IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 January 2023, accepted 30 January 2023, date of publication 8 February 2023, date of current version 18 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3243438

RESEARCH ARTICLE

Swift HARQ Based on Machine Learning for Latency Minimization in URLLC

SALEH ALMARSHED[1], (Member, IEEE),
DIONYSIA TRIANTAFYLLOPOULOU[2], (Member, IEEE),
AND KLAUS MOESSNER[2,3], (Senior Member, IEEE)
[1]King Abdulaziz City for Science and Technology (KACST), Riyadh 11442, Saudi Arabia
[2]5G and 6G Innovation Centre (5GIC and 6GIC), Institute for Communication Systems (ICS), University of Surrey, GU2 7XH Guildford, U.K.
[3]Faculty of Electrical Engineering and Information Technology, Technical University Chemnitz, 09126 Chemnitz, Germany

Corresponding author: Saleh Almarshed (s.almarshed@surrey.ac.uk)

This work was supported in part by KACST and in part by the University of Surrey.

ABSTRACT Ultra-reliable low-latency communication (URLLC) has been introduced in the 5th Generation (5G) radios for mission-critical applications that demand strict reliability and latency traffic to guarantee the rapid delivery of short packets (up to 1 ms) with a success probability rate of 99.999%. The challenging reliability and latency requirements of URLLC have significant impact on the air-interface design, especially on the Hybrid Automatic Repeat reQuest (HARQ) mechanism. This study focuses on satisfying link latency requirements by reducing the delay that arises in the presence of the HARQ operation. To this end, we propose a Swift HARQ protocol empowered by machine learning techniques to estimate the decodability of a packet early enough within its maximum number of allowable retransmission attempts. This can allow the transmitter to react faster by dropping the non-decodable packets, or activating the repetition mode where parts of the HARQ feedback can be omitted. As shown through system-level simulations, the proposed model achieves a delay reduction of more than 50% compared to the traditional HARQ, and increases the system throughput by up to 40% when multiple HARQ retransmissions are required.

INDEX TERMS 5G, 6G, artificial intelligence, HARQ, latency, machine learning, URLLC.

I. INTRODUCTION

specified, with the aim to support the slot extension more broadly in the frequency domain, while the slot duration is up a new horizon for new interactive and mission-critical communication links such as Factory Automation (FA), autonomous driving, remote surgery, intelligent grid automation, and Unmanned Aerial Vehicles (UAVs). To this end, 5G wireless networks are envisaged to support a new class of traffic named URLLC that aims at delivering short packet within up to 1 ms at a success rate of at least 99.999% [1]. The 3GPP standards committee has recognized the need for a new Orthogonal Frequency-Division Multiplexing (OFDM) based frame structure to satisfy such challenging requirements. Accordingly, the 5G numerology concept was introduced in Rel. 15, allowing the frame structure to be flexible in order support a variety of use cases based on the Quality of Service (QoS) demands of the users. In particular, in order to meet the stringent URLLC latency demands, a mini-slot frame structure and a wider subcarrier space concept are that need to be resolved to fully achieve the target latency requirements of URLLC applications. For instance, the HARQ processes over the air interface are not yet standardized. More specifically, the required time to successfully deliver a packet, including all its possible retransmission rounds in case of transmission failure, has to be minimized to meet the 1 ms latency target [2]. Otherwise, the contribution of traditional HARQ to URLLC would be doubtful.

The HARQ process can be seen as a combination of forward error correction (FEC) and error detection. Both are performed at the receiver, in order to release a

The associate editor coordinating the review of this manuscript and approving it for publication was Jad Nasreddine.

negativeacknowledgement message (NACK) when a decoding failure due to poor channel conditions is experienced, or acknowledgement message (ACK) in the case of a successful reception. Despite the network reliability goal that can be URING the recent decades, the tremendous developshortened in the time domain (i.e., the slot duration can be ment in wireless communication applications opened as short as to 0.25 ms). However, there are still open issues achieved with HARQ, it can cause increase the latency cost due to delay arising from the round-trip time (RTT) to process the multiple retransmissions and multiple signal decoding operations. Therefore, the delay associated with HARQ can create a bottleneck, compromising the achievement of the challenging URLLC latency requirement. The above challenges motivated our work, in which we designed a Swift HARQ mechanism based on machine learning (ML), capable of estimating the decodability of a packet ahead of reaching its maximum number of allowable retransmission attempts. This can allow the transmitter to react by dropping the non-decodable packets at an earlier basis or activating the repetition mode, where parts of the HARQ feedback can be eliminated.

A. RELATED WORKS

Questions surrounding URLLC have recently attracted the attention of academia and industry stakeholders. Reference [3] surveyed various techniques and methodologies pertaining to the requirements of URLLC. Reference [4] discussed information-theoretic aspects of short-packet communications. Reference [5] highlighted the impact of waveform numerology, FEC and signalling channel placement on achievable communication latency. A comparison study was carried out in [6] to find suitable coding techniques that can meet the URLLC requirements. Reference [7] aimed to maximize the URLLC link throughput by jointly optimize the block length and pilot length. Reference [8] provide an overview of 3GPP Releases 15, 16, and 17 from a Physical Layer Design perspective. The optimization of the HARQ mechanism has been addressed in the literature from diverse aspects. The performance of HARQ protocols in the presence of short packets has been studied in [9] and [10]. In [11] and [12] different adaptive power and rate allocation mechanisms are proposed to enhance the retransmission attempts to improve the overall system performance. Reference [13] proposed to couple the HARQ feedback with information about the decoder status. This information can allow the transmitter to dynamically adapt to the channel conditions during the retransmissions throughout setting new parameters to meet the target block error rate. Reference [14] studied a fixed-point equation capable of locating the optimal number of retransmissions as a function of the maximum allowable energy consumption for Chase combiningHARQ (CC). Reference [15] proposed an approach that can estimate which retransmission round has a high successful decoding probability. Based on this, the receiver can stop performing some decoding activities during the retransmission until reaching the estimated round.

In line with this study, the concept of Early HARQ (EHARQ) feedback has previously been discussed, in [16], [17], [18], and [19]. Reference [16] proposed an E-HARQ approach, utilizing the loglikelihood ratio (LLR) to estimate feedback following partial signal decoding. Reference [17] aimed to further expedite early feedback release by making estimation decisions based on only a part of the codeword. This approach utilizes Low-Density Parity-Check (LDPC) that offers excellent opportunities to exploit substructures, and therefore using the ready decoded part of the codeword during the decoding procedure. Reference [18] employed ML techniques to enhance feedback prediction throughout training an ML model to classify the decodability status, rather than utilizing the traditional method (i.e., hard threshold-based). Reference [19] further proposed an E-HARQ estimator, based on ML, capable of speeding up the feedback releasing by decoupling the feedback estimation from the signal decoding. The proposed mechanism shows better estimation performance, especially after introducing the feature inspired by the channel state information.

The works in [15], [18], [19], and [20] are most closely related to ours, as the early HARQ feedback estimation in [15] utilizes only one feature that can be concluded from the sum channel gain. As an alternative, our mechanism is relied on ML to estimate the decodability outcome based on two high-dimensional features, independently extracted for each packet. On the other side, the early HARQ feedback mechanism in [18], [19], and [20] still relies on the handshaking mechanism to release feedback following each retransmission round, which is considered a source of delay, especially when the link requires multiple HARQ retransmissions. Our proposal adopts a repetition strategy, in which some handshaking signalling can be eliminated, hence effectively minimizing the link delay. Finally, our proposed early HARQ strategy differs fundamentally from what is suggested in the literature in terms of the mechanism's ability to exploit the estimated feedback to adapt the remaining scheduled reattempts, i.e., multiple HARQ feedback, with the channel conditions, and that is through allocating varying power levels for each round. The system performance is also evaluated using Clustered Delay Line (CDL) fading conditions under different levels of Channel State Information (CSI) levels at the receiver.

B. CONTRIBUTIONS

This paper aims to reduce the link latency arising from the presence of the HARQ mechanism. Mainly, we develop a Swift HARQ to early estimate the probability of correctly decoding the received packet within the maximum permitted retransmission attempts; based on this protocol, the paper achieves the following key contributions:

- **Latency reduction:** The proposed mechanism utilizes the repetition mode when there is a high likelihood to decode the signal. Hence the receiver can reduce the transmission delay by omitting the unnecessary NACK feedback during the retransmission until successfully decoding the signal.

- **Throughput improvements:** Excluding the nondecodable signal at an early stage can efficiently contribute to improving the channel utilization. Simulation results revealed approximately 40% improvement in the system throughput at low SNR regions. The proposed scheme also achieve the exact reliability requirements at SNR values below 1 dB, compared to traditional HARQ.
- **Link reliability:** The early HARQ feedback estimates the required number of retransmissions to successfully decode the signal. This information can allow the adaption of the remaining scheduled attempts at the appropriate power level, thereby increasing the probability of decoding the packet earlier than the classical approach. To precisely evaluate the performance of Swift HARQ in a realistic environment, we train and validate the model with a standardized waveform compliant with 3GPP 5G NR Release 15 [21].
- **System evaluations:** We present an extended theoretical discussion, including analyzing the transmission scenario of a packet in a probabilistic model that incorporates the impact of imperfect early feedback estimation. As short-sized packets characterize URLLC, we modified some of the waveform parameters to align with potential URLLC waveform aspects by utilizing the mini-slot and wider subcarrier spacing concept. Accordingly, the link level performance is evaluated, from the waveform designer's point of view, under different system constraints such as input features, classification algorithms, and specific channel conditions.

