Enhanced Machine Learning Techniques for Early HARQ Feedback Prediction in 5G

Nils Strodthoff, Barış Göktepe, Thomas Schierl, Cornelius Hellge[iD], and Wojciech Samek[iD], *Member, IEEE*

Abstract—We investigate Early Hybrid Automatic Repeat reQuest (E-HARQ) feedback schemes enhanced by machine learning techniques as a path towards ultra-reliable and low-latency communication (URLLC). To this end, we propose machine learning methods to predict the outcome of the decoding process ahead of the end of the transmission. We discuss different input features and classification algorithms ranging from traditional methods to newly developed supervised autoencoders. These methods are evaluated based on their prospects of complying with the URLLC requirements of effective block error rates below 10^{-5} at small latency overheads. We provide realistic performance estimates in a system model incorporating scheduling effects to demonstrate the feasibility of E-HARQ across different signal-to-noise ratios, subcode lengths, channel conditions and system loads, and show the benefit over regular HARQ and existing E-HARQ schemes without machine learning.

Index Terms—5G, mobile communication, low latency communication, physical layer, machine learning, anomaly detection, deep learning.

I. INTRODUCTION

THE next generation Fifth Generation (5G) wireless mobile networks is driven by new emerging use cases, such as Ultra-Reliable Low Latency Communication (URLLC) [1]. URLLC applications such as tactile Internet, industrial automation or smart grids contribute to increasing demands on the underlying communication system which have not existed as such before [2]. Depending on the actual application either very low latency or high reliability or a combination of both are required. In contrast to Long Term Evolution (LTE), where services were provided in a best effort manner, 5G networks have to guarantee these requirements. In particular for URLLC, the ITU proposed an end-to-end latency of 1 ms and a packet error rate of 10^{-5} [3]. These demanding requirements have kicked-off discussions in the 3GPP Rel. 16 standardization process on how to fulfill these. Self-contained subframes and grant-free access have been

proposed to address these requirements on the air interface side [4]. However, the impact on well-known mechanisms in wireless mobile networks is still unclear. In particular, the Hybrid Automatic Repeat reQuest (HARQ) procedure poses a bottleneck for achieving the previously mentioned latencies.

HARQ is a physical layer mechanism that employs feedback to transmit at higher target Block Error Rate s (BLERs), while achieving robustness of the transmission by providing retransmissions based on the feedback (ACK - acknowledgment/ NACK - non-acknowledgment). However, it imposes an additional delay on the transmission, designated as HARQ Round Trip Time (RTT). The HARQ RTT incorporates unavoidable physical delays, such as processing times (hardware delays), propagation delays, and Transmission Time Interval s (TTIs), i.e. the transmission duration. This lead to the abandonment of HARQ for the 1 ms end-to-end latency use case of URLLC at least for the initial URLLC specification in Rel. 15 [5]. This decision implied that the code rate had to be lowered such that a single shot transmission, i.e. no retransmissions and no feedback, achieves the required BLER. On the one hand, this simplifies the system design, however on the other hand it sacrifices the overall spectral efficiency of URLLC transmissions. Hence, reducing the RTT to a minimum for enabling HARQ in URLLC becomes a critical issue. Even use cases allowing regular HARQ retransmissions within the latency budget, profit from the reduced HARQ RTT enabling more HARQ iterations, i.e. using a higher code rate for each re-/transmission due to more HARQ iterations in total within the latency budget. This improves the overall spectral efficiency in common HARQ applications.

In this work, we focus on providing the HARQ feedback earlier by predicting the decoding outcome, designated as Early HARQ (E-HARQ) in the following. As will be discussed in detail below, the HARQ RTT is composed of several components including propagation delays and processing delays. Our approach focuses solely on reducing the delay on the receiver side from the start of reception until the feedback is sent. Hence, all other components contributing to the HARQ RTT will be considered fixed for this purpose. E-HARQ schemes [6], [7] provide the feedback on the decodability of the received signal ahead of the end of the actual transmission process. The crucial component in this setting is the classification algorithm that provides the feedback, which we aim to optimize using machine learning techniques.

Earlier approaches addressing the feedback prediction problem with the sole exception of [8] focused exclusively on one-dimensional input features as Bit Error Rate (BER)

estimates in combination with hard thresholding as classification algorithms [6], [7]. In [9], authors introduced the so-called Variable Node Reliability (VNR), as it will be discussed in Section III-A below, to exploit the substructures of Low-Density Parity-Check (LDPC) codes for prediction. However, only a single feature, i.e. a single decoder iteration, in combination with hard thresholding has been used. We expect improvements in prediction accuracy by extensions in several directions in combination with more complex classification algorithms: (a) the evolution of input features through several decoder iterations considered for the first time in [8], (b) higher-dimensional intra-message features that in the ideal case leverage knowledge about the underlying block code and (c) history features that leverage information about the channel state from past submissions that is available at the receiver.

Here we significantly expand the approach put forward in [8], where we discuss first E-HARQ results empowered by machine learning techniques. The training and testing is performed on simulated data obtained from stochastic channel models, which are widely used for performance evaluations of physical layer techniques [10]. We present an extended theoretical discussion in particular including the extension to multiple retransmissions and a system model that incorporates scheduling effects for the system evaluation thereby allowing a much more precise evaluation of the performance of E-HARQ-systems in realistic environments. On the classification side, this is supplemented by extended experiments including different input features and classification algorithms such as a newly developed supervised autoencoder for a larger range of SNR conditions, subcode lengths and different channel models.

The paper is organized as follows: In Section II we review the E-HARQ feedback process and investigate the role of the classification algorithm in a simple probabilistic model and in a more realistic setting of limited system resources. In Section III we discuss machine learning approaches for the classification problem introducing different input features and algorithms. The classification performance as well as the system performance is evaluated in Section IV for different signal-to-noise ratios, subcode lengths and channel conditions. We summarize and conclude in Section V.

II. Early HARQ Feedback

As discussed in the Introduction, E-HARQ approaches aim to reduce the HARQ RTT by providing the feedback on the decodability of the received signal at an earlier stage. This enables the original transmitter to react faster to the current channel situation and to provide additional redundancy at an earlier point. In regular HARQ, the feedback generation is strongly coupled to the decoding process. In particular, the receiver applies the decoder on the whole signal representing the total codeword. An embedded Cyclic Redundancy Check (CRC) enables to check the integrity of the decoded bit stream. The result of this check is transmitted back as HARQ feedback, either acknowledging correct reception (ACK) or asking for further redundancy (NACK). Providing early feedback (E-HARQ) implies decoupling the feedback
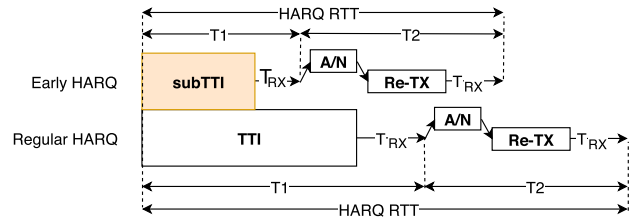

Fig. 1. Timeline of regular HARQ compared to early HARQ. (HARQ RTT: HARQ round trip time; TTI: transmission time interval; T_{RX}: processing time at the receiver; A/N: ACK/NACK feedback transmission; Re-TX: retransmission; T_1: time from initial reception to feedback transmission T_2: time from transmission of feedback to the end of the processing of the retransmission at the receiver).

generation from the decoding process, which introduces a misprediction probability since the actual outcome is not known in advance. Although misprediction errors are not avoidable, the design choices for the prediction affect the system performance a lot, i.e. asking for more retransmissions than actually required (over-provisioning) or less (under-provisioning). The impact of this design choice is evaluated more in detail in Section II-C. However, by taking this step, it is possible to use only a portion of the transmission and thus reducing the time from the start of the initial reception to transmitting the feedback (T_1). In total, the retransmission is scheduled earlier, hence also reducing the HARQ RTT, see Fig. 1. The time for transmitting the feedback and receiving the retransmission (T_2) is not affected by this. For LDPC codes, E-HARQ can be realized under exploitation of the underlying code structure by investigating the feedback prediction problem on the basis of so-called subcodes [9], [11] from the parity-check matrix. These subcodes are constructed by choosing a subset of rows from the original parity-check matrix and all the associated columns, so-called variable nodes [9]. The fraction of the subcode length to the full codelength with typical values ranging from 1/2 to 5/6, is designated as subTTI in Fig. 1. Shorter subcode lengths reduce the RTT but at the same time render the prediction problem more complicated.

