ideal channel estimation.

As shown in Figure 5(b), we can notice that considering $\gamma = 0.4$ gives the best BER performance among other thresholds. Therefore, the STA-FNN model needs only $|\mathcal{R}_{\text{STA}_i}| = 28$ subcarriers out of the full set, i.e, $|\mathcal{R}_{\text{STA}_i}| = 52$

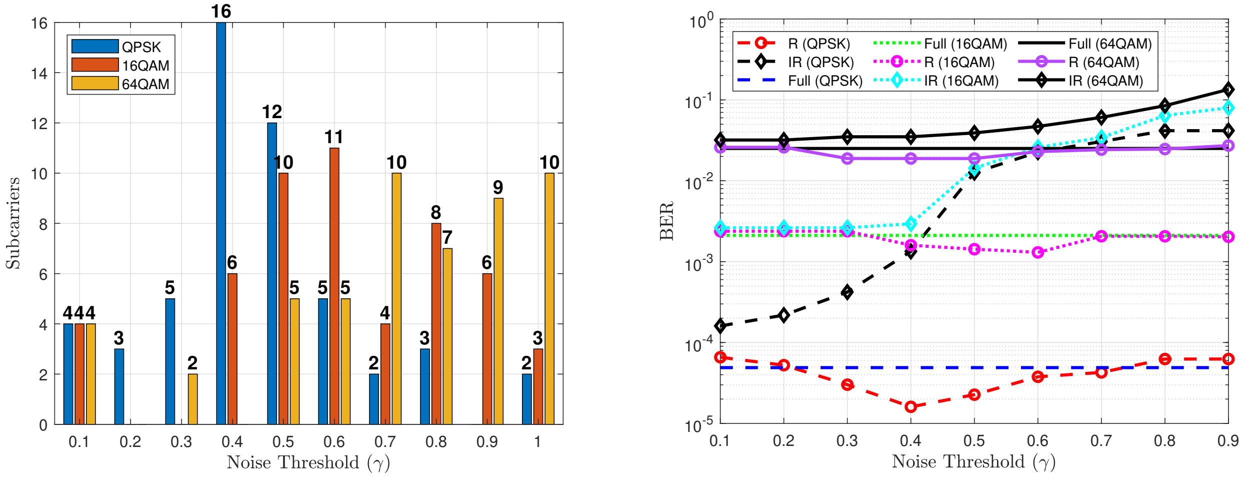


(a) HFS channel model - QPSK modulation

(b) HFS channel model - 16QAM modulation

(c) HFS channel model - 64QAM modulation

Figure 6. BER performance for HFS channel model under different modulation schemes. The simulated modulation schemes are (a) QPSK, (b) 16QAM and (c) 64QAM, respectively. To provide better reading, the best and worst relevant and irrelevant BER performances are only shown, where the regions between the irrelevant-full and full-relevant are highlighted in gray and blue, respectively.



(a) Noise weight distribution

(b) BER performance vs. noise threshold

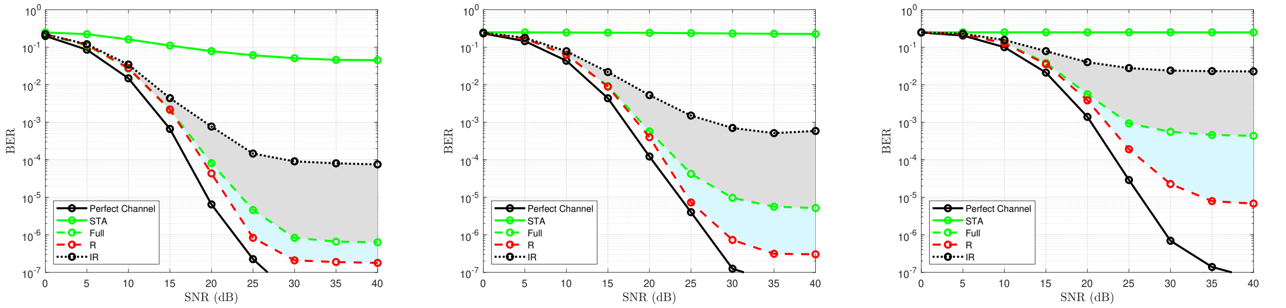
Figure 7. Noise distribution and BER threshold analysis for HFS channel model. (a) Assigned number of subcarriers for a given noise weight across different modulation schemes. (b) BER performance across noise threshold. We plot mainly R, IR, and Full which corresponds to employing only the selected relevant, irrelevant, and Full subcarriers as model inputs, respectively.

in order to provide the best possible performance in the considered HFS channel model. On the contrary, all the irrelevant subcarrier combinations are worse than the full case in terms of BER performance. In other words, considering $|\mathcal{IR}_{\text{STA}_i}| = 48$ which corresponds to excluding only the four pilot subcarriers is not enough to preserve the BER performance of the full case.

Figure 5(c) shows the BER in terms of γ considering SNR = 40 dB. Again, considering more subcarriers in $\mathcal{R}_{\text{STA}_i}$ is beneficial until reaching $\gamma = 0.4$, where the BER performance degrades. This signifies that in complicated scenarios as the case in employing the HFS channel model, the proposed perturbation-based XAI scheme can smartly filter out the relevant model inputs which maximize the its performance.