The paper is structured as follows: Section II describes the Swift HARQ concept. Section III presents the system model. Section IV presents the performance evaluation, including the classification evaluation as well as the data link performance, and finally, Section V provides concluding remarks and outlines for future work.

II. SWIFT HARQ CONCEPT

The proposed Swift HARQ mechanism is depicted in Fig.1. Let us consider a transmitter-receiver pair that communicates under fading channel conditions in the downlink direction; i.e., from the Base Station (BS) to the User Equipment (UE). When the first payload arrives from the core network, the transmitter will process each packet by encoding the payload into a parent codeword containing N codewords, each with a length dedicated to T_n channel use. These codewords share identical information, but are encoded with different parity bits. Hence, the URLLC packet is defined as the transmission of the first codeword, alongside all possible other codewords in case the involved parties required extra HARQ attempts to $(N-1)$. Based on this definition, the packet has a length of $T_{total} = \sum_{n=1}^{N} T_n$ channel use, where T_n is the channel use of codeword n, n in $(1 \ldots N)$.

TABLE 1. The definition of parameters.

Symbol	Definition
Nt	Number of transmitting antennas
Nr	Number of receiving antennas
N	Maximum allowable transmission attempts
$M+1$	Number of possible outcomes
$S_{M,k}$	Possible estimation states
V	Number of training samples
L	The codeword length
BER_L	Estimated bit error rate
CSI	Channel state information
(α_i,b)	Weight and threshold of ML model
Ω	Total delay on the forward direction
f	Total delay of HARQ feedback on the backward direction
δ_tr	Total number of channel used for standard HARQ
δ_{Swift}	Total number of channel used for Swift HARQ

On the receiver end, the UE in the traditional HARQ will receive the codeword and perform baseband processing. If the receiver succeeds in decoding the first codeword, it responds with an ACK; otherwise, it stores the received signal, responds with a NACK, and seeks another codeword. These transmission activities are known as handshaking processes, and continue until the packet is successfully decoded or the maximum permitted number of retransmission rounds is achieved. Unlike traditional HARQ, when the codeword decoding fails following the first attempt, the proposed Swift HARQ will couple the NACK message with an estimation concerning the number of retransmission rounds required to successfully decode the packet. Based on this useful feedback, the transmitter reacts early by dropping the nondecodable packet, or enters repetition mode in which the codewords will continuously retransmit. The receiver, meanwhile, will stop releasing the NACK message until the codeword is successfully decoded. At this point, it is essential to discuss the motivation behind estimating the decodability of the received codeword within the maximum number of allowable retransmissions. Firstly, the non-decodability decision permits the transmitter to drop the non-decodable packet at an early stage; hence exploiting the available resources for another transmission, leading to both enhanced throughput and reduced latency on the link. Secondly, the decodability decision, which also includes an estimation of the required number of retransmissions, can enhance the system's reliability by allowing the transmitter to schedule the remaining retransmission rounds with suitable power level parameters that can be adjusted according to the channel's conditions.

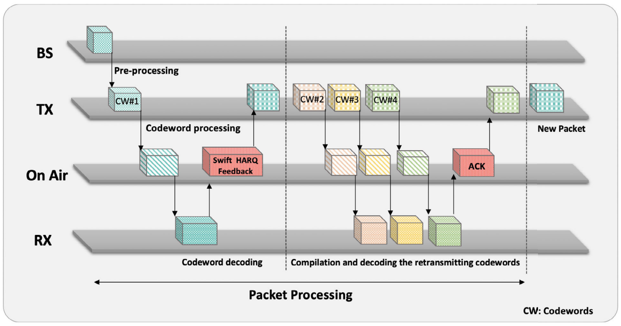


FIGURE 1. Packet travel scenario in the downlink, assuming the packet is decodable within three re-transmitting rounds.

A. PROBABILISTIC MODEL FOR SWIFT HARQ

Estimating the decodability of a packet at an early stage is a challenging task, since the estimation mechanism is heavily dependent on the characteristics of instantaneous channel conditions in the first round, whereas channel conditions can experience random fluctuations during the reception of the other retransmission attempts. This can result in estimation mismatching and two error types. The first one occurs when the packet is estimated to be decodable, but it then appears that it was not, i.e., after the Cyclic Redundancy Checking (CRC) task, resulting in a false-positive decision. The second type refers to the case when a packet is predicted to be non-decodable but can be successfully decoded, resulting in a false-negative decision. Nevertheless, obtaining a reliable estimator can effectively mitigate the occurrence of these types of errors. To evaluate the impact of imperfect estimation, the transmission scenario for a packet is analysed in a simple probabilistic model, as shown in Fig.2. The goal is to determine which type of error can negatively affect the system performance, and then calculate the probability of this effective error.

Let us assume a packet C can have up to N allowable transmission attempts, as indexed by $i = 1, 2, \dots, N$, where i corresponds to the number of transmission attempts. The packet can be decodable when $1 \leq i \leq N$ at outcome $M = i$, otherwise the decodability outcome is unsuccessful at $M = 0$. However, predicting one of these possible outcomes at an early stage can fall into one of the $N+1$ possible states. Let us denote each state as $S_{M,k}$, where k is an integer ranging from $[0, \dots, N]$. The first possible outcome is $M = 0$, representing a non-decodable outcome for a packet after consuming all the retransmission attempts. The probability of this is $P_0 \equiv P(M = 0)$. In this case, the maximum benefit when employing Swift HARQ can be gained when the estimated state $S_{M,k}$ matches the actual outcome at $M = k = 0$. This estimation refers to a true negative decision, and also has a probability:

$$P_{TN} \equiv P(S_{0,0} = 0|M = 0). \tag{1}$$

Otherwise, estimation of the other states $(S_{0,1}, \dots, S_{0,N})$ will lead to a false positive decision with probability:

$$P_{FP} \equiv P((S_{0,1}) \vee \ldots.. \vee (S_{0,N})|M = 0). \tag{2}$$

The occurrence of this false feedback will result in unnecessary retransmissions and miss the opportunity to obtain the maximum benefit of delay saving. However, since dropping the non-decodable packet at an early stage can be considered a credit for Swift HARQ, failing to estimate the correct decision will not add any overhead to the system compared to traditional HARQ; hence, the negative effect of this error on the system is negligible. Additional possible outcomes arise when $i > 0$. These outcomes represent decodable packets with a different number of transmission attempts in the range of $[1, \dots, N]$ with probability $P_i \equiv p(M = i)$. Estimating one of these outcomes can lead to two possible decisions:

Firstly, a true positive decision occurs when the feedback estimation lies along one of these decodability states $S_{1,k}, \ldots\ldots, S_{M,N}$, at k: $1, \dots, N$, and the true positive probability can be expressed as:

$$P_{TP} \equiv P((S_{M,k} = k)\vee, \ldots.., \vee(S_{M,N} = N)|M = i) \tag{3}$$

Notably, we consider all decodability states as correct decisions, although the estimated state can be mismatched with the actual outcome at $M = k$, because this inaccurate estimation will not negatively impact link latency. This means BS will repeatedly retransmit until acknowledgment is attained, or the maximum allowable transmission is reached, regardless of the estimated required number of retransmission rounds. An accurately matched estimation will be considered as a credit to enhance the link reliability by adjusting retransmitting codewords with the suitable power level (i.e., adjusting the transmission power of the remaining attempts).

Secondly, a false negative decision is obtained, when the estimated value lies on the non-decodable state $(S_{M,k})$ at $k = 0$ and $M > 1$. This false prediction negatively affects the system's reliability, as the decodable block will be erroneously dropped at an early stage. The probability of this type of error can be determined as follows:

$$P_{FN} \equiv P(S_{M,k} = 0|M = i). \tag{4}$$

Accordingly, the cumulative probability for link packet error rate can be found:

$$PBLER = PPE.PFN, \tag{5}$$

where P_{PE} represents the packet error detectable empirically by the simulated model, and P_{FN} can be determined from the confusion matrix. This simple argument indicates the importance of adjusting the classifier's threshold to mitigate false-negative decisions (i.e., considering an effective-block error). This can be realized by designing a classifier with bias towards a false-positive decision to counterbalance a falsenegative decision. However, this strategy slightly reduces the efficiency of the proposed swift HARQ, but still can be considered due to the low probability of this error, as will be shown in Section IV.

Assumption 1 (Perfect Feedback Delivering): we assume that the ACK/NACK message can be successfully delivered on the uplink channel.

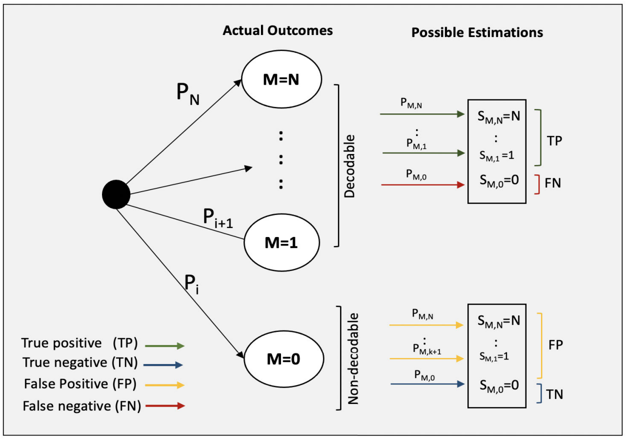


FIGURE 2. Probabilistic model for packet estimation scenario (the red arrows represent the effective block error).