In this section, we first introduce a simple probabilistic system model in Section II-A to provide an easy tool that evaluates the performance of the here presented E-HARQ schemes. However, this model only provides a measure in means of the final BLER and additionally implies the assumption of infinite resources by not penalizing unnecessary retransmissions. Hence, in Section II-B, we provide a more realistic system model together with the analysis of implications of finite size systems in Section II-C. This model provides a more suitable tool to evaluate the performance in practical systems, such as 5G and LTE. The finite-size system argument establishes an optimal point of operation for the E-HARQ schemes that is specific for the available system resources and does not exist in a system with unlimited resources.

A. Probabilistic Model for Single-Retransmission E-HARQ

We analyze single-retransmission E-HARQ in a simple probabilistic model. The straightforward extension to multiple retransmissions is discussed in Appendix A. The structure of
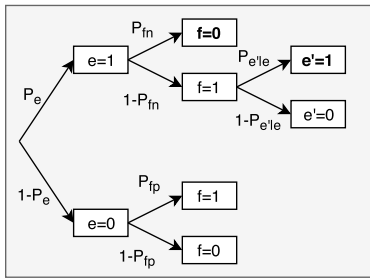
Fig. 2. Probabilistic model for single-retransmission E-HARQ (terminal nodes in bold face lead to an effective block error). We use binary random variables e/e' to reflect the state of the transmission ($e/e' = 0$: no block error, $e/e' = 1$: block error) and a binary random variable f to quantify the feedback sent ($f = 0$: *ACK*, $f = 1$: *NACK*).

the probabilistic model for E-HARQ is reflected in Figure 2. After the initial transmission we end up in a block error state ($e = 1$) with probability $P_e \equiv P(e = 1)$. Here we follow the common scheme in imbalanced classification problems encoding the minority i.e. block error class as positive, even though the opposite assignment is often used in the communications literature. In the case $e = 0$ the codeword gets decoded correctly irrespective of the feedback sent and a false positive feedback only implies an unnecessary transmission, which has no effect on the performance under the infinite resources assumption. We use a binary random variable f to reflect *ACK* ($f = 0$) or *NACK* ($f = 1$) feedback. In the former case we send either *ACK* with probability $P_{fn} \equiv P(f = 0|e = 1)$, which leads to an effective block error, or *NACK* with probability $P(f = 1|e = 1) = 1 - P_{fn}$. In the latter case the message gets retransmitted which leads to an effective block error with probability $P_{e'|e} = P(e' = 1|e = 1)$. The value for $P_{e'|e}$ crucially depends on the design of the feedback system most notably on the code rate used for the retransmission. However, one has to keep in mind that a decreased block error rate for the retransmission due to a decreased code rate might lead to latency losses due to the necessity of accommodating longer retransmissions. For identical retransmissions using an independent channel realization we would have $P_{e'|e} = P_e$ or even $P_{e'|e} < P_e$ if the decoder makes use of information from both transmissions for example using chase combining. For later reference we also define the joint probability $P_{e \wedge e'} \equiv P_e \cdot P_{e'|e} = P((e = 1) \wedge (e' = 1))$. This simple argument leads to an effective block error probability

$$p_{BLE,eff,1} = P_e \cdot \big(P_{fn} + (1 - P_{fn})P_{e'|e}\big)$$
$$= P_e P_{fn} + P_{e \wedge e'}(1 - P_{fn}). \quad (1)$$

The effect of an imperfect feedback channel could be easily incorporated in this formalism by defining effective false negative/positive rates but is omitted here for simplicity. Empirically we can replace P_e and $P_{e \wedge e'}$ by estimated block error rates and the conditional probability P_{fn} by the classifier's false negative rate (FNR) as obtained from the confusion matrix. Obviously the lowest possible effective BLER is achieved for perfect feedback, i.e. $P_{fn} = 0$, for which we have $p_{BLE,eff,1} = P_{e \wedge e'}$. (1) only depends on the baseline BLERs

P_e and $P_{e \wedge e'}$ and the classifier's false negative rate P_{fn} with leading order contribution given by $P_{fn} \cdot P_e$. In the limit where the $P_{fn} \ll P_{fb,e}$ the leading behavior is just $P_e \cdot P_{fb,e}$ and hence independent of the classification performance.

Considering the question of latency, the simplest metric is to consider the expected number of retransmissions $\langle \Delta T_1 \rangle$, which relates to the spectral efficiency of the approach. Therefore we evaluate the probability $P_{r,1}$ for a single retransmission. Again using Figure 2, we obtain

$$P_{r,1} \equiv P_r = P_e(1 - P_{fn}) + (1 - P_e)P_{fp}. \quad (2)$$

As above, the conditional probability $P_{fp} \equiv P(f = 1|e = 0)$ can be identified empirically with the classifier's false positive rate (FPR). The leading order contribution to (3) is given by $P_e + P_{fp}$ and the number of expected retransmissions therefore profits from a decreased FPR. For the case of a single retransmission, the expected number of retransmissions $\langle \Delta T_1 \rangle$ coincides with the single-retransmission probability,

$$\langle \Delta T_1 \rangle = P_{r,1}. \quad (3)$$

These results already hint at the crucial importance of adjusting the classifier's working point by balancing FNR versus FPR: A reduction of the FNR leads to a smaller effective block error probability, see (1), but comes along with an increased FPR, as the two kinds of classification errors counterbalance each other. This in turn leads to an increase in latency, see (3). From the present discussion it might seem a reasonable strategy to target an arbitrarily small FNR such that the effective block error probability approaches the theoretical limit. However, this argument only holds for a system with unlimited resources, as will be discussed below.

B. System Model

In order to derive a tool for evaluation of the performance of the discussed predictors, we introduce in this section a more sophisticated system model that leans on the structure of today's mobile network technologies. In cellular networks, such as LTE and 5G, Orthogonal Frequency Division Multiplexing Access (OFDMA) has been established due to its scheduling flexibility. Especially, opportunistic scheduling allows to use the best possible channel for a transmission. Here, we assume a simplified OFDMA system with equally sized N_{res} resources, i.e. frequency resources and a defined transmission duration in time, so-called TTI. The HARQ mechanism, regular HARQ as well as E-HARQ, requests based on the received parts of the transmission a retransmission, which is scheduled at earliest after T_{RTT} time slots.

The main advantage of E-HARQ over regular HARQ is the reduced HARQ RTT. Hence, depending on the latency budget more HARQ iterations, i.e. more re-/transmissions each incorporating feedback from the receiver within the latency constraint, might be used to improve the system performance. In the following, we assume the TTI length the default time unit. In practical systems, there exist several possibilities to design the TTI length such that it fits the requirements. The number of Orthogonal Frequency Division Multiplexing (OFDM) symbols belonging to a TTI or the subcarrier spacing

TABLE I

SYSTEM EVALUATION PARAMETERS

UE packet arrival rate - $P_{\mathrm{A,UE}}$	medium load - 0.3, high load - 0.36
Number of UEs - N_{UE}	20
Number of resources per time slot - N_{res}	10
Delay constraint - T_{c}	long symbols - 3, short symbols - 11
long TTI HARQ RTT - T_{RTT}	1 (E-HARQ 1/2), 2 (regular HARQ)
short TTI HARQ RTT - T_{RTT}	5 (E-HARQ 5/6), 6 (regular HARQ)
BLER of (re-)transmissions - P_{e}	as given in Table III

of the OFDM modulation can be changed, such that the physical time duration of a TTI scales accordingly. In this work, we evaluated two different system approaches, long and short TTI lengths.