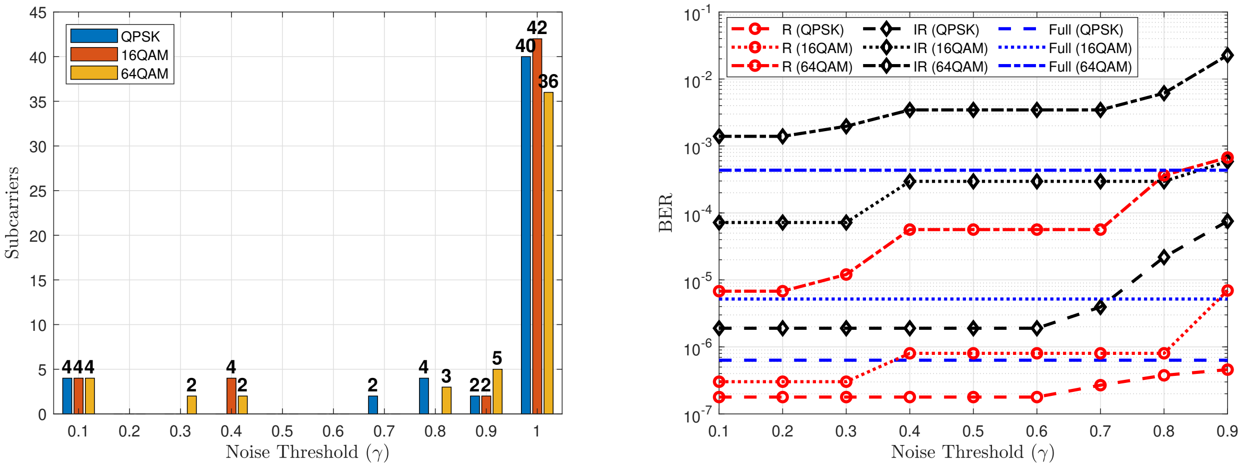
B. IMPACT OF MODULATION ORDER

In this section, we further investigate the impact of the employed modulation order on the noise weight distribution considering also the HFS channel model. Figure 6 shows the BER performance of employing the HFS channel model using QPSK, 16QAM, and 64QAM, respectively. In general, the BER performance degrades as the modulation order increases. This degradation is mainly due to the impact of the dominant multi-path fading in addition to the DPA remapping error. Moreover, in this scenario, employing only the four pilot subcarriers performs almost similarly to the full case. To improve further the BER performance, more relevant subcarriers are needed. Therefore, when the environment becomes more challenging, the channel variation increases among the subcarriers, thus, the noise distribution is shifted towards zero signifying the need for more relevant subcarriers.



(a) LFS channel model - QPSK modulation (b) LFS channel model - 16QAM modulation (c) LFS channel model - 64QAM modulation

Figure 8. BER performance for LFS channel model under different modulation schemes. The simulated modulation schemes are (a) QPSK, (b) 16QAM and (c) 64QAM, respectively. We note that the highlighted gray and blue areas refer to the BER performance when considering the irrelevant and relevant subcarriers, respectively.

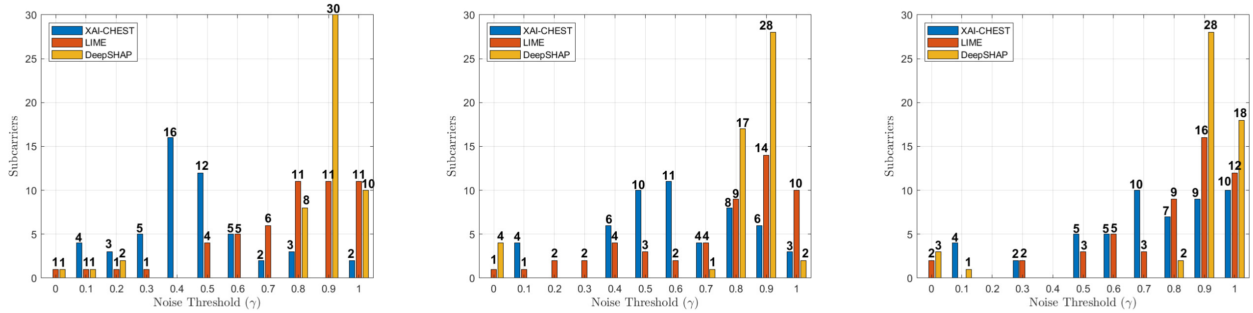


(a) Noise weight distribution (b) BER performance vs. noise threshold

Figure 9. Noise distribution and BER threshold analysis for LFS channel model. (a) Assigned number of subcarriers for a given noise weight across different modulation schemes. (b) BER performance across noise threshold. We plot mainly R, IR, and Full which corresponds to employing only the selected relevant, irrelevant, and Full subcarriers as model inputs, respectively.

However, we can notice that for higher the modulation order, the number of neglected subcarriers increases. This is because the conventional STA channel estimates at these subcarriers are so noisy, so the STA-FNN model neglects them. It is worth mentioning that, the STA-FNN model treats the conventional STA estimated channels separately and does not consider the time correlation between successive OFDM symbols due to the architectural design of the FNN network. Hence, the channel tracking over time is applied within the conventional STA scheme, where the FNN model is used to capture the frequency correlation of the channel samples as well as coping with the conventional STA estimation error. In this context, the STA-FNN model neglects subcarriers due to two main reasons: (i) LFS: The channel variation among the subcarriers is slow, so few subcarriers are required to accurately estimate the channel, as we will discuss in the next Section. (i) HFS: Here the channel variation is significant among the subcarriers, thus, the U model should consider more relevant subcarriers to guarantee good channel estimation accuracy. However, this is subject to the condition where the conventional estimated channel at the considered subcarriers is useful and not so noisy. Therefore, in the HFS channel model, more relevant subcarriers are needed and this is shown in generally shifting the noise weight distribution towards zero, as shown in Figure 7(a). However, for higher modulation orders, mainly 64QAM, the neglected subcarriers are huge due to the bad channel estimation quality at these subcarriers. Hence, avoiding them is useful to guarantee BER performance. We note that the four pilot subcarriers are assigned the lowest noise weight for all the modulation orders. Therefore, the U model is able to classify the pilots as the most relevant subcarriers regardless of the channel's high selectivity and the employed modulation order.

Figure 7(b) shows the BER in terms of γ considering SNR = 40 dB using the HFS channel model. We can notice that ($\gamma = 0.5$, $|\mathcal{R}_{\text{STA}_i}| = 20$), and ($\gamma = 0.5$, $|\mathcal{R}_{\text{STA}_i}| = 11$) are

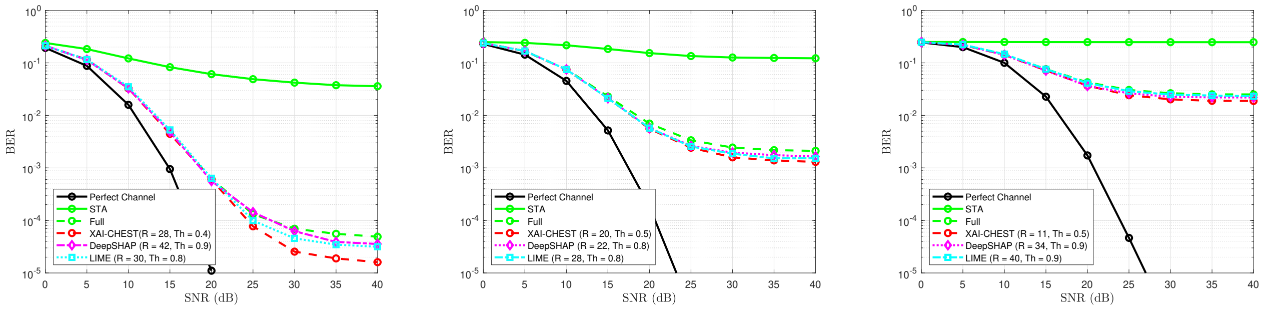


(a) QPSK modulation

(b) 16QAM modulation

(c) 64QAM modulation

Figure 10. **Noise distribution of LIME, DeepSHAP, and the proposed XAI-CHEST framework considering the HFS channel model under different modulation schemes. The simulated modulation schemes are (a) QPSK, (b) 16QAM and (c) 64QAM, respectively.**



(a) HFS channel model - QPSK modulation

(b) HFS channel model - 16QAM modulation

(c) HFS channel model - 64QAM modulation

Figure 11. **BER of LIME, DeepSHAP, and the proposed XAI-CHEST framework considering HFS channel model under different modulation schemes. The simulated modulation schemes are (a) QPSK, (b) 16QAM and (c) 64QAM, respectively.**

the best options corresponding to the 16QAM, and 64QAM modulation orders, respectively. Again we can notice that the functionality of the interpretability model in classifying the subcarriers into relevant and irrelevant is based on the context, i.e., the employed modulation order in this case.