Assumption 2 (Unlimited Resources): we assume infinite resources are available for retransmitting under a limited maximum number of attempts (N). Hence, there is no interruption in resources resulting from the upper layers or scheduling procedure.

B. LATENCY ANALYSIS

To evaluate the anticipated benefits of Swift HARQ, we need to carefully analyse the user plane latency. The chief components that comprise the user plane latency are the modem processing times, radio Transmission Time Interval (TTI), and the contribution of the HARQ retransmissions. Let us assume that f represents the delay of HARQ feedback on the backward direction (in channel uses), and $\Omega(L)$ denotes the delay to the transmission of the packet in the forward direction, including the delays that are caused by sending and decoding the codeword that has L channel use length. This delay critically depends on the coding scheme, which is not yet defined for URLLC. In traditional HARQ, the intended receiver sends acknowledgement feedback after every transmission to the BS, indicating the success/failure of the packet decoding process. If the attempt fails, the HARQ process will be performed repeatedly until the packet is successfully decoded at re-attempt j without exceeding the maximum allowable retransmission $N - 1$. The total latency can be expressed as:

$$\delta_{tr} = (j + 1)(L + \Omega + f). \tag{6}$$

On the other hand, when Swift HARQ is enabled, the intended receiver may react faster, falling under one of the following scenarios.

In the first case, a NACK is received that prompts the sender to drop the non-decodable packet before the remaining attempts are anticipated. The latency in this case can be formulated as follows:

$$\delta_{Swift} = L + \Omega + f \tag{7}$$

The second case occurs when Swift HARQ predicts the decodability of the packet at attempt j, where $j \in 2, \ldots, N$. The latency can then be formulated as follows:

$$\delta_{Swift} = \begin{cases} (2(L + \Omega + f) & \text{if j} = 2 \\ (j + 1)(L + \Omega) + 2f & otherwise. \end{cases} \tag{8}$$

In this case, the delay reduction is based on the receiver omitting $(j - 1)$ feedback message(s) as a result of activating repetition mode. However, the maximum gain from our proposed scheme becomes significant at low/moderate SNRs, in which the high probability of required retransmissions is observed. Meanwhile, the probability of decoding the message in the first round increases at high SNRs, potentially minimizing the performance gains from employing Swift HARQ.

C. POWER ALLOCATION POLICY

In this section, we conduct an initial discussion on a variable power allocation strategy. Traditionally when feedback information besides the ACK/ NACK is limited, the only option is to allocate a fixed power level for each round. On the other hand, the proposed Swift HARQ mechanism can proactively estimate the required number of re-attempts to decode the signal successfully. This information can be exploited to adapt the remaining scheduled re-attempts with the channel conditions, and that is through allocating varying power levels for each round. To gain deep insight, let us define N, as the maximum number of transmission attempts, Φ_n, as the probability that the packet is not correctly decoded until the *nth* transmission attempt, P_n, as the transmission power employed in the $n-$ *th* retransmission round, and P_{budget} as the consumed energy to transmit the whole packet, respectively. In the case of a fixed power allocation strategy, the transmit power in the *nth* round would be $P_n = \frac{P_{\text{budget}}}{N}$, and the maximum power budget of a packet transmission can be found as follows:

$$P_budget = \sum_{n=1}^{N} Pn\Phi n - 1. \tag{9}$$

In [22], a power allocation algorithm is investigated to minimize P_{budget} in the finite block-length regime by utilizing Karush-Kuhn-Tucker (KKT) conditions. The authors in [23] optimized the allocated power for each round P_n in order to minimize the outage probability under the maximum power budget constraint. In this work, the aim is to decrease the number of scheduled retransmission rounds, which can minimize the round trip time and reduce the outage probability. To this end, we proposed a decremental power allocation approach based on early HARQ feedback, according to which the power allocation values begin with a high weight, which then gradually decreases based on the channel conditions, while respecting the maximum power budget. We experimentally found that the following expression of the power allocation strategy for the remaining retransmission attempts results in improved performance in reducing the link block error rate:

$$P_j = \begin{cases} P_t\sqrt{\left(\frac{J}{2^j}\right)} & 1 < j < J \\ P_t\sqrt{\frac{J}{2^J-1}} & j = J \end{cases} \tag{10}$$

where J is the maximum number of retransmission attempts, P_j is the allocated power for the jth retransmission attempt, and P_t is the allocated power in the traditional method. We also assume that the allocated power level at the highest weight will not exceed the maximum allowable power transmission. Accordingly, the total consumed energy for the whole packet can be found:

$$P_{\text{budget}} = P_t + \sum_{j=1}^{J-1} P_t\sqrt{\left(\frac{J}{2^j}\right)} + P_t\sqrt{\frac{J}{2^{J-1}}}\Bigg). \tag{11}$$

III. SYSTEM MODEL

In order to derive the necessary tools for the evaluation of the performance of the proposed protocol, we study a sophisticated system that is inspired by the structure of 3GPP 5G NR Release 15. Let us define the single URLLC packet as the transmission of a codeword alongside all possible HARQ-based retransmission rounds. The packet travels on the downlink shared channel (DSCH) throughout a point-topoint multiple-input multiple-output (MIMO) system, which has Nt transmitter and Nr receiver antennas, respectively. Each user is allocated with L orthogonal groups of assigned Resource Blocks (RBs). Each RB spans orthogonality for S_f subcarriers over the frequency domain and S_t consecutive OFDM symbols over the time domain. Unlike the legacy LTE, this duration is varying and depends on subcarrier spacing types (i.e., 15, 30, 60, and 120 kHz), offering a higher degree of flexibility to satisfy the diverse requirements of 5G use cases. Hence, each packet contains a total of $P = L\times\ S_f\ \times S_t$ symbols and has a time duration referred to as TTI. The number of bits inside each URLLC packet will be encoded to a codeword via an LDPC encoder. Each coded bit on the codeword is mapped to a symbol using an appropriate modulation scheme (i.e., QPSK, 16-QAM). Then, the reference symbols will be inserted to the modulated codeword S while processing the other transmitter functions, such as MIMO precoding and resource mapping.

In this way, the channel input-output can be expressed as follows:

$$Y = SH + W, \tag{12}$$

where $S \in C^{SfStNt}$ and $Y \in C^{SfStNr}$ are the transmitted and received signals, respectively; $W \in C^{SfStNr}$ denotes the additive white Gaussian noise (AWGN), which has independent $W \sim CN\,(0, 1)$. Finally, $H \in C^{NtNr}$ is the fading matrix that is assumed to support clustered delay line (CDL) profiles, as it is envisioned by 3GPP for 5G communication systems in frequencies from 0.5 to 100 GHz [24].

A. DATA COLLECTION AND PREPROCESSING

The proposed Swift HARQ mechanism relies on a supervised learning principle to estimate the decodability outcomes; The idea is to train a learning model to find a hypothesis F that can separate the decodability labels to their potential classes using pre-collected samples. Each sample represents the observation of each codeword alongside all possible HARQbased retransmission rounds. Since URLLC standardization activities are still in progress, no suitable public dataset is available for our research. Therefore, we used MATLAB to obtain the training and test dataset from the described above data transmission model. The link-level parameters are summarized in Table 2. It is essential to utilize practical features that contain highly correlated information to the decodability status. In this study, we extract two key features that can infer useful information about the channel conditions. The first feature is BER_L, which is based on determining the probability of error estimating the log-likelihood ratio for the received-coded signal ahead of the decoding process. The extraction strategy starts by finding the log-likelihood ratio of the k^{th} received bit b_k is either 1 or 0, as expressed below:

$$Lo\,(b_k) = \log\frac{P\,(b_k = 1 \mid y)}{P\,(b_k = 0 \mid y)} \tag{13}$$

Following that, the probability of each bit being incorrectly estimated is calculated as follows [25]:

$$P_w\,(b_k) = \frac{1}{1 + |Lo\,(b_k)|} \tag{14}$$

Finally, the feature can be extracted by finding the average error rate for each codeword as follows:

$$BER_L = \frac{1}{L}\sum\frac{1}{1+P_w}. \tag{15}$$

The second feature (CSI) is derived by the channel state estimation. When the UE receives the transmitted signal affected by complex channel gain and noise, the known pilot symbols at both parties can be utilized to estimate the channel conditions and then equalize the channel's effects on the received signal. The instantaneous channel estimation for a subset of Resource Elements (REs) within a subframe can be computed using the least-squares estimate [26] as follows:

$$H_p(k) = \frac{Y_p(k)}{X_p(k)} \tag{16}$$

where $H_p(k)$ represents the channel response for the RE occupied by the pilot symbol, $Y_p(k)$ is the received pilot symbol, and $X_p(k)$ represents the known transmitted pilot symbol, the proposed feature is after that extracted by computing the average of estimated pilots within a codeword. Finally, The dataset labels (i.e., the decodability outcomes) are collected after the completion of all receiving tasks, including the process of checking the integrity of the decoded bits. It is essential to mention that estimating the early feedback will not overwhelm the receiver with significant processing, as the estimating relies on features that are already processed as part of the reception functions in the traditional link.