The HARQ RTT is mainly comprised by the processing time, which scales with the TTI length in general [4], and the time required for transmitting the feedback, which does not depend on the TTI length. Thus, for long TTI lengths this time can be considered relatively small. However, for short TTI lengths this constant component has to be considered for E-HARQ as well as for regular HARQ systems. Hence, for long TTIs, we assumed $T_{\mathrm{RTT}} = 1$ for rate-1/2 E-HARQ, which means that the retransmission is received in the next TTI, and $T_{\mathrm{RTT}} = 2$ for regular HARQ, so that for regular HARQ one TTI has to be skipped. Analogously, for short TTIs, $T_{\mathrm{RTT}} = 5$ for rate-5/6 E-HARQ and $T_{\mathrm{RTT}} = 6$ for regular HARQ. Depending delay constraint T_{c} this results to a maximum number of retransmissions possible within the said latency budget. For long and short TTIs this allows depending on the system load up to two retransmissions in the E-HARQ-scheme compared to only one in the regular HARQ-scheme. Due to the scalability of the TTI length, the absolute value of T_{c} might be set to an arbitrary value, e.g. 1 ms. Thanks to the previously mentioned opportunistic scheduling possibilities of OFDMA, we assume that the retransmission is independent of the previous transmission, i.e. $P_{\mathrm{e'|e}} = P_{\mathrm{e}}$ and the total BLER $P_{\mathrm{e,total}} = (P_{\mathrm{e}})^{n+1}$, where n is the number of retransmissions. Furthermore, an i.i.d. packet arrival rate $P_{\mathrm{A,UE}}$ for each User Equipment (UE) is assumed. Thus, a single UE can only have one new transmission per time slot. The arrival rate for the high load scenario, as given in Table II-B, has been chosen such that the E-HARQ performance saturates in the URLLC relevant range. For the medium load scenario, the arrival rate has been chosen slightly lower such that no saturation behavior is observable in the relevant range. For simplicity the following argument is carried for a perfect feedback channel, i.e. for $P_{\mathrm{fb,e}} = 0$, which is a reasonable assumption considering the results of the previous implying that the feedback error probability is at most of subleading importance. The system parameters are summarized in Table II-B.

C. Implications of Finite System Size

In practical systems, there is a trade-off between the False-Negative Rate (FNR) and False-Positive Rate (FPR) due to the limited amount of available resources. Whereas a lower FNR increases the effective BLER, as shown in the Section II-A, it increases the transmission overhead on the other hand. Depending on the available resources this leads to resource shortage, also causing additional delays since transmissions cannot be scheduled in the designated time slots. This brings us to the term of packet failure rate which is described by the probability that a packet is delivered successfully within a given delay constraint, i.e. latency budget T_{c}. Interestingly, there is an optimal operation point which captures the trade-off such that the packet failure rate is minimized.

For the assumptions on the system model described in the previous section, the packet failure probability is given as

$$P_{\mathrm{pf}} = (1 - P_{\mathrm{S,0}}) + P_{\mathrm{S,0}} P_{\mathrm{e}} P_{\mathrm{H,e,1}}, \qquad (4)$$

where $P_{\mathrm{S},j} \equiv P(T_j \leq T_c)$ denotes the probability of scheduling j transmissions within the time constraint T_c and $P_{\mathrm{H,e},j}$ denotes the failure probability after the jth transmission. For a single retransmission ($n = 1$) the latter is given by $P_{\mathrm{H,e,1}} = P_{\mathrm{fn}} + (1 - P_{\mathrm{fn}})[(1 - \frac{P_{\mathrm{S,1}}}{P_{\mathrm{S,0}}}) + \frac{P_{\mathrm{S,1}}}{P_{\mathrm{S,0}}} P_e]$. Inserting this expression into (4) leads to the familiar form (1) up to scheduling probabilities $P_{\mathrm{S},j}$. Generalizing to multiple retransmissions, the error probability for the jth resubmission can be defined recursively via

$$P_{\mathrm{H,e},j} = P_{\mathrm{fn}} + (1 - P_{\mathrm{fn}}) \left[\left(1 - \frac{P_{\mathrm{S},j}}{P_{\mathrm{S},j-1}} \right) + \frac{P_{\mathrm{S},j}}{P_{\mathrm{S},j-1}} P_{\mathrm{e}} P_{\mathrm{H,e},j+1} \right], \qquad (5)$$

where we set $P_{\mathrm{H,e},j} = 1$ if j exceeds the maximum number of retransmissions, i.e. if $j > n$. As shown above for a single retransmission, (4) reduces to (1) if one sets all scheduling probabilities to one. The same applies for multiple retransmissions for which one obtains (10). Therefore, (4) can be seen as a generalized version of the effective BLER.

However, the effective BLER does not consider the finite resources and thus cannot capture the actual performance of the evaluated HARQ schemes in a practical implementation. We will refer to this case as the infinite resource baseline compared to the finite resource baselines discussed below.

At first glance, (4) suggests minimizing the FNR P_{fn}. However, a closer examination reveals that the scheduling probabilities $P_{\mathrm{S},j}$ carry a dependence on both FNR and FPR via the underlying resource distribution function, see Appendix C. FNR and FPR counteract each other in the sense that a decreased FNR will lead to an increase in the FPR. Considering the dependency on the resource distribution function, an increase of the FPR P_{fp} increases the load on the system, thus lowers the probability that a transmission and its retransmission is scheduled within the time constraint. This fact is already apparent from the expected number of retransmission as obtained in (3) which scales with the FPR at leading order. This suggests that the packet failure probability seen as a function of the FNR will show a minimum characterizing an optimal trade-off between FNR and FPR for the given system resources.

The derived packet failure probability P_{pf} within a fixed latency budget rather than spectral efficiency represents the most relevant performance metric for practical evaluations.

Additionally, apart from comparing the different E-HARQ schemes among each other, it enables a performance comparison with regular HARQ, which is crucial if E-HARQ is considered for URLLC. Here, aside the system setup presented in the previous section, for regular HARQ the FNR and FPR is assumed to be zero as false predictions can be neglected due to the CRC included in the transmission.

III. MACHINE LEARNING FOR EARLY HARQ

The machine learning task of predicting the decodability of a message based on information from at most the first few decoder iterations is an inherently imbalanced classification problem. This imbalance is a direct consequence of the base BLERs of the order 10^{-3} that are required in order to be able to reach effective BLERs of the order 10^{-5}, see (1). Different ways of dealing with this imbalance have been explored, see [12] for a review. These can be categorized as cost-sensitive learning, rebalancing techniques and threshold moving. The discussion in this section focuses on the latter in the sense of readjusting the decision boundary of any trained model that outputs probabilities for the predicted classes, see also [13] and references therein.

By moving the decision boundary one is able to investigate the discriminative power of a given classifier over a whole range of different working points. This is typically analyzed in terms of Receiver-Operation curves (ROC) or Precision-Recall (PR) curves. In order to summarize the classifier's performance with a single number, one conventionally resorts to reporting area-under-curve (AUC) metrics. Here we focus on the PR curve and the corresponding area under the PR curve, AUC-PR, rather than the ROC-curve as the former has been shown to better reflect the classifier's performance for highly skewed datasets [14], [15]. However, when summarizing the discriminative power of a classifier using a single figure, one loses fine-grained information about classification performance at different working points. This is particularly true since the full AUC naturally covers the whole range values for the decision boundary, many of which are irrelevant for practical applications where the classification performance in the small FNR-regime is most relevant. In addition, the actual implementation of the classifier requires a definite choice for the decision threshold. Therefore we supplement the global AUC-PR information with an analysis based on FNR-PPR curves. It is worth noting that the FNR-FPR curves directly relate to ROC curves since the true positive rate TPR that is plotted on the ordinate of the ROC-curve relates to the FNR via TPR = 1 - FNR. FNR and FPR represents the natural choice in our case since they represent the key output figures from the system point of view, see Sec II-A.

A. Input Features

We distinguish single-transmission-features derived from a single transmission and history information from past transmissions. In principle all these features can be combined at will to form the set of input features for the classification algorithm.

The raw data for a single transmission provided by the simulation is given by (a posteriori) LLR values after different decoder iterations. E-HARQ approaches to reduce the HARQ RTT have been first discussed in [6] and [7]. This approach estimates the BER based on the Log-Likelihood Ratio s (LLRs) and utilizes a hard threshold to predict the decodability of the received signal. The LLR gives information on the likelihood of a bit being either 1 or 0. Denoting $\boldsymbol{y}$ as the observed sequence at the receiver, the LLR of the kth bit b_k is defined as

$$L(b_k) = \log \frac{P(b_k = 1|\boldsymbol{y})}{P(b_k = 0|\boldsymbol{y})}. \quad (6)$$

Having the LLRs of a subcode or the whole codeword allows to calculate an estimated BER for the received signal vector via

$$B\hat{E}R = \frac{1}{M} \sum_k \frac{1}{1 + |L(b_k)|}, \quad (7)$$

where M is the length of the LLR vector. Based on this metric the decoding outcome is predicted, where a higher $B\hat{E}R$ means a lower probability of successful decoding.