C. IMPACT OF CHANNEL FREQUENCY SELECTIVITY

In this section, we will investigate the performance evaluation using the same methodology of Section B but considering the LFS channel model. Figure 8 shows the BER results of employing QPSK, 16QAM, and 64QAM modulation orders, respectively. We can notice a significant performance degradation as the modulation order increases which is expected. The nice thing lies in employing the pilot subcarriers only, where the corresponding BER performance improves in comparison to the full case. In other words, the BER performance improvement of employing the pilots in comparison to the full case for 64QAM modulation is higher than that for 16QAM and QPSK modulations, respectively. This is because applying the frequency and time domain averaging in the conventional STA channel estimation is no longer reliable due to high demapping error resulting from the DPA channel estimation (10) that is applied prior to the STA estimation. Similarly to the discussion in Section A, employing more relevant and irrelevant subcarriers leads to a BER performance degradation in both cases where in the relevant case, the BER performance is approaching the full case, while in the irrelevant case, the performance is going off the full case.

Figure 9(a) illustrates the noise weight distribution of training the models using different modulation orders. We can notice that distribution is shifted towards one, where the majority of subcarriers are assigned noise weight equal to one. This signifies that these subcarriers are not important for the decision-making methodology of the U model. This is because, in the LFS channel model, the channel presents a smooth variation over the subcarriers, thus, the STA-FNN model needs few subcarriers to accomplish the channel estimation task. Moreover, as the modulation order increases, the noise weight distribution becomes wider, where more subcarriers are assigned more weights. For example, in the 64QM modulation order, it seems that the model needs more subcarriers to preserve good performance, thus the number of subcarriers that are assigned noise weight = 1 decreases. Moreover, in all cases, the model is able to classify pilots as the most relevant subcarriers by assigning them the lowest noise weight regardless of the employed modulation order. The BER vs the noise weight for the considered modulation orders is shown in Figure 9(b) where we can notice that considering only the pilots in the LFS

Table 3. Comparison between LIME, DeepSHAP, and XAI-CHEST.

Method	Type	Features Correlation	Features Dimensionality	Interpretability Resolution	Performance	Complexity	Mechanism
LIME	Perturbation-based	Partially Considered	Low	Erroneous	Limited	$\mathcal{O}(D_{\text{LIME}}K_{\text{on}}^2)$	Iterative
DeepSHAP	Permutation-based	Partially Considered	Low	Erroneous	Limited	$\mathcal{O}(D_{\text{SHAP}}K_{\text{on}})$	Iterative
XAI-CHEST	Pertubation-based	Fully Considered	High	Efficient	Good	$\mathcal{O}(K_{\text{on}})$	One-shot

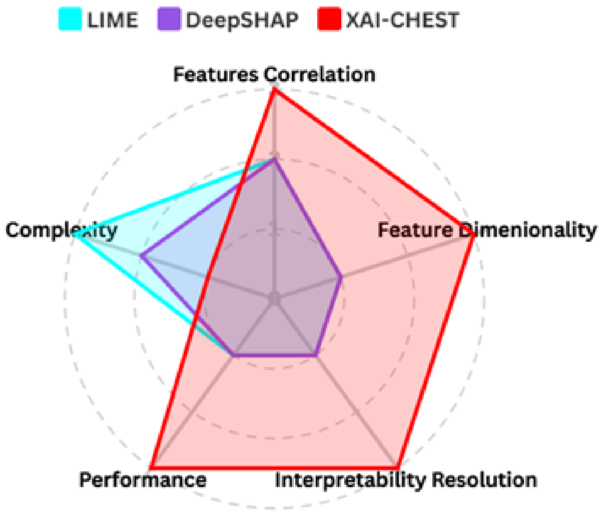


Figure 12. Radar chart comparing the robustness of LIME, DeepSHAP, and the proposed XAI-CHEST framework.

channel model is enough, and there is no need to consider any other subcarriers. On the contrary, all the irrelevant subcarrier combinations are worse than the full case in terms of BER performance. Hence, the absence of the four pilots leads to performance degradation even if the other $|\Psi| = 48$ subcarriers are considered.

D. IMPACT OF INTERPRETABILITY RESOLUTION,

Following the same methodology of Sections A, B, and C, we investigated the interpretability resolution of LIME and DeepSHAP XAI schemes in comparison to the proposed XAI-CHEST framework. We note that the interpretability resolution signifies the ability of the XAI scheme in filtering accurately the relevant subcarriers needed by the U model to optimize the BER performance.

Figure 10 shows the noise weight distribution considering the HFS channel model employing QPSK, 16QAM, and 64QAM modulation orders. Recall that the evaluation of the benchmarked XAI schemes is based on two main criteria: (i) The ability to assign the most relevant scores to the pilot subcarriers. (ii) The ability to select more relevant subcarriers according to the considered scenario.We note that the XAI-CHEST framework induces lower noise on the most relevant subcarriers and vice versa. Hence, for the analysis to be coherent, the normalized complement of LIME and DeepSHAP relevance scores are computed[2], as shown in Figure 10. Therefore, lower LIME and DeepSHAP scores correspond to higher relevance of the subcarriers. For the pilot subcarriers, we can notice that all the benchmarked methods are able to classify the four pilot subcarriers as the most relevant ones. Concerning the second criterion, in contrast to LIME and DeepSHAP, the proposed XAI-CHEST framework outperforms LIME and DeepSHAP while considering the lowest number of relevant subcarriers, as shown in Figure 11(a). This signifies the superiority of the proposed XAI-CHEST framework in selecting the correct relevant subcarriers that contribute to optimizing the BER performance.