B. CLASSIFICATION ALGORITHMS

The Support Vector Machines (SVM) classifier was selected to solve the classification problem in this study, since it shows excellent generalization performance in processing high-dimensional data with the presence of irrelevant features that share overlapped classes [27]. This section explains the basic theory of SVM and its relevance to implementation and refers to [28], [29], and [30] for more detailed theories and complete formulation. Consider the following training samples: $\{x_1, y_1\}, \ldots, \{x_V, y_V\}$, where $x_i \in R^m$ is an m-dimensional feature vector representing the *ith* training sample, and y_i is the corresponding class label. Equation $w^T xi + b = 0$ describes a hyperplane in the feature space, where $w \in R^m$ is a weight vector and b is a threshold. The SVM is trained to construct an optimal hyperplane that acts as a class descriptor between two different classes. The final goal is to accurately represent decision boundaries that clearly separate the training data without under-fitting or over-fitting. However, the problem in this study can be seen as a non-linear case, as the training samples cannot be linearly separable. For such a case, the method of ''kernel trick'' is employed to map the samples into a much higher dimensional space where the optimal hyperplane can be determined. In other words, the Kernel function allows the mapping of vectors from input space to feature space and computes the inner product between two vectors [30]. The present study uses the SVM polynomial kernel, which has d degree polynomials, as shown below:

$$K(x_i, x_j) = (x_i^T x_j + 1)^d. \tag{17}$$

Following the representation of the samples in highdimensional space, the decision boundary for SVM can be constructed using the following discriminant function:

$$f(x) = \sum\nolimits_{i=1}^{V} \alpha_i y_i K(x_i, x_j) + b \tag{18}$$

To find the parameter pair (α_i, b) corresponding to the optimal hyperplane that can maximize the margin between the closed samples and the hyperplane, the following optimization problem shall be solved as:

$$\begin{aligned} &\min: \; L(w, \xi_i) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{V}\xi_i \\ &\text{S.t.: } y_i\left(\alpha_i y_i k\left(x_i, x_j\right) + b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0, \; i = 1, \ldots V, \end{aligned} \tag{19}$$

where C is a regularization parameter for the trade-off between model complexity and training error, and ξ_i is a slack variables that is related to the soft margin [31]. The above discussion comprises the fundamentals of binary SVM. However, our classification problem is considered as a multi-class problem, where the decodability outcomes can lie on one of M classes, where the label $y_i \in 0, \ldots, N$. Several methods have been proposed to extend binary SVMs to solve multi-classification problems. The one-Against-All method is considered as the earliest extension of binary SVM. The key concept is to construct M binary SVM classifiers. Each SVM model is constructed to discriminate the *jth* class against the other $M - 1$ classes, where all the samples in the *jth* class are indexed with positive labels and all other samples with a negative label. The desired class y_s for each training sample x_i is defined as follows [32]:

$$y_s = \begin{cases} +1 & \text{if } y_j = j \\ -1 & \text{if } y_j \neq j \end{cases} \tag{20}$$

Afterwards, all SVMs shall solve the optimization problem x that yields the *ith* decision function $f_i(x) = w^T x + b$. Finally at the classification phase, the ''winner-takes-all'' decision strategy is used, where sample x is classified as in class i^* for which f_i achieves the largest value as follows [33]:

$$i^* = \underset{i=1,\ldots,M}{\operatorname{argmax}} f_i(x) = \underset{i=1,\ldots,M}{\operatorname{argmax}} (w_i^T x + b_i). \tag{21}$$

Notably, this study has elaborated on the discussion of SVM, which offers superior performance compared to the logistic regression and decision trees. Details on theses classifiers are given in Appendix-A.

IV. PERFORMANCE ANALYSIS WITH MULTIPLE TRANSMISSION

In this section we evaluate the behavior of the proposed Swift HARQ mechanism when applying different ML classifiers. Moreover, we show the potential of the proposed mechanism compared to traditional HARQ. For this, we performed linklevel simulations using the simulation parameters that are summarized in Table 2. We applied 3 MM transmissions with independent channel realizations. As external parameters, we varied the SNR values under low/moderate channel quality conditions for two reasons. Firstly, erroneous reception is most likely to occur at low SNR; thus, early feedback will inevitably be more effective. Secondly, we can obtain a balanced dataset in this region, characterized by fair equality between decodability outcomes, i.e., the number of observations inside each decodable class being similar to the number of corrupted packets. This strategy aims to prevent the classifier from failing to manage an imbalanced training dataset and evaluating the classifier in a critical operational region. Meanwhile, we extend the evaluation of the proposed classification algorithms for Swift HARQ by finding the Area Under Curve (AUC) curves, which summarize the trade-off between the true-positive and false-positive rates for predictive models under different probability thresholds. These curves can help determine the suitable classifier with a desirable balance between the false positives and false negatives, hence, mitigating the costly false-positive predictions, which were explained previously in section III-A. It is essential to mention that no specific theoretical criteria exist to determine the optimal classifier for such a problem, therefore the classifier's determination takes place empirically based on observation of the models' response during the training and testing processes.

TABLE 2. System-level evaluation parameters.

Parameters	Description
Waveform	MIMO, CP-OFDM
Subcarrier Spacing	30, 60 KHz
Cyclic Prefix	Normal
Bandwidth	20 MHz
Channel Coding	LDPC, Rate: 2/5,5/6
Modulation	QPSK, 16-QAM
Codeword Size	480 Symbols
Channel Model	CDL-C (Urban Macro-cell) [24]
TX Antennas (N_t)	8
RX Antennas (N_r)	2
Equalizer	MMES
Decoding iterations	5
PDSCH Mapping Type	Singular Value Decomposition (SVD)

TABLE 3. Classification comparison between different ML algorithms.

	Accuracy	
Classification Method	**CR=2/5**	**CR=5/6**
Support vector machine (SVM)	0.93	0.91
Decision trees (DT)	0.91	0.90
Logistic regression (LR)	0.84	0.83

A. CLASSIFICATION PERFORMANCE

The classification strategy for the proposed model relies on two key features: (1) Log-likelihood ratios (BER_L) and (2) Channel State Information (CSI). Fig.3(a) represents the distribution of these features with respect to the corresponding decodability outcomes. To obtain a reliable classifier to classify these features into their potential classes, we train the ML model using different classifiers, including Decision Trees (DT), Logistic Regression (LR), and Support Vector Machines (SVM). The trained models' accuracy is initially validated by finding the ratio for the number of correct predictions to the total number of input samples.

This ratio characterizes the overall discriminatory power of the algorithm, which is typically close to 1 in the case of a perfectly discriminative classifier. As seen in Table.1, the accuracy of the classifiers examined across different SNR-values reveals the superiority of SVM, closely followed by DT. In addition, we extend observations of such behavior by varying the channel coding rate (CR) to 5/6. It appears in Table. 2 that the classifiers' performance is effective, and that SVM still maintains a lead.

It should be noted here that accuracy evaluation metrics can only reflect mismatched predicted outcomes with actual outcomes, which can produce false-negative or false-positive decisions. However, as Swift HARQ is more sensitive to false-negative decisions, we specifically need to determine the appropriate classifier that combines mitigating the falsenegative decisions and enhancing the classification accuracy. To this end, we evaluated the behavior of the classifiers using the Receiver Operating Characteristic (ROC) curve, as shown in Fig.4. The Y-axis represents TPR, which contains falsenegative decisions in its denominators, as follows:

$$TPR = \frac{TruePositive}{TruePositive + FalseNegative}. \tag{22}$$

TABLE 4. Area under the curve (AUC) metric for different classifiers.

Classifier	AUC
Support Vector Machine	0.998
Decision trees	0.995
Logistic Regression	0.994

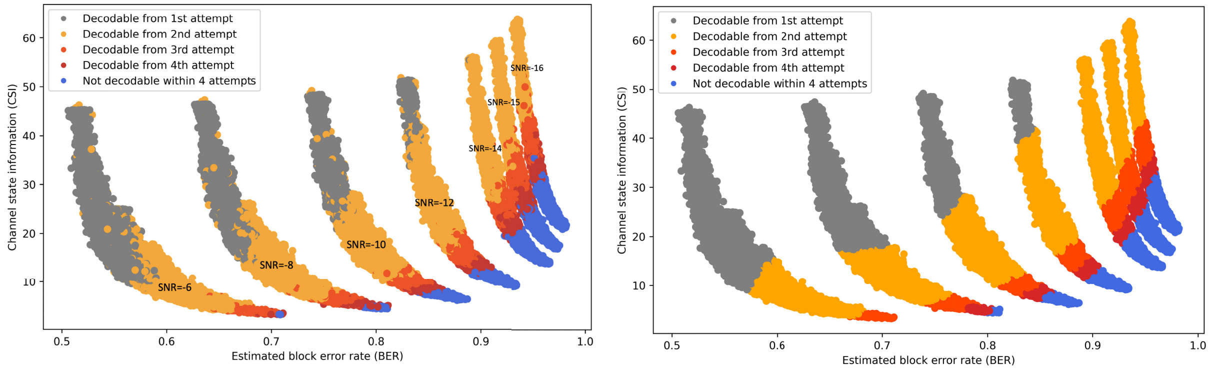
Hence, the main goal is to maximize the TPR so that it is close to one, by minimizing the false-negative scores. Notably, SVM rapidly approached one at low FPR. Meanwhile, logistic regression generally produces good overall performance, but is considerably weaker than other classifiers in terms of the TPR.