A further improved approach has been presented in [9] and [11]. The authors propose to exploit the code structure to improve the prediction performance. In case of LDPC codes, this is realized by constructing so-called subcodes from the parity-check matrix. Using a belief-propagation based decoder on the LLRs of the subcodeword results in a posteriori LLRs:

$$\Lambda_k^{(j)} = \Lambda_k^{(j-1)} + \sum_{m \in \mathcal{M}(k)} \beta_{m,k}^{(j)}, \quad (8)$$

where $\Lambda_k^{(0)} \equiv L(b_k)$, $\mathcal{M}(k)$ is the set of check nodes which are associated to the variable node of k and $\beta_{m,k}^{(j)}$ is the check-to-variable node message from check node m to variable k. Here we use the superscript j in $\Lambda_k^{(j)}$ to denote the decoder iteration after which the posteriori LLRs were extracted with the obvious identification $\Lambda_k^{(0)} \equiv L(b_k)$. Again, the a posteriori LLRs are mapped to the same metric for each belief-propagation iteration, designated as VNR,

$$VNR_j = \frac{1}{M} \sum_i \frac{1}{1 + |\Lambda_i^{(j)}|}, \quad (9)$$

where M is the length of the subcodeword and j denotes the belief-propagation iteration. Hence, VNR_0 corresponds to $B\hat{E}R$. In [9], the authors used a hard threshold applied VNR_5 to predict decodability.

Assuming the receiver is operating on the same channel across different transmissions, it might be possible to increase the prediction performance by incorporating information from previous transmissions. This includes all features used as single-transmission features and in addition features that are only available after the end of the decoding process. As two representative examples for history features we investigate VNRs from past submissions (VNR_HIST) and information about the Euclidean distance between the correct codeword and the received signal vector (EUCD_HIST). Here one has to keep in mind that the latter information is only available if the correct codeword is known to the receiver as for example from

a previous pilot transmission but strictly speaking it cannot be reliably obtained from an ordinary previous transmission as even a correct CRC does not imply a correctly decoded transmission. For a given set of history features we consider means of the history features under consideration extracted from different numbers of past transmissions using sliding windows of length 1,2,5 and 9 in order to allow the classifier to extract information from past channel realizations at different time scales.

B. Classification Algorithms

As discussed in the introduction, we can view the problem either as a heavily imbalanced classification problem or as an anomaly detection problem. Here we briefly discuss suitable algorithms for both of approaches. As examples for binary classification algorithms we consider hard threshold (HT) classifiers, logistic regression (LR) (with L_2 regularization and balanced class weights) and Random Forests (RF). HT applied to VNR_0/VNR_5-data (referred to as HT0 and HT5 in the following) yield the classifiers used in the literature so far [6], [9]. For anomaly detection [16] one distinguishes unsupervised, semi-supervised and supervised approaches depending on whether only unlabeled examples, only majority-class examples or labeled examples from both classes are available for training. As anomaly detection algorithms we consider Isolation Forests (IF) [17] as classical tree-based semi-supervised anomaly detection algorithm and supervised autoencoder (SAE) as a novel neural-network based approach for supervised anomaly detection, see Appendix B for details. All classifiers apart from HT0 and HT5 operate on the first six VNRs $VNR_0, \ldots, VNR_5$ as input features. We leverage the implementations from scikit-learn [18] apart from SAC that was implemented in PyTorch [19].

IV. RESULTS

A. Simulation Setup

We compare classification performance of different classifiers based on AUC-PR and FNR-FPR curves. As external parameters we vary the SNR between 3.0 and 4.0 dB which results to BLERs in the considered URLLC regime taking the HARQ retransmissions into account, and subcode lengths between 1/2 and 5/6 to evaluate a rather aggressive prediction versus a more conservative one. The simulation setup used to produce training and test data follows the one reported in [9] and summarized in Table II. We use the raw simulation output as well as a number of derived features. Here we consider both single-transmission features as well as history-features that incorporate information from a number of past transmissions, see Appendix III-A for a detailed discussion. We then investigate the performance of a number of classification algorithms operating on these input features, see Appendix III-B for a detailed breakdown. In all cases we use 1M transmissions with independent channel realizations for training and evaluate on a test set comprising at least 1M transmissions. The size of the test set for each SNR/subcode combination is given in the second column of Table III. Hyperparameter tuning is performed once for the pedestrian channel (at SNR 4.0 dB

TABLE II
LINK-LEVEL SIMULATION ASSUMPTIONS FOR TRAINING AND TEST SET GENERATION

Transport block size	360 bits
Channel Code	Rate-1/5 LDPC BG2 with Z = 36, see [20]
Modulation order and algorithm	QPSK, Approximated LLR
Waveform	3GPP OFDM, 1.4 MHz, normal cyclic-prefix
Channel type	1 Tx 1 Rx, TDL-C 100 n , 2.9 GHz, 3.0 km/h (pedestrian) or 100.0 km/h (vehicular)
Equalizer	Frequency domain MMSE
Decoder type	Min-Sum
Decoding iterations	50
VNR iterations	5

and subcode length 5/6) on an additional validation set also comprising 1M samples. We standard-scale all different sets of input features independently using training set statistics. In this way we obtain a reasonable input normalization that is required for certain classification algorithms while keeping relative difference within different input feature groups intact.

B. Classification Performance

We start by discussing the classification performance for different classification algorithms based on VNR-features extending the analysis from [8]. The classification results are compiled in Table III. We compare AUC-PR that characterizes the overall discriminative power of the algorithm and which tends to 1 for a perfectly discriminative classifier. The largest improvements to the simplest thresholding method HT0 is seen for longer subcode lengths such as 5/6. In these cases more complex classification methods applied to the full VNR-range show only small improvements over the HT5 threshold baseline. A different picture emerges at smaller subcode lengths. Here using VNRs from higher decoder iterations (HT5) does not improve or even worsen the classification performance compared to HT0. Here more complex classification algorithms show their true strengths and show larger improvements compared to HT0/HT5. This is a plausible result since decreasing the subcode length renders the classification problem more complicated and more complex classifiers can profit more from this complication. If we assess the difficulty of the classification problem based on the scores achieved by the classifiers, a clear picture emerges: As discussed before decreasing the subcode length for fixed SNR renders the classification problem more difficult, whereas decreasing the SNR for fixed subcode length has the opposite effect most notably because of an increasing BLER. On the other hand the BLER sets the baseline for the HARQ performance, see (1), which overcompensates the positive effects of the improved classification performance. The overall best discriminative power across different SNR-values, subcode lengths and channel conditions shows the supervised autoencoder closely followed by regularized logistic regression. The fact that the AUC-PR results for LR, RF and SAE are so close just reflects a similar overall discriminative power of these algorithms despite of fundamentally different underlying principles.

TABLE III
COMPARING CLASSIFICATION PERFORMANCE BASED ON AUC-PR (CLASSIFIERS AS SPECIFIED IN SECTION III-B)

| SNR SC ch	#train/#test	BLER	HT0	HT5	LR	RF	IF	SAE
4.0dB 5/6 ped	1M/3M	0.001604	0.811	0.902	0.905	0.907	0.890	**0.908**
4.0dB 1/2 ped	1M/4M	0.001626	0.801	0.799	**0.834**	0.832	0.827	**0.834**
3.5dB 5/6 ped	1M/1M	0.002841	0.844	0.920	0.921	0.924	0.912	**0.926**
3.5dB 1/2 ped	1M/4M	0.002777	0.821	0.814	**0.847**	0.846	0.839	**0.847**
3.0dB 5/6 ped	1M/1.5M	0.004742	0.863	0.927	**0.934**	**0.934**	0.923	**0.934**
3.0dB 1/2 ped	1M/1.5M	0.004742	0.851	0.840	0.872	0.871	0.865	**0.874**
3.5dB 1/2 veh	1M/3M	0.002866	0.824	0.818	**0.851**	0.850	0.846	**0.851**