We note that Figure 11 shows the BER performance of the benchmarked XAI schemes, where the best results for each are considered, i.e., the threshold γ maximizing the BER performance is selected by using the corresponding $\mathcal{R}_{\Phi_i}$ subcarrier set as input to the U model. Hence, illustrating the maximum potential of each method regardless of the selected relevant subcarriers, since our aim here is to optimize the BER performance as shown in (28). We can notice that even though LIME and DeepShap schemes assume more relevant subcarriers in comparison to the proposed XAI-CHEST framework, they under-perform the BER performance achieved by the proposed XAI-CHEST framework. This signifies that LIME and DeepShap schemes are not currently selecting the relevant subcarriers needed to maximize the BER performance. Instead, they are erroneously mixing real relevant with irrelevant subcarriers, resulting in the recorded under-performance. Moreover, the DeepShap scheme assumes that the U model is not concerned about the majority of the subcarriers by assigning high scores to them. However, this is not correct when employing the HFS channel model since the U model in such scenario considers more relevant subcarriers to maximize the BER performance as discussed in previous sections. It is worth mentioning that erroneously selecting the relevant subcarriers by LIME and DeepSHAP is related to their working methodology. For example, LIME provides the relevance scores by trying to replace the original U model by another interpretable model trained on a generated perturbated dataset. However, this dataset partially considers the correlation between the estimated channel at different subcarriers according to the used proximity measure by the LIME scheme. Therefore, leads to untrusted interpretations. On the other hand, DeepShap par-

[2] In literature, higher relevance scores assigned by LIME and DeepSHAP indicate more feature importance.

tially considers the correlation between the estimated channel at different subcarriers since it chooses modified random samples from the training dataset; hence, the correlation is partially maintained. It is worth mentioning that the proposed XAI-CHEST framework considers the conventional estimated channel directly without any modifications, therefore, it considers the full correlation of the estimated channels between different subcarriers, resulting in trusted interpretations. The overall computational complexity of the employed XAI scheme is another essential factor to consider, especially when dealing with real-time applications. LIME requires $\mathcal{O}(D_{\text{LIME}}K_{\text{on}}^2)$, whereas DeepShap requires $\mathcal{O}(D_{\text{Shap}}K_{\text{on}})$. In contrast, the proposed XAI-CHEST framework requires only one-shot forward pass of the employed interpretability N model. Hence, it requires $\mathcal{O}(K_{\text{on}})$. As a result, the required computational complexity by the proposed XAI-CHEST framework is reduced to $\mathcal{O}(K_{\text{on}})$ making it more efficient in practical scenarios.

Table 3 shows the main characteristics of the benchmarked XAI schemes, as a recap, LIME and DeepSHAP limited performance is mainly due to their poor interpretability resolution which is directly impacted by their working methodology in partially considering the correlation between several input features. In addition, their iterative mechanism increases their computational complexity while limiting their applicability to low-dimenional input space. In contrast, these limitations are efficiently tackled by the proposed XAI-CHEST framework, where the overall performance has been improved by applying a low-complex on-shot mechanism considering the full correlation between input features, which efficiently improves the corresponding interpretability resolution.

Figure 12 illustrates the key performance indicators (KPIs) of the benchmarked XAI schemes in terms of three evaluation scales, low, medium and high. LIME and DeepSHAP share a low support for performance interpretability resolution and feature dimensionality, where a medium support is assigned to feature correlation. Whereas, a medium and high complexity is assigned to DeepSHAP and LIME, respectively. In contrast, the proposed XAI-CHEST framework records the highest score for all KPIs except the complexity, which is the lowest among the benchmarked XAI schemes.

E. IMPACT OF RF NON-LINEAR DISTORTION

In order to further analyze the impact of HPA-induced nonlinearities, we employ QPSK modulation and IBO = 2 dB in the HFS channel models. Recall that this non-linearity impact is equivalent to an additive, uncorrelated noise at the receiver as shown in (2) and (3). Hence, the nonlinear distortion becomes equivalent to a perfectly linear system operating with a degraded SNR, where it uniformly impacts all the subcarriers. As a result, the HPA-induced nonlinearities lead to: (*i*) Pilot contamination: the channel estimates derived from the distorted pilots are inherently noisy and less accurate. (*ii*) Impaired channel estimation: The noisy channel estimates prevent perfect channel equalization in the subsequent data detection stage. (*iii*) Limited BER gain: The combined effect of this SNR degradation impacting the data detection and the impaired channel estimation significantly limits the potential improvement in BER performance.

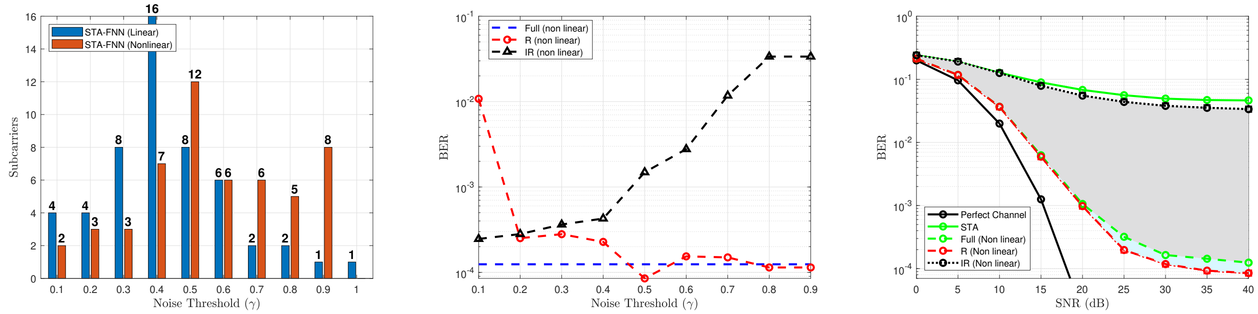
Figure 13 shows the noise weight distribution as well as the BER analysis. It can be noticed that only 2 pilots are assigned the lowest noise weight in comparison to 4 in the linear case. This ensures that the HPA-induced nonlinearities contribute to confusing the subcarrier filtering procedure. However, a slight BER rate performance improvement can be achieved by employing $|\mathcal{R}_{\text{STA}_i}| = 27$ for $\gamma = 0.5$. Therefore, similar insights can be concluded as the linear case where the proposed perturbation based XAI framework is able to filter out the relevant subcarriers while preserving the BER performance when using the full subscribers as an input to the U model.