We also evaluate the AUC scores to summarize the ROC curve. The scores lie between 0.5 (for random prediction) to 1.0 (for a perfect score). As illustrated in Table. 3, the AUC is scored highly for all classifiers, and the SVM is the best of the three.

To explain the exceptional performance of SVM, we visualized the dataset distributions after the classification in Fig.3(b), and compared it with Fig.3(a). As demonstrated, false-negative decisions can occur in the overlapped zone between the red and blue classes, which represent misestimations of decodable packets within three rounds (red class) to create a non-decodable packet (blue class). However, SVM overcomes this problem with a bias towards the blue class, as shown from that decision boundary. This biasing intuitively comes at the expense of the false-positive decision. However, unlike the false-negative decision, such false decisions have no impact on the link BLER performance, as the impact will be limited to generating unnecessary transmissions. To summarize, the simulation results show that SVM, DT, and LR have similar overall performance in terms of classification accuracy, despite their fundamentally different underlying principles. However, SVM maintains the lead in terms of accuracy performance, thereby mitigating false-positive decisions.

The next step is to attain deep insight into the URLLC operation region at SNR $= -6$. We studied the behavior of the dataset in order to mitigate false positive decisions and overcome the imbalanced dataset behavior, thereby obtaining the full benefit from employing the Swift HARQ mechanism. It is notable from Fig.3(a) that the samples have a heavily imbalanced behavior, and this is evident in the left-most cluster of collected samples in SNR $= -6$. There are only two non-decodable samples belonging to class 5, and the SVM failed to predict these outcomes, as seen in Fig.3(b).

To better understand this behavior, we calculated the confusion matrix for SVM, as seen in Fig.5, which summarizes



(a) The dataset before classification.

(b) The dataset after classification.

FIGURE 3. Visualization of the test dataset before and after classification using Support Vector Machine. QPSK, CR = 5/6, N = 4.

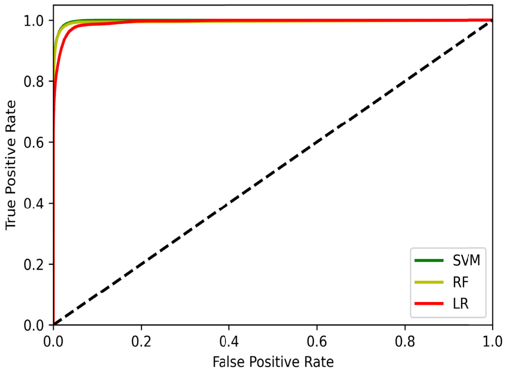


FIGURE 4. Selected example for classification performance based on AUC in the CDL channel.

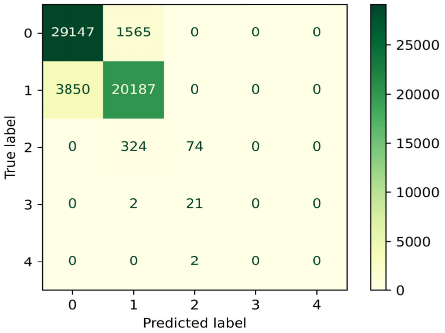


FIGURE 5. Confusion Matrix for Swift HARQ using SVM.

the classification results by showing the number of predictions per class. The rows represent the true labels for the data points, the columns represent the predicted labels, the diagonal cells (from top left to bottom right) represent accurate predictions, and the all off-diagonal cells represent incorrect classifications.

Let us examine, for example, the rows from three to five in Fig.5. The majority of the samples in the third row belonging to Class 2 (324 data points) were misclassified as Class 1, and all data points in Class 3 were misclassified as either Class 1 (2 data points) or Class 2 (21 data points), and lastly, all the data points in Class 4 were misclassified as Class 2.

TABLE 5. Distribution of samples by classes for SNR = −6.

Class	Number of samples
0	153936
1	119855
2	1891
3	168
4 (non-decodable)	2

The situation described above is an example of an extreme class imbalance. Table. 4 depicts the distribution of all the data points by class for the SNR = −6 dataset. Note that, while classes 0 and 1 comprise over over 99% of the total number samples, classes 2, 3, and 4 combined share (not equally) the remaining number of samples, which is less than 1% of the total. Such extreme class imbalance, if left unaddressed, can result in the classifier having too few samples from the minority classes to learn to predict them effectively. Thus, the classifier will always have inferior performance across the minority classes leading to more false-positive decisions.

The two main approaches for addressing class imbalances are under-sampling (of the majority classes) and oversampling (of the minority). A popular technique used when under-sampling the majority class involves randomly selecting data points to be excluded from the training set. On the other hand, a popular technique for over-sampling the minority class is the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic data points to complement existing data points in a minority class. SMOTE works by selecting a random data point from the minority class, choosing k of the nearest data points to the selected data point, and creating a new data point between the selected data point and a random one of the k neighbors in feature space. This process is repeated until a specified number of synthetic samples have been created.

Given the extreme class imbalance presented in Table. 4, over-sampling of the minority classes (Classes 2 and 3) using SMOTE was essential to improve classification performance. Note that class 4 presents a problem for SMOTE, as relying

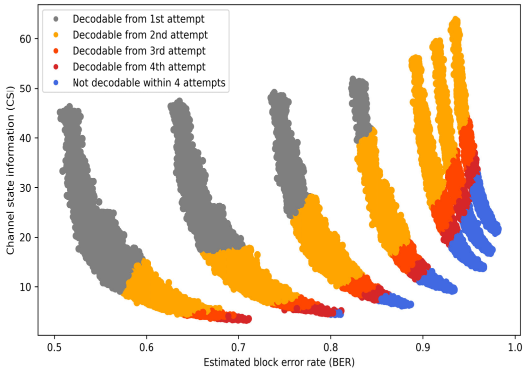


FIGURE 6. Visualization of the test dataset after classification using data augmentation.

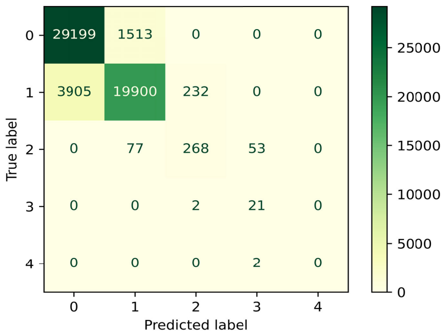


FIGURE 7. Confusion matrix for Swift HARQ after data augmentation using SVM.

on only two samples to generate more synthetic data will not provide diversity in the synthesized samples. However, this class can be neglected, as the probability of the packet lying on this class is very small, i.e., every 100,000 packet transmissions. Additionally, its occurrence will not impact overall the BLER performance, as the impact will only be limited to preventing the advantage of Swift HARQ.

Fig.6 visualizes the test dataset after classification using data augmentation, as see more data points classified as Class 3 (dark red points).

To clearly see the classification enhancement following data augmentation, we present the confusion matrix in Fig.7. As seen, the classification performance of Classes 2 and 3 improved compared to Fig.5, where the majority of the data points of Class 2 (269 data points) have been correctly classified, with 76 data points misclassified as Class 1, and 53 data points misclassified as Class 3. This is a significant improvement compared to the only 74 Class 2 data points that were correctly classified prior to data augmentation.

Looking at the samples under Class 3, 21 samples were correctly classified, and only 2 data points were misclassified as Class 2. Again, this is a substantial improvement compared to the 0 Class 3 data points that were correctly predicted before data augmentation. These results show that data augmentation had a significant positive impact on the classification accuracy of SVM. Note that no data samples from classes 0 to 3 have been misclassified as Class 4, which was demonstrated by locating all the zeroes in the right-hand column.

The argument above shows the benefits of employing the SMOTE technique to enhance the accuracy of Swift HARQ by mitigating the false-positive decision, besides maintaining the prevention of the false-negative decisions.

To summarize, it is difficult to obtain a classifier that has an error-free performance, but we can balance these errors in line with the target application's requirements. To maintain the target BLER requirements, we concentrated on preventing negative errors as a top priority in this study. We then considered the possibility of mitigating positive-false errors to take advantage of Swift HARQ in enhancing link latency and throughput.