(a) 4.0 dB subcode 5/6

(b) 4.0 dB subcode 1/2
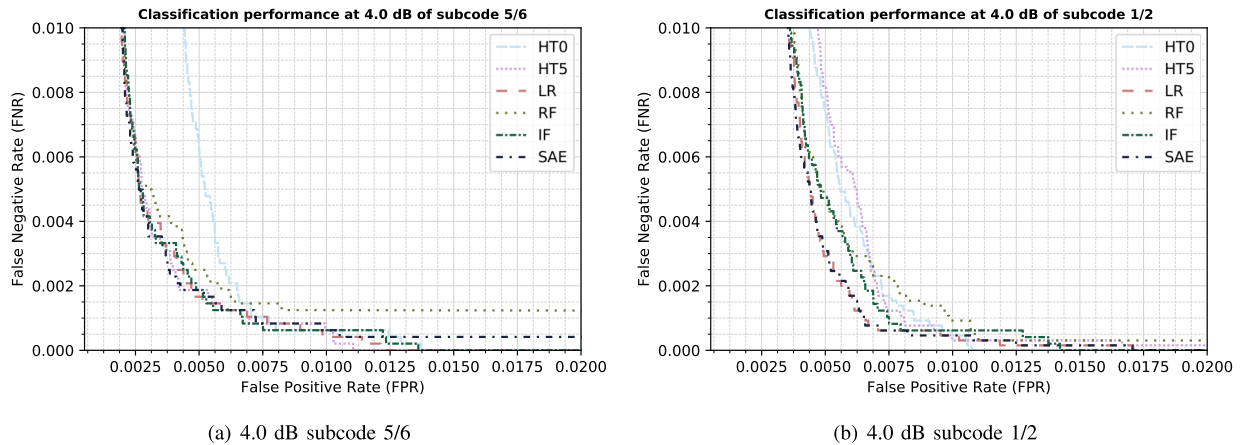
Fig. 3. Selected examples for classification performance based on VNR-features in the pedestrian channel.

This does, however, not imply coinciding FNR-FPR curves, where the classifiers show rather different behavior in certain FNR regions, see Figure 3 for selected results. Random Forests, for example, show in general a very good overall performance but are considerably weaker than other classifiers in the small FNR-regime. When looking at FNR-FPR curves as the ones presented in Figure 3, one has to keep in mind that it is very difficult in the extremely imbalanced regime to obtain reliable estimates of the FNR as both the numerator (false negatives) and the denominator (sum of false negatives and true positives) are small numbers requiring large sample sizes for a stable evaluation. This applies in particular to the region of small FNRs below 0.001.

To summarize, we clearly demonstrated that incorporating the evolution of the VNR across the first five decoder iterations into more complex classification algorithms such as logistic regression or supervised autoencoders leads to gains in the overall classification performance in particular in comparison to hard threshold baselines. This conclusion holds for various SNR-values, subcode lengths and channel conditions. Implications of these findings for the system performance will be discussed in Section IV-C.

We restrict the investigation of history features to the SAE classifier as the best-performing classifier from the previous section. However, we checked that the qualitative conclusions about the importance of history features hold irrespective of the classification algorithm under consideration. In Table IV we discuss the impact of history features on the classification performance in addition to the VNR-features discussed above.

TABLE IV
COMPARING CLASSIFICATION PERFORMANCE BASED ON AUC-PR
UPON INCLUDING HISTORY FEATURES (FOR SAE)

features	4.0dB 1/2 ped	3.5dB 1/2 ped	3.5dB 1/2 veh
VNR	0.834	0.847	0.851
VNR+VNR_HIST	0.860	0.872	0.852
VNR+EUCD_HIST	**0.883**	**0.892**	**0.861**

Irrespective of SNR, subcode length and underlying pedestrian or vehicular channel model, we see an improvement in classification performance upon including history features with best results achieved by incorporating Euclidean distance features. History information seems to lead to larger improvements in the pedestrian channel compared to the vehicular channel. This is in line with the channel conditions remaining unchanged for a longer time in the pedestrian compared to the vehicular case.

There are different caveats to this result. First of all, as discussed in Section III-A, the Euclidean distance is only known to the receiver if the underlying codeword is known as it would be the case for a previous pilot transmission, which would however lead to latency overheads. Therefore the result including Euclidean history features most likely overestimates the improvements in classification performance that can be obtained from using history features. Secondly, the use of history features is at tension with the assumption of an independent channel realization for the retransmission in the sense of $P_{e'|e} = P_e$ as used in our system model. It is very unlikely that the improvements in classification
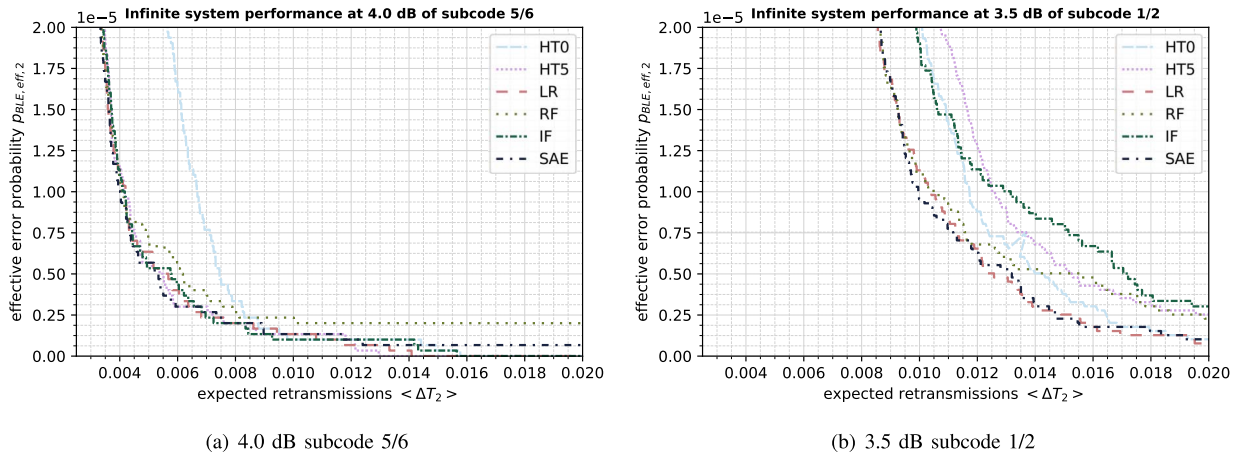
(a) 4.0 dB subcode 5/6

(b) 3.5 dB subcode 1/2

Fig. 4. Selected examples for system performance in the pedestrian channel for two-retransmission E-HARQ with unlimited system resources.

performance can compensate the loss of approximately one order of magnitude in the error rate for the retransmission of $P_{e'|e} \approx 10^{-2}$ using the same channel compared to the baseline BLER of the order of 10^{-3} for an independent retransmission. Therefore the system level analysis is carried out using VNR-features only. Nevertheless the results put forward here stress the prospects of further investigations of features that explicitly characterize the channel state such as explicit channel state information that could have been obtained by a pilot transmission preceding the transmission.

C. System Performance

We start by discussing system performance based on the simple probabilistic model for E-HARQ with unlimited system resources as introduced in Section II-A. The results are obtained straightforwardly from the FNR-FPR-curves presented in Section IV-B using (1) and (3) or the corresponding generalizations for multiple retransmissions (10) and (17). Here we adopt $P_{e'|e} = P_e$ as in Section II-B. Here we present results for two retransmissions that are possible for E-HARQ in both TTI scenarios discussed in Section II-B. In fact, increasing the number of retransmissions beyond two does not lead to further noticeable improvements in the given FNR range. In all cases effective BLERs of the order 10^{-5} are attainable. Decreasing the subcode length from 5/6 to 1/2 while keeping the same effective BLER of $1 \cdot 10^{-5}$ as a definite example requires an increase of 40% and 45% in retransmissions at SNR 4 dB and 3 dB respectively. Correspondingly, decreasing the SNR for fixed subcode length from 4 dB to 3 dB while again keeping the effective BLER fixed leads to an overhead of 70% and 77% in retransmissions for subcode 5/6 and 1/2 respectively. However, as discussed in Section II-C, the presented effective BLERs only represent theoretical lower bounds for actual packet failure rates that are achievable in actual systems as they do not incorporate scheduling effects. In this infinite system setting there is no distinguished working point for the classifier and the only way of discriminating between different classifiers in the system setting is to rank by the number of expected transmissions for fixed effective error probability.