F. IMPACT OF TRAINING SNR

The sensitivity of the U model training, considering different SNR values, is analyzed in this section. Figure 14(a) shows the noise distribution when considering several training SNRs employing the LFS channel model and QPSK modulation order. Starting by training SNR = $0 - 5$ dB, we can see that the pilot subcarriers are assigned 0.2 noise weight and the distribution is flattened along the entire range. This reveals that even though the pilots have accurate channel estimates, due to the dominant impact of AWGN noise, the U model is not able to assign the lowest noise weight to the pilot subcarriers. It is worth mentioning that when training on SNR = 10 dB, the model starts to identify the pilot subcarriers as the most relevant subcarriers by assigning to two pilots the lowest noise weight, i.e., 0.1. Moreover, as the training SNR increases, the noise distribution is shifted more towards one, signifying that the model is better identifying the relevant and irrelevant subcarriers. Figure 14(b) shows the BER performance when the U model is trained on a specific SNR and tested on the entire SNR range. We can notice that training on higher SNR gives better performance than training on the lower SNR due to the fact the AWGN noise is negligible at high SNRs, thus the U model can learn more efficiently the channel. In addition, the trained model on high SNR can perform well when tested on lower SNRs due to the generalization ability of FNN networks. In conclusion, training on low SNR values leads to a limited performance improvement over the conventional STA channel estimation. Whereas training on high SNR allows the smart feature selection resulting in optimizing the U model input size, as well as significantly improving the BER performance in comparison to the conventional STA channel estimation.

G. IMPACT OF CONVENTIONAL CHANNEL ESTIMATION

To further analyze the impact of the conventional channel estimation, which is implemented prior to the FNN pro-

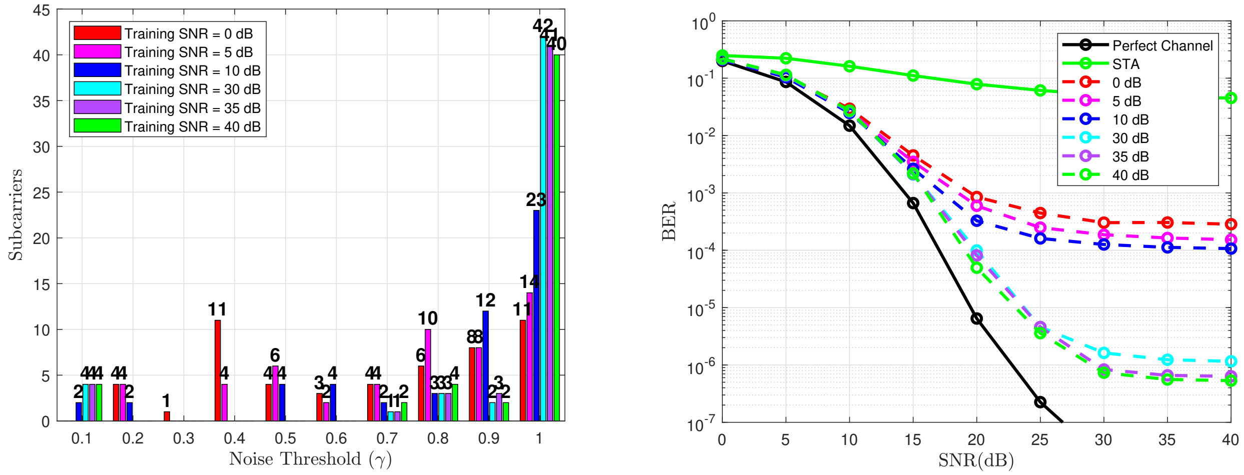


(a) Noise weight distribution

(b) BER performance vs. noise threshold

(c) HFS channel model - QPSK modulation

Figure 13. Noise distribution and BER threshold analysis for HFS channel model while setting Input back-off (IBO) to 2 dB. (a) Assigned number of subcarriers for a given noise weight while using QPSK as a modulation scheme. (b) BER performance across noise threshold. We plot mainly R, IR, and Full which corresponds to employing only the selected relevant, irrelevant, and Full subcarriers as model inputs, respectively. (c) BER performance where the highlighted gray and blue areas corresponds to the irrelevant and relevant subcarriers, respectively.



(a) Noise weight distribution

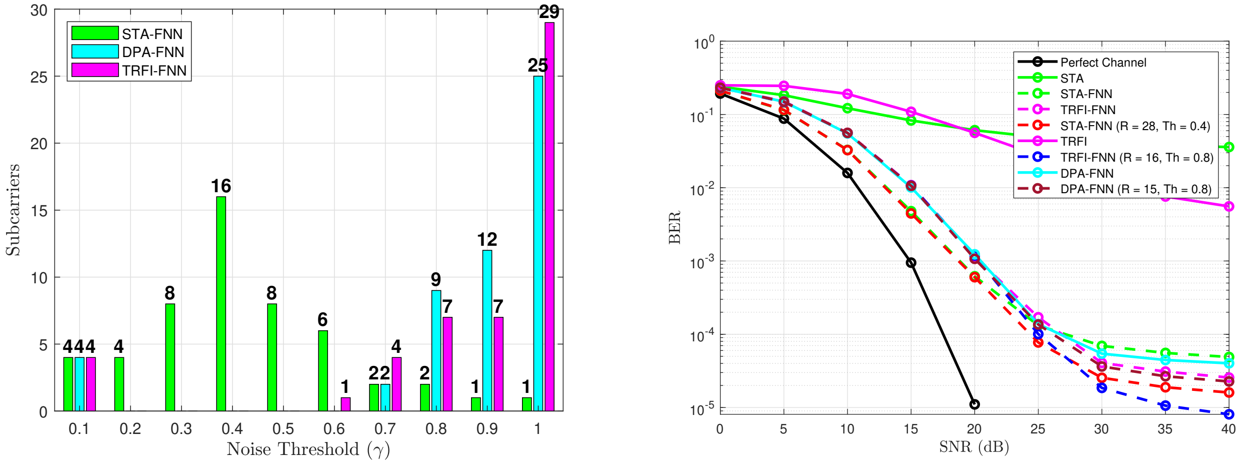
(b) LFS channel model - QPSK modulation

Figure 14. Noise distribution and BER performance considering the LFS channel model and QPSK modulation under different training SNRs. (a) Assigned number of subcarriers for a given noise weight while varying the training SNR from 0 to 40 db. (b) BER performance across different training SNRs.

Table 4. Computational complexity of the optimized FNN architecture employed in the STA-FNN channel estimation scheme.