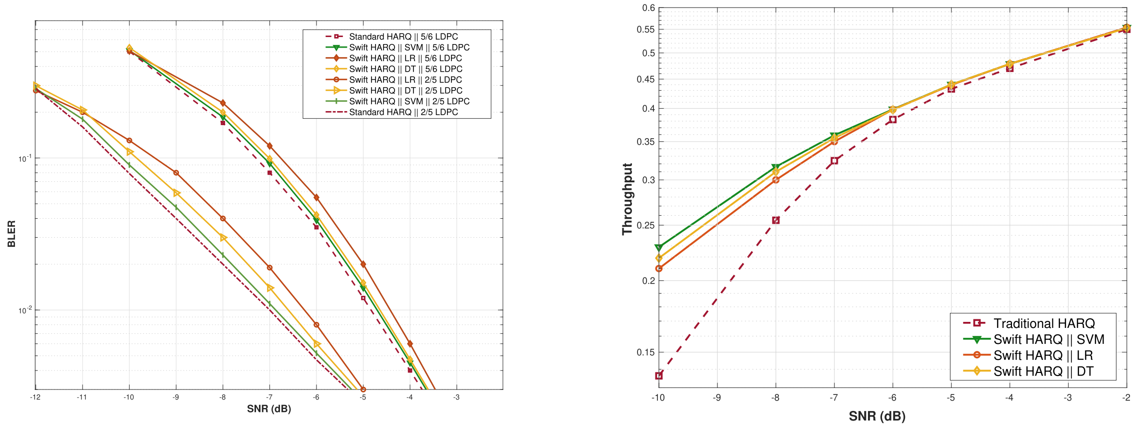
B. DATA LINK PERFORMANCE

Our next step was to evaluate the link performance under the assumption of the urban macro-cell model with CDL. Fig. 8(a) details the BLER as a function of the SNR for standard HARQ and Swift HARQ under different classification algorithms. Notably, applying SVM to the proposed model revealed a favorable performance comparative to standard HARQ, despite the decodability of the packet having been estimated ahead of completion of the HARQ processing. The performance of LR and DT is lower than SVM. It should also be noted that the performance regression started to expand negatively at SNR values between -9 dB and -6 dB. This behaviour can be attributed to the dataset collecting at critical SNR regions, where the variance of the extracted features becomes significantly close, despite the probability of their outcomes belonging to different classes. This can explain why LR and DT experience difficulties when handling overlapped classes at the edge of the decision boundary. Conversely, SVM has been more successful at dealing with an overlapped dataset behavior due to having applied the hyperplane principle, as described in section IV-B.

Meanwhile, the emerging gap in performance vanished for all classifiers at both the low and high SNR regions. This occurred because the majority of the decodability outcomes in these SNR regions have a monopolistic majority over one or two classes. In other words, the majority of the outcomes affecting low SNR are non-decodable. In contrast, the packet can be decodable within one or two attempts at high SNR. This can then facilitate the classifiers' task of distinguishing the correct outcomes among fewer classes.

Next, we studied the robustness of Swift HARQ by varying the channel coding rate to 5/6. We observed that while the BLER performance of all schemes dropped compared to the case of 2/5, the Swift HARQ performance for all the classifiers converged at a standard HARQ. This indicates the interaction of Swift HARQ with the varying coding rate, and therefore this difference in performance must be taken into account when choosing a robust classifier.

In Fig.8(b), we presented throughput performance versus SNR. Evidently, employing the Swift HARQ mechanism can yield considerable gains, with the SVM achieving



(a) Block error rate versus SNR under different coding rates.

(b) On the effect of Swift HARQ in the link throughput.

FIGURE 8. **Exemplary system performance comparison for Standard HARQ and Swift HARQ under different classification algorithms. 16 QAM, CR = 2/5, N = 4.**

100% performance improvement of the throughput at a moderate SNR (SNR $= -12$). The other classifiers also showed highly comparable performance. However, this enhancement depends substantially on the channel quality and becomes significant under poor channel conditions, as the non-decodable packet will be omitted early, increasing the availability of the channel use. However, the gain starts to decline in the high SNR regions, where the packet can be decoded in fewer attempts (i.e., before reaching the maximum allowable number of retransmissions), leading Swift HARQ to be less effective in saving resources.

Fig. 9(a) shows exemplary results detailing the expected delay (in channel use) as a function of SNR under the assumption of SubCarrier Spacing (SCS) = 60 kHz, QPSK, L = 480 and C = 0.5. Apparently, the transmission delay decreases for all techniques with the enhancement of the SNR. It is also notable that the maximum benefits when applying Swift HARQ can be observed with low/moderate SNR values, where disregarding the non-decodable packet or omitting the HARQ feedback early yields a significant delay in savings. However, the gap between the performance of Swift HARQ and traditional HARQ begins to converge at high SNR, as the number of packets requiring multiple HARQ retransmissions reduces when compared to the packets that can be decodable within 1 or 2 attempts. However, this behaviour does not diminish the importance of using Swift HARQ, as in the practical scenario, the UE may encounter a deep fading channel at high SNR, necessitating multiple retransmission rounds for the whole transmission session.

The effect of applying a variable power strategy is visible in Fig.9(a) at high SNR, where sensing the decodability status at an earlier stage enables the transmitter to detect the packet that requires multiple retransmission, hence increasing the power weight for the earliest rounds of retransmission. This action facilitates the decoding of packets with fewer retransmission rounds than traditional HARQ, thereby minimizing packet transmission duration.

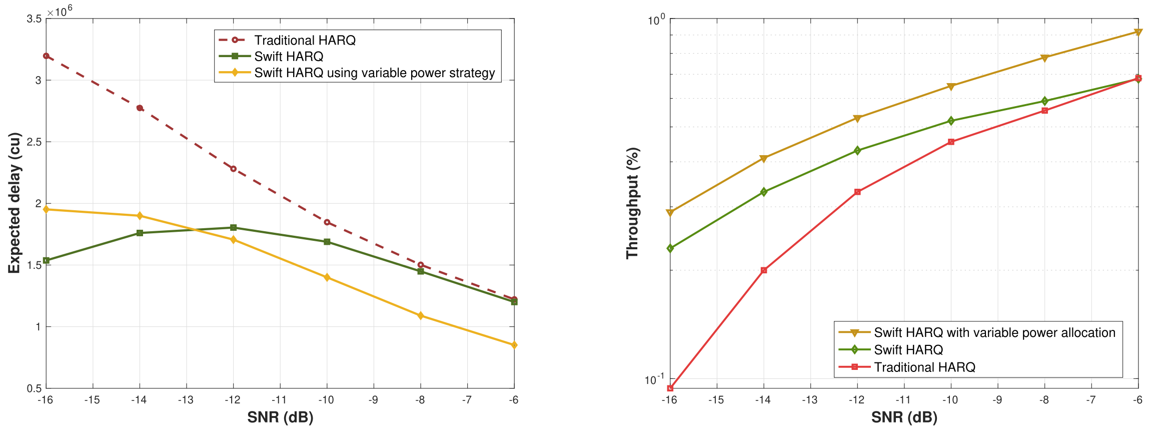
At low SNR, the variable power technique reveals a different behavior than classical Swift HARQ, where the transmission delay increases at this region. This is because the transmitter in classical Swift HARQ will drop nondecodable packets early. This led to delay reduction equivalent to the N-1 rounds. On the other hand, employing the variable power strategy can contribute to correcting some of the non-decodable packets, and thus a delay occurs when processing these packets at the expense of improving the system throughput. This finding is evident in Fig9(b), where the system throughput, at low SNR, is enhanced by 26% and 102% compared to Swift HARQ and traditional HARQbaseline, respectively.

Since the Swift HARQ estimation mechanism depends on a feature that relies on the channel status information, we extended the performance evaluation by taking the channel estimation error into consideration. As shown in Fig. 10, the overall system performance shows a similar trend, with an expected deterioration in BLER performance due to imperfect channel estimation. This indicated the feasibility of Swift HARQ, even when errors were present in the channel state estimation. It is notable also that Swift HARQ with the proposed variable power policy performs better than traditional HARQ. This is due to the ability of the new mechanism to adapt the remaining scheduled re-attempts with the appropriate power level. This can contribute to correcting some of the non-decodable packets, thereby enhancing the system's reliability.

We also evaluated the expected user-plane latency (in a duration of time) for each packet transmission, by accumulating the main components comprising the user plane delay. The total latency in the case of traditional HARQ can be written as:

$$T)_{tr} = T_{AG} + N(\Delta_{FW} + TTI_{Tx} + 2\rho + \phi_{UE} + TTI_{Rx} + \Delta_{BW}). \quad (23)$$

It is assumed that the processing time of 5G UE (ϕ_{UE}) is compliant with the 3GPP standard, known as processing



(a) The expected delay versus SNR. F=L/4, Ω= L/2.

(b) The link throughput versus SNR.

FIGURE 9. Exemplary system performance comparison for Standard HARQ and Swift HARQ using varied power allocation strategy. QPSK, CR = 5/6, N = 4.

capability 2 for low latency service [34], and the processing delay at the BS (Δ_{BW}, Δ_{FW}) is equal to half of the processing time at the UE, since the processing time of network elements is still not yet defined [34]. However, the round trip latency when the Swift HARQ successfully estimates the decodability outcome can be formulated as:

$$T_{Swift} = T_{AG} + N(TTI_{Tx} + \phi_{UE}) \\ + 2 * (TTI_{Rx} + \Delta_{FW} + \Delta_{BW}) + (N + 2)\rho. \quad (24)$$

This expression is based on the fact that the receiver will omit up to N-2 feedback message(s), and the assumption that the transmitter can handle processing the retransmission rounds in parallel. The comparison of the RTT for HARQ and Swift HARQ is illustrated in Table 6, assuming the packet required 3 retransmission rounds. In Fig.11, the effect of widening the subcarrier spacing is shown, where the RTT of packet transmission (i.e., including the forward and the backward directions) significantly decreases. This may lead to relaxing the latency constraint for all mechanisms with superiority in favor of Swift HARQ.