Figure 5 shows exemplary results of the packet failure rate over the FNR of the E-HARQ schemes under medium ($P_{A,UE} = 0.3$) and high system load ($P_{A,UE} = 0.36$) together with the regular HARQ-baseline and the infinite system results from (10). The upper Figures 5(a) and 5(b) show the long TTI design, as described in Section II-B, at 3.5 dB. For the high load (Figure 5(a)) as well as the medium load (Figure 5(b)) scenarios, the E-HARQ schemes achieve a superior performance compared to the regular HARQ thanks to the additional retransmission which is possible within the same latency constraint. However, a packet failure rate less than 10^{-5} is only achieved in the medium load scenario. Here, we note that the actual performance of the E-HARQ schemes is approximated well by the approach with infinite resources, at least for high packet failure rates above 10^{-5}. Only in the lower region an attenuation of the decrease is visible, whereas all prediction schemes achieve a comparable performance. In the high load scenario in Figure 5(a), we see the trade-off behavior, discussed in Section II-C. The packet failure rate decreases only up to a certain minimum at the optimal FNR-FPR trade-off and starts increasing after passing that point. So, lowering the FNR further after passing that point increases the packet failure due to the resource shortage. In this region, the actual performance of the prediction schemes becomes critical. Hence, SAE and LR have the lowest optimum. HT0 and HT5 perform worse at their optimal operation points, whereas HT0 is still performing better than HT5.

The resource shortage effect is clearly visible in Figure 6, where the same load is applied in both scenarios but the latency constraint is relaxed in Figure Figure 6(b). As obvious in Figure 6(a), the packet failure rate for all schemes is far away from the targeted packet failure rate of 10^{-5}. With a relaxed latency constraint, as shown in Figure Figure 6(b), the performance is closer to the target packet failure rate. This improvement is explainable by two effects. First, the E-HARQ schemes benefit from the additional retransmission, which is possible in the relaxed latency constraint and thus in total achieve still a better performance than the regular HARQ. However, the gap is smaller compared to the normal latency constraint. Especially in the high load scenario,

(a) 3.5 dB subcode 1/2 (high load, $P_{\mathrm{A,UE}} = 0.36$)

(b) 3.5 dB subcode 1/2 (medium load, $P_{\mathrm{A,UE}} = 0.30$)

(c) 4.0 dB subcode 5/6 (high load, $P_{\mathrm{A,UE}} = 0.36$)

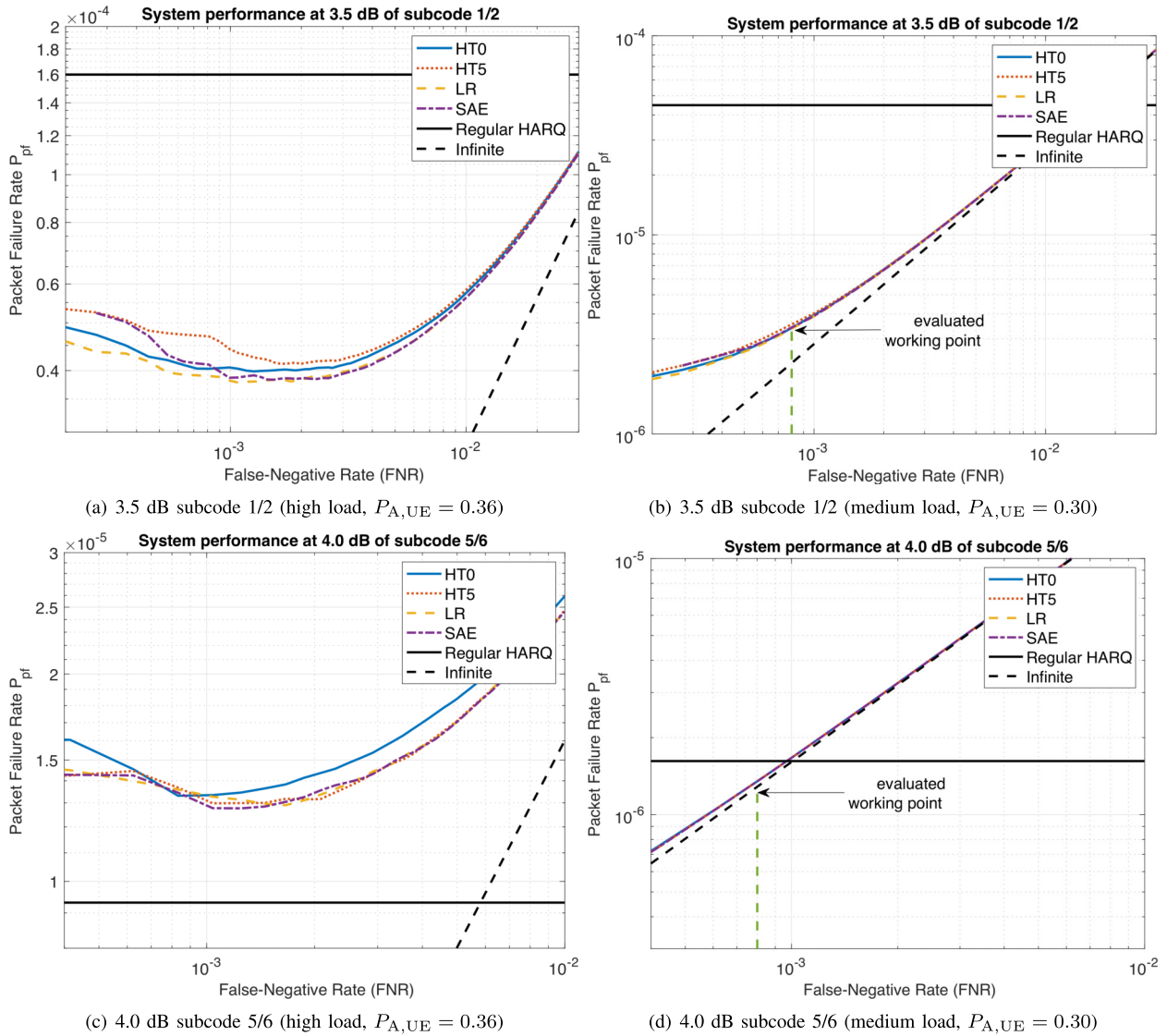(d) 4.0 dB subcode 5/6 (medium load, $P_{\mathrm{A,UE}} = 0.30$)

Fig. 5. Exemplary system performance comparison for rate 1/2 and 5/6 prediction schemes in high load and medium load scenarios (green dashed line indicates $\mathrm{FNR_{eval}}$).

the regular HARQ profits from the increased scheduling flexibility although it can only perform the same number of HARQ retransmissions. The resource shortage effect is also observable for the regular HARQ performance comparing the medium load and the high load scenarios. It is notable that the regular HARQ could at least achieve a packet failure rate less than 10^{-4} in the medium load scenario, whereas it is performing even worse in the high load scenario. We can see that even more clearly in the short TTI design in Figures 5(c) and 5(d). In the medium load scenario in Figure 5(d), the regular HARQ achieves a packet failure rate of almost 10^{-6}, which corresponds approximately to the ideal performance of HARQ. In this system setup the regular HARQ makes use of the whole scheduling flexibility and thus, at least for the medium load scenario, the influence of scheduling probabilities can be neglected for the regular HARQ. Despite the limited scheduling flexibilities of the E-HARQ schemes, they achieve a better performance than the regular HARQ.

However, this changes in the high load scenario in Figure 5(c). Here, we observe that the regular HARQ benefits from its scheduling gain and thus, achieves the lower packet failure rate. In the high load scenario, we see that all prediction schemes achieve a similar performance, except the HT0 which is remarkably less performing than the others.