FNN Architecture	Full (15-15-15)	Relevant (15-15-15)	Relevant (15-15)	Relevant (15)	Relevant (10)	Relevant (5)
FLOPS	7.52 K	4.64 K	4.13 K	3.62 K	2.48 K	1.34 K

cessing on the noise weight distribution, we considered in this section the DPA-FNN [7] and TRFI-FNN [9] channel estimation schemes in addition to STA-FNN. We note that we consider the HFS channel model with QPSK modulation in this analysis since it is more challenging. We recall that the noise weight distribution highlights the behavior of the studied black-box model. If the noise weight distribution is shifted towards 1, then the black-box model employs fewer subcarriers to accomplish the channel estimation task. On the other hand, if the noise weight distribution is shifted towards 0, then more subcarriers are required by the considered black-box model. Moreover, the noise weight distribution is directly impacted by the accuracy of the conventional channel estimation applied prior to the black-box model. Hence, the best relevant subcarriers set is optimized according to the conventional channel estimation. As we can see from Figure 15(b), the conventional TRFI channel estimation outperforms the STA channel estimation in the high SNR region. This is due to the cubic interpolation employed on top of the DPA channel estimation in the TRFI scheme. Similar behavior can be seen with respect to the TRFI-FNN and STA-FNN channel estimators, where the TRFI-FNN also outperforms the STA-FNN in the high SNR region.



(a) Noise weight distribution

(b) HFS channel model - QPSK modulation

Figure 15. Noise distribution and BER performance considering the HFS channel model and QPSK modulation under different channel estimation schemes. (a) Assigned number of subcarriers for a given noise weight considering the DPA-FNN, STA-FNN, and TRFI-FNN channel estimation schemes. (b) BER performance across different channel estimation schemes.

Motivated by the fact that the conventional TRFI is better than the conventional STA channel estimation, it is expected that the TRFI-FNN model may neglect more subcarriers than the STA-FNN channel estimation scheme. This is shown in Figure 14(a), where we can notice that even though we consider the HFS channel model, the noise distribution of the HFS channel model is still shifted towards one, signifying that the TRFI-FNN requires less relevant subcarriers than the STA-FNN channel estimation scheme in order to preserve the BER performance as the full case, where $|\mathcal{R}_{\text{STA}_i}| = 52$. Similar behavior is recorded for the DPA-FNN channel estimation scheme, where the distribution is also shifted towards one in a close manner as the TRFI-FNN scheme. This is because the conventional TRFI scheme slightly outperforms the DPA channel estimation in the considered scenario. However, the STA-FNN and TRFI-FNN channel estimation schemes outperform the DPA-FNN due to the averaging operations and cubic interpolation employed in the conventional STA and TRFI schemes, respectively.

In this context, we can conclude that as the accuracy of the conventional channel estimation increases, the number of selected important subcarriers decreases, where STA-FNN requires $|\mathcal{R}_{\text{STA}_i}| = 28$ relevant subcarriers, which are greater than the relevant subcarriers required by the TRFI-FNN, i.e., $|\mathcal{R}_{\text{STA}_i}| = 16$. This means that the N model is able to induce more noise to the TRFI-FNN input, whereas less noise is induced to the STA-FNN input since it is already noisy.

H. COMPUTATIONAL COMPLEXITY REDUCTION

This section aims to investigate the possibility of optimizing the U model architecture following selecting the most relevant subcarriers so that the BER performance improvement as well as reducing the computational complexity can be achieved. In this context, we consider the LFS channel model with the QPSK modulation order, where the pilot subcarriers are fed to the U model with different architectures. The objective is to reduce the computational complexity of the classical STA-FNN model $(15-15-15)$ while preserving the BER performance of the best relevant case, i.e., employing only the pilots in the LFS channel model.

Figure 16 shows the BER performance of different STA-FNN architectures. We can notice that the FNN architecture could be reduced up to one hidden layer with 15 neurons while preserving the best possible performance. Moreover, decreasing the number of neurons within this architecture to 10 performs the same as the classical STA-FNN channel estimation scheme, i.e., considering the full subcarriers as inputs with the $(15-15-15)$ FNN architecture. However, employing shallow FNN architecture with 5 neurons is not useful at all, where a significant performance degradation is recorded in comparison to the classical STA-FNN architecture.

The computational complexity of the employed FNNs is computed in terms of the number of FLOPS[3] required by each FNN architecture, as shown in Table 4. Employing the same FNN architecture as the classical STA-FNN one but using the pilot subcarriers as input reduces the computational complexity by around $1.5\times$ times in comparison to the classical STA-FNN channel estimation scheme. However, further complexity reduction can be reduced by employing a shallow FNN with 15 neurons, where $2\times$ times can be achieved in comparison to the classical STA-FNN channel estimation scheme. We would like to mention that in terms of $\mathcal{O}(.)$, employing the pilots subcarriers reduce the complexity from $\mathcal{O}(K_{\text{on}})$ to $\mathcal{O}(K_p)$ which is almost equivalent. Because of this we used the number of FLOPS in this section in

[3] We note that the number of FLOPS are calculated using the pytorch-OpCounter package [49].

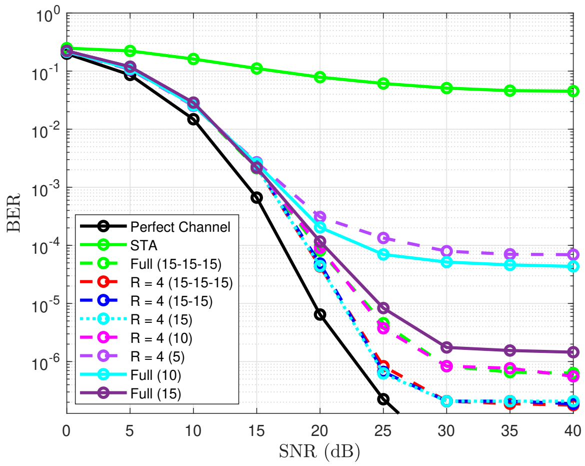


Figure 16. BER performance considering the LFS channel model and QPSK modulation across different FNN architectures. We considered only the pilots as FNN inputs, where the FNN architecture varied from 3 hidden layers, 15 neurons per layer, denoted here as (15-15-15) to 1 hidden layer with 5 neurons, denoted here as (5).

order to further highlight the practical complexity reduction achieved by different configurations. We would like to mention that similar BER performance can be guaranteed as the classical STA-FNN channel estimator by feeding the four pilots to a shallow FNN architecture with 10 neurons, where 3× times less computational complexity is required. Finally, we would like to mention that the proposed XAI-CHEST framework resolves the main issues related to the black box DL models by providing interpretability to the model behavior, performance improvement, and computational complexity reduction by selecting the relevant model inputs and optimizing the architecture of the employed FNN model.