Next, we evaluated the round trip latency for different HARQ mechanisms by changing the number of retransmission rounds and setting the subcarrier spacing to 60 kHz. As seen from the bar chart in Fig.12, the variance in the round trip duration becomes evident as the number of retransmissions increases; consequently, this will have a significant impact on the link latency. It should be also noted that when the network allows only one HARQ retransmission at N = 2, E-HARQ [20] can meet the low latency requirement compared to Swift HARQ and traditional HARQ. This is attributed to the capability of E-HARQ mechanism to release the feedback ahead of the full signal decoding, which leads to the mitigation of the processing delay in the first round. We further notice an outstanding performance for Swift HARQ at higher numbers of retransmissions rounds, where omitting some handshaking processes plays a primary role in minimizing the latency. In contrast to the other two mechanisms, Swift HARQ has a steady performance with the increase of the number of retransmissions, as the handshaking activities will be performed twice regardless of the required number of retransmissions.

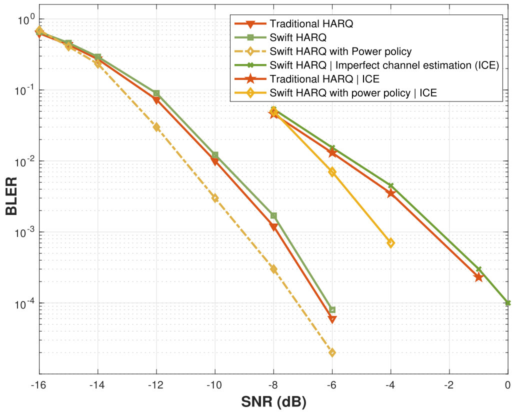


FIGURE 10. Block error rate versus SNR for Standard HARQ and Swift HARQ under imperfect channel estimation assumption. QPSK, CR = 5/6, N = 4.

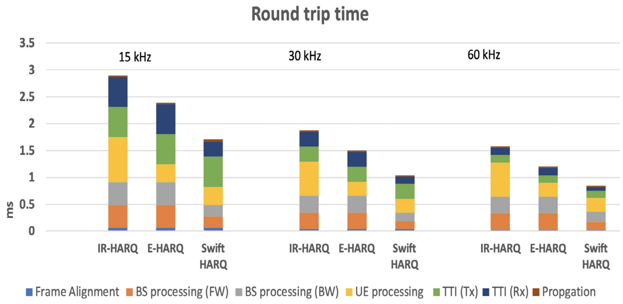


FIGURE 11. Comparing the two-way latency with different subcarrier spacing in the case of mini-slot transmission with UE capability 2. Assuming the link requires four HARQ attempts.

Summa summarum, the estimation of the packet decodability on a proactive basis causes, in general, a slight deterioration of the BLER performance. This is based on the

TABLE 6. Two-way latency with different subcarrier spacing.

Delay Component (μs)	Symbol	15 kHz	30 kHz	60 kHz
TTI duration	T_{TTI}	140	70	35
Frame Alignment	T_{AG}	70	35	17.5
BS Processing (FW)	$\triangle_{FW}$	105	87	93
BS Processing (BW)	$\triangle_{BW}$	105	95	95
UE Processing	Φ_{UE}	210	168	168
Propagation Delay	ρ	3	3	3
Total delay for HARQ (ms)		2.8	1.9	1.6
Total delay for Swift HARQ (ms)		1.7	1.00	0.85

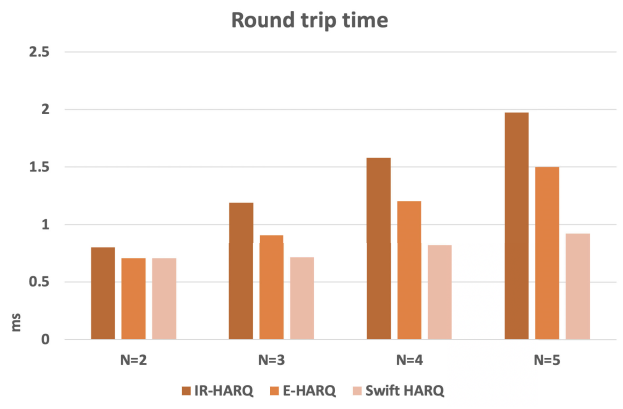


FIGURE 12. Comparing the two-way latency with different maximum numbers of transmissions in the case of 60 kHz and mini-slot transmission with UE capability 2.

fact that feedback is tightly dependent on the channel state at the first round, while this channel state may differ, either positively or negatively, from what was expected when all the re-transmission attempts are acquired. However, the proposed Swift HARQ mechanism demonstrates the capability of achieving a very comparative outage probability to the standard HARQ. This can be attributed to the advantage of enabling the ML techniques that show a good ability to be trained to remember and analyze the characteristics of sophisticated wireless channels, then utilized on-board to sense the decodability status proactively. The proposed scheme also consistently outperforms the standard HARQ in terms of throughput and expected delay. The maximum benefit becomes evident at low/moderate SNRs, where it is very common to decode the packet with multiple HARQ processing or failure to decode the packet. It is noteworthy to mention that the need for multiple HARQ processing is less present at high SNRs due to the fact that the number of packets that can experience a deep fading is minor in the simulated environment. However, the channel conditions in the real field may differ from the simulated environment, especially when the whole transmission session exposes to deep fading conditions and therefore needs multiple retransmissions frequently, which could lead to exceeding the required latency in URLLC. This behavior triggers the vital need for early feedback-based mechanisms at high SNR to counteract the effects of deep fading distortion and maintain the URLLC latency requirements. It is crucial to mention that employing ML mechanisms can deliver outstanding accuracy but comes at the expense of high computational complexity, demanding a lengthy training period. However, in our application, the complexity associated with the training task is less critical, as Swift HARQ can be pre-trained offline to determine its optimal ML network parameters. The trained model can be employed on radio devices (i.e., on-board) without any training overhead during the signal detection. We also emphasize the positive relation between the packet delivery duration and channel coherence with the context of URLLC. The packet delivery is target to achieve 1 ms; this duration, in most scenarios, is shorter than the duration of the channel coherence. Thus, we believe that employing early HARQ techniques aligns with this unique characteristic as the constancy of channel coherence facilitates the ability to predict the channel conditions on a proactive basis, and thus the rapid response by sending all the required packets without waiting for more exchange handshaking signaling.

V. CONCLUSION

This paper studied the performance of HARQ in multiple retransmitting scenarios. We proposed the Swift HARQ mechanism empowered by machine learning techniques to estimate early feedback about the decodability of a packet. This allows the transmitter to stop re-transmission activities for the non-decodable packets or to activate the repetition mode where some handshaking overhead can be eliminated. The proposed model reveals a considerable latency reduction that reaches 60%, 90%, and 110%, compared to the stateof-art HARQ mechanism with 3, 4, and 5 re-transmission rounds, respectively. Moreover, the results show a significant improvement in the link throughput, with the SVM achieving double throughput at low channel conditions. In addition, we coupled the feedback with an estimation of the required number of attempts to decode the packet successfully, then exploited this information to allocate the suitable power that can reduce the processing latency and guarantee at least the achievement of the target outage probability. Our future plans include studying the capacity to re-tune the ML models onboard, specifically when the trained model experiences new channel conditions that do not match the channel models considered during training, a situation referred to as model drift. In such cases, model estimations are no longer as accurate as they were in test tests. There are open questions for research in this area, including how the appropriate deviation is determined in URLLC and how frequently the model should be retrained. From the resource allocation perspective, this study has demonstrated the efficacy of using early feedback mechanisms in terms of minimizing the transmission delay and increasing throughput. However, employing these techniques may result in some minor changes regarding traffic scheduling and resource allocation. There is a paucity of studies in the literature addressing the impact of these modifications on the network performance, especially in the context of multiplexing the URLLC with eMBB.

APPENDIX

A. DECISION TREES

Decision trees are a type of tree-based learning algorithm that make decisions by partitioning the instance space into a set of regions determined by the feature values of the training dataset. In general, given a data of N observations, where each observation consists of a m dimensional input vector and a response label (x_i, y_i) for $i = 1, 2, \ldots, V$ and $y_i \epsilon 1, 2, \ldots, V$, a tree inducer needs to decide on a splitting feature and splitting point at each node. These are determined based on some cost metric based on the proportion of classes at the node. For a node m representing a region R_m with N_m observations. These are determined based on some cost metric based on the proportion of classes at the node. Examples of common cost measures for a node include:

$$Gini\ index: \sum_{k=1}^{K} Pt_{mk}(1 - Pt_{mk}) \tag{25}$$

$$Cross\ entropy: \sum_{K=1}^{K} Pt_{mk}\ log(Pt_{mk}), \tag{26}$$

where Pt_{mk} is the proportion of training set observations of class k in region R_m, and is calculated using the following equation:

$$Pt_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m}^{K} I\left(y_i = k\right). \tag{27}$$

The classification for the instances in any region R_m is defined as the class with the majority proportion in R_m given by the equation:

$$k(m) = argmax_k\ Pt_{mk}. \tag{28}$$

B. LOGISTIC REGRESSION

Logistic regression is inherently a binary classification algorithm. However, a general multinomial expression of Logistic regression that extends to K classes is also available. For a total of K classes, the multinomial logistic regression model is expressed in terms of $K - 1$ log-odds transformations, which satisfy the constraint that the probabilities for all classes sum up to 1. The multinomial expression is provided in following two equations [35].

$$p(k) = \frac{e^{-(a_k + \beta_k x)}}{1 + \sum_{l=0}^{K-1} e^{-(a_l + \beta_l x)}} \tag{29}$$

where $p(k)$ is the probability of $y = k$, $k \epsilon (1, 2, \ldots, K)$.