As already visible in the previous results, there is no clear winning scheme for all the scenarios. However, to compare the overall performance of the schemes, we introduce the total score $t_s = \sum_t \log_{10} \frac{P_{\mathrm{pfr},s,t}}{\min_s P_{\mathrm{pfr},s,t}}$, where t is the enumerator over all SNRs and prediction rates and s is the enumerator over all HARQ schemes. In Table V we present the results for all scenarios, where the "$<$" sign indicates that an FNR larger than the optimal FNR has been used for evaluations. As already notable in Figure 5, the available data does not allow arbitrary small FNRs and thus the optimal operation point cannot be reached for the medium load case. Hence, we used $\mathrm{FNR_{eval}} = 8 \cdot 10^{-4}$ for the medium load evaluations

(a) 3.0 dB subcode 1/2 (high load, $P_{\mathrm{A,UE}} = 0.36$ and $T_c = 3$)   (b) 3.0 dB subcode 1/2 (high load, $P_{\mathrm{A,UE}} = 0.36$ and $T_c = 4$)
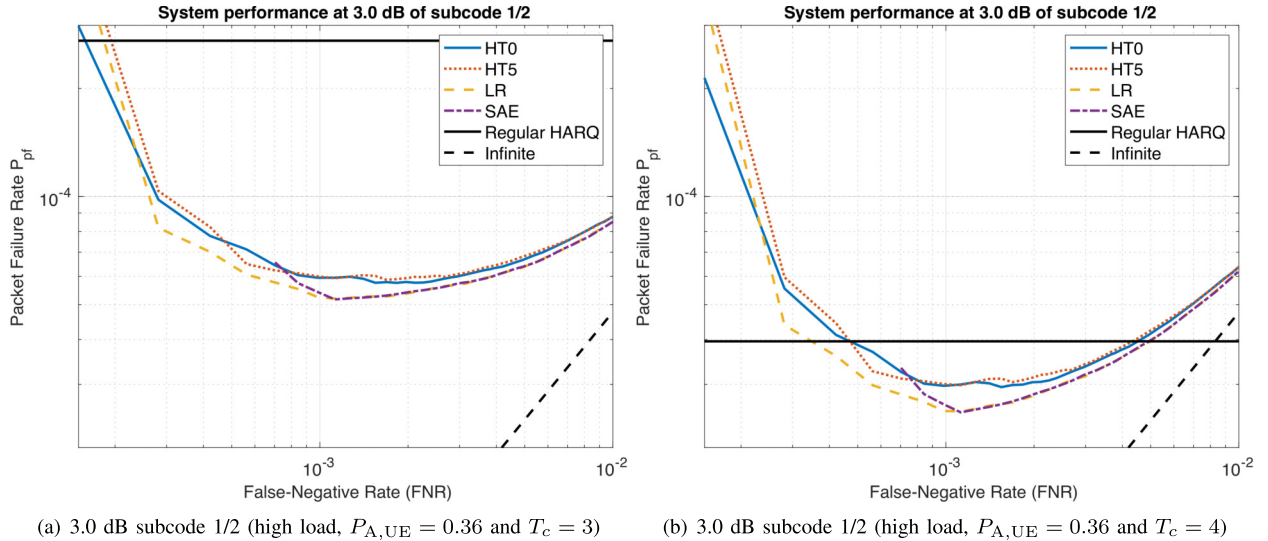
Fig. 6.   Effects of the scheduling gain in the high load regime for the strict and relaxed latency constraint.

TABLE V

COMPARING SYSTEM PERFORMANCE (P_{PF}) AT THEIR OPTIMAL FNR-FPR TRADE-OFF, AS DESCRIBED IN SECTION II-C

	scenario	regular HARQ	HT0	HT5	LR	SAE
medium load	3.0dB 1/2 ped	$8.59 \cdot 10^{-5}$	$< 6.26 \cdot 10^{-6}$	$< 6.28 \cdot 10^{-6}$	$< \mathbf{6.13 \cdot 10^{-6}}$	$< 6.18 \cdot 10^{-6}$
	3.5dB 1/2 ped	$4.48 \cdot 10^{-5}$	$< 3.43 \cdot 10^{-6}$	$< 3.80 \cdot 10^{-6}$	$< \mathbf{3.40 \cdot 10^{-6}}$	$< 3.67 \cdot 10^{-6}$
	4.0dB 1/2 ped	$2.36 \cdot 10^{-5}$	$< 1.93 \cdot 10^{-6}$	$< 1.92 \cdot 10^{-6}$	$< \mathbf{1.88 \cdot 10^{-6}}$	$< 1.92 \cdot 10^{-6}$
	3.0dB 5/6 ped	$8.59 \cdot 10^{-5}$	$< 6.36 \cdot 10^{-6}$	$< 6.15 \cdot 10^{-6}$	$< 6.08 \cdot 10^{-6}$	$< \mathbf{5.99 \cdot 10^{-6}}$
	3.5dB 5/6 ped	$7.86 \cdot 10^{-6}$	$< 2.25 \cdot 10^{-6}$	$< \mathbf{2.22 \cdot 10^{-6}}$	$< \mathbf{2.22 \cdot 10^{-6}}$	$< 2.23 \cdot 10^{-6}$
	4.0dB 5/6 ped	$1.62 \cdot 10^{-6}$	$< \mathbf{1.40 \cdot 10^{-6}}$	$< \mathbf{1.40 \cdot 10^{-6}}$	$< \mathbf{1.40 \cdot 10^{-6}}$	$< \mathbf{1.40 \cdot 10^{-6}}$
	3.5dB 1/2 veh	$4.48 \cdot 10^{-5}$	$< 3.47 \cdot 10^{-6}$	$< 3.48 \cdot 10^{-6}$	$< \mathbf{3.37 \cdot 10^{-6}}$	$< 3.68 \cdot 10^{-6}$
	total score t_s	6.2577	0.0685	0.0936	**0.0075**	0.0866
high load	3.0dB 1/2 ped	$2.72 \cdot 10^{-4}$	$5.75 \cdot 10^{-5}$	$5.87 \cdot 10^{-5}$	$5.20 \cdot 10^{-5}$	$\mathbf{5.17 \cdot 10^{-5}}$
	3.5dB 1/2 ped	$1.60 \cdot 10^{-4}$	$3.99 \cdot 10^{-5}$	$4.13 \cdot 10^{-5}$	$\mathbf{3.78 \cdot 10^{-5}}$	$3.83 \cdot 10^{-5}$
	4.0dB 1/2 ped	$9.56 \cdot 10^{-5}$	$2.94 \cdot 10^{-5}$	$2.88 \cdot 10^{-5}$	$\mathbf{2.76 \cdot 10^{-5}}$	$2.81 \cdot 10^{-5}$
	3.0dB 5/6 ped	$2.72 \cdot 10^{-4}$	$5.59 \cdot 10^{-5}$	$4.99 \cdot 10^{-5}$	$4.89 \cdot 10^{-5}$	$\mathbf{4.70 \cdot 10^{-5}}$
	3.5dB 5/6 ped	$1.61 \cdot 10^{-5}$	$2.05 \cdot 10^{-5}$	$\mathbf{1.61 \cdot 10^{-5}}$	$1.65 \cdot 10^{-5}$	$1.68 \cdot 10^{-5}$
	4.0dB 5/6 ped	$\mathbf{9.32 \cdot 10^{-6}}$	$1.33 \cdot 10^{-5}$	$1.30 \cdot 10^{-5}$	$1.29 \cdot 10^{-5}$	$1.28 \cdot 10^{-5}$
	3.5dB 1/2 veh	$1.60 \cdot 10^{-4}$	$3.88 \cdot 10^{-5}$	$4.06 \cdot 10^{-5}$	$\mathbf{3.64 \cdot 10^{-5}}$	$\mathbf{3.64 \cdot 10^{-5}}$
	total score t_s	3.2918	0.4599	0.3306	0.1713	**0.1703**

since it provides a sufficiently reliable estimation. The evaluation at fixed FNR underestimates the overall performance compared to regular HARQ but allow a reliable ranking between different classifiers. Obviously, for reaching the optimal point of operation more data is required in the medium load case.

Nevertheless, in the medium load regime, LR achieves by far the best overall performance. The other E-HARQ schemes achieve a similar performance, where HT0 is able to achieve a slightly better performance than the other two. Interestingly here, SAE has a worse performance compared to LR although it was the best performing classifier in the previous section. A closer inspection reveals that for very low FNR SAE cannot keep up with the other classifiers. Especially that region, being not relevant for the performance metrics of the previous section, explains the contradicting results. However, the expected performance for SAE is observed going to the high load regime. Here, SAE and LR are the best performing E-HARQ schemes far ahead HT0, HT5 and regular HARQ. As already noted in Figure 5, in the high load regime the

performance at higher FNRs is key. Hence, SAE is again in a well-operating region. In this region, we also note that HT0 is performing the worst among the classifiers though having the second-best performance in the medium load regime.

Summa summarum, E-HARQ is able to achieve large gains in means of packet failure rate compared to regular HARQ under latency constraints. The rather small performance difference in medium and high load scenarios stems from the resolution issue at very small FNRs in the medium load scenarios. Especially, LR is a promising approach, which achieves a good overall performance in high load as well as medium load regimes. The SAE as best-performing algorithm in the high-load case and the more extendable approach compared to LR might provide a viable alternative if the performance at very low FNRs is improved.