VI. CONCLUSION AND FUTURE PERSPECTIVES

Ensuring the transparency and trustworthiness of AI is crucial for its efficient deployment in critical applications. In this paper, we designed a novel XAI-CHEST framework that provides the interpretability of the FNN models employed in the channel estimation application. The XAI-CHEST framework aims to classify the black-box model inputs into relevant and irrelevant inputs by using a perturbation-based methodology. We developed the theoretical foundations of the XAI-CHEST framework by formalizing the related loss functions. Extensive simulations have been conducted, where the results reveal that a trustworthy, optimized, and low-complexity channel estimation scheme can be designed by selecting only the relevant inputs. In addition, the proposed XAI-CHEST framework outperforms classical feature selection XAI schemes such as LIME and DeepSHAP mainly in terms of interpretability resolution, performance, and computational complexity. As a future perspective, three main research directions could be established:

- XAI-CHEST for RNN-based channel estimation: The functionality of the proposed XAI-CHEST framework is limited since it is not yet adapted to cope with the time variation of the wireless channel. Hence, it is essential to extend the XAI-CHEST framework to deal with RNN-based channel estimation. Thanks to the RNN memory that allows the prediction of the current channel based on previously estimated channels, RNN networks such as long short-term memory (LSTM) and gated recurrent unit (GRU) are able to perform channel estimation and tracking over time. Therefore, adapting the XAI-CHEST to the RNN-based channel estimation provides better noise allocation that varies over time among the received OFDM symbols. Moreover, investigating the impact of the RNN memory size of the efficiency of the noise allocation could provide further performance-complexity trade-offs.
- XAI-CHEST for MIMO-OFDM: The combination of multi-antenna and multi-carrier technologies is a promising technique for ensuring the efficiency of high-speed transmission in wireless communication systems. However, providing performance-complexity trade-offs is crucial for better designing the MIMO-OFDM receiver. In this context, extending the proposed XAI-CHEST for MIMO-OFDM could be beneficial in adapting the size of the considered MIMO system based on the channel correlation and the desired performance.
- Gradient-assisted XAI-CHEST framework: The proposed XAI-CHEST framework is based on an external model-agnostic perturbation-based methodology. Hence, the provided model's interpretability is impacted only by studying the influence of the model inputs on its decision, where the internal architecture of the model remains black-box. Therefore, the current XAI-CHEST framework provides a smart input filtering strategy where model-driven optimization is still unexplored. In this context, investigating internal gradient-based XAI schemes [50] and integrating them within the XAI-CHEST framework will provide a double optimization strategy that leads to filtering the relevant model inputs, as well as fine-tuning the model architecture where relevant layers and neurons are preserved.

References

[1] C.-X. Wang, X. You, X. Gao, X. Zhu, Z. Li, C. Zhang, H. Wang, Y. Huang, Y. Chen, H. Haas, J. S. Thompson, E. G. Larsson, M. D. Renzo, W. Tong, P. Zhu, X. Shen, H. V. Poor, and L. Hanzo, "On the Road to 6G: Visions, Requirements, Key Technologies, and Testbeds," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 905–974, 2023.

[2] G. Liu, Y. Huang, N. Li, J. Dong, J. Jin, Q. Wang, and N. Li, "Vision, requirements and network architecture of 6G mobile network beyond 2030," *China Communications*, vol. 17, no. 9, pp. 92–104, 2020.

[3] H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, and K. Wu, "Artificial-Intelligence-Enabled Intelligent 6G Networks," *IEEE Network*, vol. 34, no. 6, pp. 272–280, 2020.

[4] M. I. Ashraf, Chen-Feng Liu, M. Bennis, and W. Saad, "Towards Low-Latency and Ultra-Reliable Vehicle-to-Vehicle Communication,"

in *2017 European Conference on Networks and Communications (EuCNC)*, 2017, pp. 1–5.

[5] A. K. Gizzini and M. Chafii, "A Survey on Deep Learning Based Channel Estimation in Doubly Dispersive Environments," *IEEE Access*, vol. 10, pp. 70 595–70 619, 2022.

[6] H. Huang, S. Guo, G. Gui, Z. Yang, J. Zhang, H. Sari, and F. Adachi, "Deep Learning for Physical-Layer 5G Wireless Techniques: Opportunities, Challenges and Solutions," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 214–222, 2020.

[7] S. Han, Y. Oh, and C. Song, "A Deep Learning Based Channel Estimation Scheme for IEEE 802.11p Systems," in *IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[8] A. K. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, "Deep Learning Based Channel Estimation Schemes for IEEE 802.11p Standard," *IEEE Access*, vol. 8, pp. 113 751–113 765, 2020.

[9] A. K. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, "Joint TRFI and Deep Learning for Vehicular Channel Estimation," in *IEEE GLOBECOM 2020*, Taipei, Taiwan, Dec. 2020.

[10] A. Karim Gizzini, M. Chafii, A. Nimr, R. M. Shubair, and G. Fettweis, "CNN Aided Weighted Interpolation for Channel Estimation in Vehicular Communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 12, pp. 12 796–12 811, 2021.

[11] H. Ye, G. Y. Li, and B. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.

[12] X. Ma, H. Ye, and Y. Li, "Learning Assisted Estimation for Time-Varying Channels," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, 2018, pp. 1–5.

[13] Y. Yang, F. Gao, X. Ma, and S. Zhang, "Deep Learning-Based Channel Estimation for Doubly Selective Fading Channels," *IEEE Access*, vol. 7, pp. 36 579–36 589, 2019.

[14] J. A. Fernandez, K. Borries, L. Cheng, B. V. K. Vijaya Kumar, D. D. Stancil, and F. Bai, "Performance of the 802.11p Physical Layer in Vehicle-to-Vehicle Environments," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 3–14, 2012.

[15] Yoon-Kyeong Kim, Jang-Mi Oh, Yoo-Ho Shin, and Cheol Mun, "Time and Frequency Domain Channel Estimation Scheme for IEEE 802.11p," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 1085–1090.

[16] A. K. Gizzini and M. Chafii, "RNN Based Channel Estimation in Doubly Selective Environments," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 1–18, 2024.

[17] J. Pan, H. Shan, R. Li, Y. Wu, W. Wu, and T. Q. S. Quek, "Channel Estimation Based on Deep Learning in Vehicle-to-Everything Environments," *IEEE Communications Letters*, vol. 25, no. 6, pp. 1891–1895, 2021.

[18] A. K. Gizzini and M. Chafii, "Deep Learning Based Channel Estimation in High Mobility Communications Using Bi-RNN Networks," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 2607–2612.

[19] L. Li, H. Chen, H.-H. Chang, and L. Liu, "Deep Residual Learning Meets OFDM Channel Estimation," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 615–618, 2020.