REFERENCES

[1] C. She, C. Yang, and T. Q. S. Quek, "Cross-layer optimization for ultra-reliable and low-latency radio access networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 127–141, Jan. 2018.

[2] F. E. Airod, H. Chafnaji, and H. Yanikomeroglu, "HARQ in full-duplex relay-assisted transmissions for URLLC," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 409–422, 2021.

[3] M. Series, "Minimum requirements related to technical performance for IMT-2020 radio interface (S)," ITU, Geneva, Switzerland, Tech. Rep. 2410-0, 2017.

[4] S. Dogan, A. Tusha, and H. Arslan, "NOMA with index modulation for uplink URLLC through grant-free access," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 6, pp. 1249–1257, Oct. 2019.

[5] S. Ashraf, Y. E. Wang, S. Eldessoki, B. Holfeld, D. Parruca, M. Serror, and J. Gross, "From radio design to system evaluations for ultra-reliable and low-latency communication," in *Proc. Eur. Wireless 23th Eur. Wireless Conf.*, May 2017, pp. 1–8.

[6] B. Husain and A. Czylwik, "Channel coding and low latency HARQ for industrial wireless sensor networks," in *Proc. Wireless Days (WD)*, Apr. 2019, pp. 1–5.

[7] J. Cao, X. Zhu, Y. Jiang, Y. Liu, and F.-C. Zheng, "Joint block length and pilot length optimization for URLLC in the finite block length regime," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[8] T.-K. Le, U. Salim, and F. Kaltenberger, "An overview of physical layer design for ultra-reliable low-latency communications in 3GPP releases 15, 16, and 17," *IEEE Access*, vol. 9, pp. 433–444, 2020.

[9] S. H. Kim, D. K. Sung, and T. Le-Ngoc, "Performance analysis of incremental redundancy type hybrid ARQ for finite-length packets in AWGN channel," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 2063–2068.

[10] B. Makki, T. Svensson, and M. Zorzi, "Green communication via type-I ARQ: Finite block-length analysis," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 2673–2677.

[11] H. Seo and B. G. Lee, "Optimal transmission power for single- and multi-hop links in wireless packet networks with ARQ capability," *IEEE Trans. Commun.*, vol. 55, no. 5, pp. 996–1006, May 2007.

[12] T. V. K. Chaitanya and E. G. Larsson, "Outage-optimal power allocation for hybrid ARQ with incremental redundancy," *IEEE Trans. Wireless Commun.*, vol. 10, no. 7, pp. 2069–2074, Jul. 2011.

[13] S. R. Khosravirad, K. I. Pedersen, L. Mudolo, and K. Bakowski, "HARQ enriched feedback design for 5G technology," in *Proc. IEEE 84th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2016, pp. 1–5.

[14] J. P. B. Nadas, O. Onireti, R. D. Souza, H. Alves, G. Brante, and M. A. Imran, "Performance analysis of hybrid ARQ for ultra-reliable low latency communications," *IEEE Sensors J.*, vol. 19, no. 9, pp. 3521–3531, May 2019.

[15] B. Makki, T. Svensson, G. Caire, and M. Zorzi, "Fast HARQ over finite blocklength codes: A technique for low-latency reliable communication," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 194–209, Jan. 2019.

[16] G. Berardinelli, S. R. Khosravirad, K. I. Pedersen, F. Frederiksen, and P. Mogensen, "Enabling early HARQ feedback in 5G networks," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, May 2016, pp. 1–5.

[17] B. Goektepe, S. Faehse, L. Thiele, T. Schierl, and C. Hellge, "Subcode-based early HARQ for 5G," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[18] N. Strodthoff, B. Göktepe, T. Schierl, C. Hellge, and W. Samek, "Enhanced machine learning techniques for early HARQ feedback prediction in 5G," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2573–2587, Nov. 2019.

[19] S. AlMarshed, D. Triantafyllopoulou, and K. Moessner, "Supervised learning for enhanced early HARQ feedback prediction in URLLC," in *Proc. IEEE Int. Conf. Commun., Netw. Satell. (Comnetsat)*, Dec. 2020, pp. 26–31.

[20] S. AlMarshed, D. Triantafyllopoulou, and K. Moessner, "Deep learning-based estimator for fast HARQ feedback in URLLC," in *Proc. IEEE 32nd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2021, pp. 642–647.

[21] *Release Description; Release 15, 3rd Generation Partnership Project; Technical Specification Group Radio Access Network*, Standard TR21.915, 3GPP, 2011.

[22] E. Dosti, M. Shehab, H. Alves, and M. Latva-Aho, "Ultra reliable communication via optimum power allocation for HARQ retransmission schemes," *IEEE Access*, vol. 8, pp. 89768–89781, 2020.

[23] L. Vangelista and M. Centenaro, "Performance evaluation of HARQ schemes for the Internet of Things," *Computers*, vol. 7, no. 4, p. 48, Sep. 2018.

[24] *Study on Channel Model for Frequencies From 0.5 to 100 GHz*, 3GPP TSG RAN, TR 38.901, 3GPP, 2020.

[25] G. Berardinelli, S. R. Khosravirad, K. I. Pedersen, F. Frederiksen, and P. Mogensen, "Enabling early HARQ feedback in 5G networks," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, May 2016, pp. 1–5.

[26] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in OFDM systems," *IEEE Trans. Broadcast.*, vol. 48, no. 3, pp. 223–229, Sep. 2002.

[27] B. Ghaddar and J. Naoum-Sawaya, "High dimensional data classification and feature selection using support vector machines," *Eur. J. Oper. Res.*, vol. 265, no. 3, pp. 993–1004, Mar. 2018.

[28] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[29] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Jul. 1995, doi: 10.1023/A:1022627411411.

[30] V. Vapnik and V. Vapnik, "Statistical learning theory," New York, NY, USA, Springer-Verlag, Tech. Rep. 1998.

[31] Y. Liu and Y. F. Zheng, "One-against-all multi-class SVM classification using reliability measures," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2, Jul. 2005, pp. 849–854.

[32] J. Blanc-Talon, W. Philips, D. Popescu, and P. Scheunders, *Advanced Concepts for Intelligent Vision Systems*. Berlin, Germany: Springer, 2007.

[33] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2001.

[34] H. Holma, A. Toskala, and T. Nakamura, *5G Technology: 3GPP New Radio*, 2nd ed. Hoboken, NJ, USA: Wiley, 2019.

[35] T. Hastie, R. Tibshirani, and J. Friedman, "Model assessment and selection," in *The Elements of Statistical Learning*. Cham, Switzerland: Springer, 2009, pp. 219–259.

SALEH ALMARSHED (Member, IEEE) received the bachelor's degree in electrical and communication engineering from King Saud University, Riyadh, Kingdom of Saudi Arabia, in 2010, the master's degree from the University of Central Florida, Orlando, USA, in 2018, and the Ph.D. degree from the Department of Electrical Engineering, University of Surrey, in 2022. He has been a Senior Communication Systems Researcher with the King Abdulaziz City for Science and Technology (KACST), since 2010. During his employment, he has been involved in several industrial and national research and development projects sponsored by leading companies such as Lockheed Martin, Surrey Satellite Technology Ltd., and Aselsan. His current research interests include machine learning and its applications in telecommunication.

DIONYSIA TRIANTAFYLLOPOULOU (Member, IEEE) received the B.Sc. degree in computer science, the M.Sc. degree in communication systems and networks, and the Ph.D. degree from the Department of Informatics and Telecommunications, University of Athens, Athens, Greece, in 2005, 2007, and 2009, respectively. She was a Researcher at the University of Athens, from 2005 to 2011. She is currently a Senior Research Fellow with the Institute for Communication Systems, University of Surrey, U.K.

KLAUS MOESSNER (Senior Member, IEEE) is currently a Professor of communications engineering with the University of Technology Chemnitz and a Professor of cognitive networks with the Institute for Communication Systems and the 5G Innovation Centre, University of Surrey. He was involved in a large number of projects in the cognitive communications, service provision, and the IoT areas. He was responsible for the work on cognitive decision making mechanisms in the CR Project ORACLE, and led the work on radio awareness in the ICT FP7 Project QoSMOS, led the H2020 Speed5G Project. His research interests include cognitive networks, the IoT deployments, and sensor data-based knowledge generation, as well as reconfiguration and resource management. He does lead the EUTaiwan Project Clear5G investigating the extensions 5G systems need to serve the particular requirements of the Factories of the future. He was the Founding Chair of the IEEE DYSPAN Working Group (WG6) on Sensing Interfaces for Future and Cognitive Communication Systems.