[20] D. Luan and J. S. Thompson, "Channelformer: Attention Based Neural Solution for Wireless Channel Estimation and Effective Online Training," *IEEE Transactions on Wireless Communications*, vol. 22, no. 10, pp. 6562–6577, 2023.

[21] D. Kaur, S. Uslu, A. Durresi, S. Badve, and M. Dundar, "Trustworthy explainability acceptance: A new metric to measure the trustworthiness of interpretable ai medical diagnostic systems," in *Complex, Intelligent and Software Intensive Systems*, L. Barolli, K. Yim, and T. Enokido, Eds. Cham: Springer International Publishing, 2021, pp. 35–46.

[22] I. E. Nielsen, D. Dera, G. Rasool, R. P. Ramachandran, and N. C. Bouaynaya, "Robust Explainability: A Tutorial on Gradient-based Attribution Methods for Deep Neural Networks," *IEEE Signal Processing Magazine*, vol. 39, no. 4, pp. 73–84, 2022.

[23] V. Arya, R. K. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, A. Mojsilović *et al.*, "One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques," *arXiv preprint arXiv:1909.03012*, 2019.

[24] W. Guo, "Explainable Artificial Intelligence for 6G: Improving Trust between Human and Machine," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 39–45, 2020.

[25] Y. Wu, G. Lin, and J. Ge, "Knowledge-Powered Explainable Artificial Intelligence for Network Automation toward 6G," *IEEE Network*, vol. 36, no. 3, pp. 16–23, 2022.

[26] B. Brik, H. Chergui, L. Zanzi, F. Devoti, A. Ksentini, M. S. Siddiqui, X. Costa-Pérez, and C. Verikoukis, "A Survey on Explainable AI for 6G O-RAN: Architecture, Use Cases, Challenges and Research Directions," 2023.

[27] F. Rezazadeh, H. Chergui, L. Alonso, and C. Verikoukis, "SliceOps: Explainable MLOps for Streamlined Automation-Native 6G Networks," 2023.

[28] F. Rezazadeh, H. Chergui, and J. Mangues-Bafalluy, "Explanation-Guided Deep Reinforcement Learning for Trustworthy 6G RAN Slicing," in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2023, pp. 1026–1031.

[29] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[30] P. Barnard, I. Macaluso, N. Marchetti, and L. A. DaSilva, "Resource Reservation in Sliced Networks: An Explainable Artificial Intelligence (XAI) Approach," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 1530–1535.

[31] N. Khan, A. Abdallah, A. Celik, A. M. Eltawil, and S. Coleri, "Explainable AI-aided Feature Selection and Model Reduction for DRL-based V2X Resource Allocation," *IEEE Transactions on Communications*, pp. 1–1, 2025.

[32] A.-D. Marcu, S. K. Gowtam Peesapati, J. Moysen Cortes, S. Imtiaz, and J. Gross, "Explainable Artificial Intelligence for Energy-Efficient Radio Resource Management," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, 2023, pp. 1–6.

[33] N. Khan, S. Coleri, A. Abdallah, A. Celik, and A. M. Eltawil, "Explainable and Robust Artificial Intelligence for Trustworthy Resource Management in 6G Networks," *IEEE Communications Magazine*, pp. 1–7, 2023.

[34] S. K. Jagatheesaperumal, Q.-V. Pham, R. Ruby, Z. Yang, C. Xu, and Z. Zhang, "Explainable AI Over the Internet of Things (IoT): Overview, State-of-the-Art and Future Directions," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 2106–2136, 2022.

[35] M. Zolanvari, Z. Yang, K. Khan, R. Jain, and N. Meskin, "TRUST XAI: Model-Agnostic Explanations for AI With a Case Study on IIoT Security," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 2967–2978, 2023.

[36] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You? Explaining the Predictions of Any Classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[37] A. K. Gizzini, Y. Medjahdi, A. J. Ghandour, and L. Clavier, "Towards Explainable AI for Channel Estimation in Wireless Communications," *IEEE Transactions on Vehicular Technology*, pp. 1–6, 2023.

[38] A. Abdelgader and L. Wu, "The Physical Layer of the IEEE 802.11 p WAVE Communication Standard: The Specifications and Challenges," in *The Physical Layer of the IEEE 802.11 p WAVE Communication Standard: The Specifications and Challenges*, vol. 2, 10 2014.

[39] J. J. Bussgang, "Crosscorrelation functions of amplitude-distorted Gaussian signals," 1952.

[40] N. D. Ricklin, "Time Varying Channels : Characterization, Estimation, and Detection," Ph.D. dissertation, University of California, San Diego, 2010.

[41] A. K. Gizzini and M. Chafii, "Low complex methods for robust channel estimation in doubly dispersive environments," *IEEE Access*, vol. 10, pp. 34 321–34 339, 2022.

[42] P. Knab, S. Marton, U. Schlegel, and C. Bartelt, "Which lime should i trust? concepts, challenges, and solutions," *arXiv preprint arXiv:2503.24365*, 2025.

[43] E. Štrumbelj and I. Kononenko, "A general method for visualizing and explaining black-box regression models," in *Adaptive and Natural Computing Algorithms: 10th International Conference, ICANNGA 2011, Ljubljana, Slovenia, April 14-16, 2011, Proceedings, Part II 10*. Springer, 2011, pp. 21–30.

[44] E. Mosca, F. Szigeti, S. Tragianni, D. Gallagher, and G. Groh, "Shap-based explanation methods: a review for nlp interpretability," in *Proceedings of the 29th international conference on computational linguistics*, 2022, pp. 4593–4603.

[45] A. M. Salih, Z. Raisi-Estabragh, I. B. Galazzo, P. Radeva, S. E. Petersen, K. Lekadir, and G. Menegaz, "A perspective on explainable

artificial intelligence methods: Shap and lime," *Advanced Intelligent Systems*, vol. 7, no. 1, p. 2400304, 2025.

[46] S. P. Boyd and L. Vandenberghef, *Convex optimization*. Cambridge university press, 2004.

[47] I. Sen and D. W. Matolak, "Vehicle–Vehicle Channel Models for the 5-GHz Band," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 2, pp. 235–245, 2008.

[48] A. K. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, "Enhancing Least Square Channel Estimation Using Deep Learning," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[49] V. Sovrasov. (2018-2023) ptflops: a Flops Counting Tool for Neural Networks in Pytorch Framework. [Online]. Available: https://github.com/sovrasov/flops-counter.pytorch

[50] A. K. Gizzini, Y. Medjahdi, and M. B. Mabrouk, "GRACE: Gradient-based XAI Scheme for Channel Estimation in Wireless Communications," in *2024 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, 2024, pp. 572–577.