V. SUMMARY AND CONCLUSIONS

In this work we investigated machine learning techniques for E-HARQ by means of more elaborate classification

methods to predict the decoding result ahead of the final decoder iteration. We demonstrated that more complex estimators such as logistic regression or supervised autoencoder that exploit the evolution of the subcodeword during the first few decoder iterations lead to quantitative improvements in the prediction performance over baseline results across different SNR and channel conditions. We put forward a simple probabilistic model and a more elaborate system model incorporating scheduling effects to evaluate system performance in a realistic environment. In this way we were able to demonstrate the practical feasibility of reaching effective packet error rates of the order 10^{-5} as required for URLLC across a range of different SNRs, subcode lengths and system loads. More importantly, we showed that enabling more HARQ iterations by introducing E-HARQ improves the overall reliability over regular HARQ under strict maximum latency constraints. We hope to extend this analysis to other URLLC-relevant scenarios such as MIMO in the future.

Further improvements of the classification performance are conceivable extending the approach presented in this work. Our results suggest that history features incorporating channel information from previous transmissions positively influence the classification performance but remain to be investigated in more detail. Similarly it seems very likely that classification algorithms could profit from intra-message features that go beyond the simple averaging features such as VNRs considered in this work, which ideally directly incorporate the code structure of the underlying channel code. However, such features suffer from high dimensionality and large correlations. Here a challenge remains to identify the most discriminative set of input features and appropriate classification algorithms to further improve the classification performance.

Ultimately, more advanced classification algorithms, which are within reach using techniques presented in this work, might allow more fine-grained feedback instead of a binary NACK/ACK response. Incorporating this information on the level of the feedback protocol would allow to design custom feedback schemes with potentially large latency gains.

APPENDIX A
PROBABILISTIC MODEL FOR MULTIPLE-RETRANSMISSION E-HARQ

In this section, we present the generalization of the results from Section II-A. These are obtained straightforwardly using the same formalism as above. The generalization of the effective error probability from (1) to the case of n retransmissions is given by the iterative relation

$$p_{\text{BLE,eff,n}} = P_{\text{e}} P_{\text{H,e,1}} , \qquad (10)$$

where we defined for $j \leq n$

$$P_{\text{H,e,j}} = P_{\text{fn}} + (1 - P_{\text{fn}}) \cdot P_{e^{(j)}|e^{(j-1)} \wedge \ldots \wedge e^{(0)}} \cdot P_{\text{H,e,j+1}}, \qquad (11)$$

and otherwise $P_{\text{H,e,j}} = 1$, which reduces to (1) for $n = 1$. For simplicity we can work with independent retransmissions i.e. $P_{e^{(j)}|e^{(j-1)} \wedge \ldots \wedge e^{(0)}} = P_{\text{e}}$, where we used the shorthand notation $P_{e^{(j)}|e^{(j-1)} \wedge \ldots \wedge e^{(0)}} \equiv P_{(e^{(j)}=1)|(e^{(j-1)}=1) \wedge \ldots \wedge (e^{(0)}=1)} \cdot$

Explicit expression for up to three retransmissions are in this case given by

$$p_{\text{BLE,eff,1}} = P_{\text{e}} \left(P_{\text{fn}} + (1 - P_{\text{fn}}) P_{\text{e}} \right) , \qquad (12)$$

$$p_{\text{BLE,eff,2}} = P_{\text{e}} (P_{\text{fn}} + (1 - P_{\text{fn}}) P_{\text{e}} \\ \cdot (P_{\text{fn}} + (1 - P_{\text{fn}}) P_{\text{e}})) , \qquad (13)$$

$$p_{\text{BLE,eff,3}} = P_{\text{e}} (P_{\text{fn}} + (1 - P_{\text{fn}}) P_{\text{e}} \\ \cdot (P_{\text{fn}} + (1 - P_{\text{fn}}) P_{\text{e}} \\ \cdot (P_{\text{fn}} + (1 - P_{\text{fn}}) P_{\text{e}}))) . \qquad (14)$$

If we denote the set of binary sequences of length n by $\mathcal{S}_n$, the probability $P_{\text{r,n}}$ for having n retransmissions is given by

$$P_{\text{r,n}} = \sum_{(x_0, x_1, \ldots x_{n-1}) \in \mathcal{S}_n} \prod_{i=0}^{n-1} (1 - P_{\text{fn}})^{x_i} P_{\text{fp}}^{1-x_i}$$
$$\times \prod_{j=0}^{n-1} P_{(e^{(j)}=x_j)|(e^{(j-1)}=x_{j-1}) \wedge \ldots \wedge (e^{(0)}=x_0)} , \qquad (15)$$

which again reduces to (2) for $n = 1$. Again we may set $P_{(e^{(j)}=x_j)|(e^{(j-1)}=x_{j-1}) \wedge \ldots \wedge (e^{(0)}=x_0)} = P_{\text{e}}$ for independent transmissions. In this case Eq. 15 simplifies to

$$P_{\text{r,n}} = (P_{\text{e}}(1 - P_{\text{fn}}) + (1 - P_{\text{e}}) P_{\text{fp}})^n . \qquad (16)$$

The total number of expected transmissions $\langle \Delta T_n \rangle$ is then simply given by

$$\langle \Delta T_n \rangle = \sum_{i=1}^{n} i \cdot P_{\text{r,i}} . \qquad (17)$$

APPENDIX B
SUPERVISED AUTOENCODER FOR SUPERVISED ANOMALY DETECTION

The supervised autoencoder is a neural-network-based supervised anomaly detection algorithm. It enjoys a number of advantages compared to for example shallow neural network classifiers applied directly to the input data that arise from the fact that the classifier is not applied to the data directly but rather to the bottleneck features of an autoencoder. Therefore it is able to work in heavily imbalanced scenarios as the one considered in this work and does not suffer from highly correlated input.

For the construction of the SAE we leverage the approach put forward in [21] albeit in a supervised anomaly detection setting. Similar to their work we use a regular multi-layer fully-connected autoencoder with L_2 loss as a backbone. In addition, we jointly train a fully-connected classifier operating on the bottleneck features that is trained using cross entropy loss, see Figure 7 The idea behind the joint training is to allows the autoencoder to not only build a reduced representation but also to build bottleneck features that contain most discriminative information for the classification task. We also experimented with using features derived from the reconstruction error (measured using cosine distance and reduced Euclidean distance) as additional input to the classifier as proposed in [21] but found no improvement.
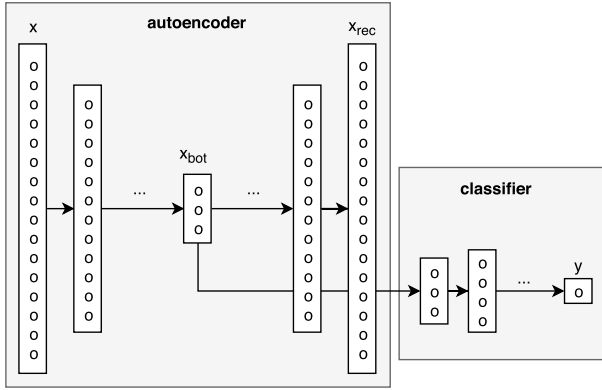
Fig. 7. Architecture for supervised anomaly detection using a jointly trained supervised autoencoder (x: input, x_{rec}: reconstructed input, x_{bot}: bottleneck features, y: predicted label).

There are multiple ways of preventing overfitting in this setting: early stopping, reducing the bottleneck dimension, implementing the SAE as a denoising autoencoder [22] or regularization using dropout [23]. In our case dropout regularization both in the classifier as well as in the autoencoder itself proved most effective.

The network configuration reads for the autoencoder [FC(d,25), FC(25,10), FC(10,3), FC(3,10), FC(10,25), Lin(25,d)] and for the classifier [FC(3,10), FC(10,5), Lin(5,2), SM] with FC(x,y) $\equiv$ [Lin(x,y), BN, ReLU, DO] and input dimension d. Here Lin(x,y) denotes a linear transformation layer, BN a Batch Normalization-layer [24], ReLU a ReLU activation layer, DO a dropout layer at a dropout rate fixed via hyperparameter tuning (both 0.2) and SM a softmax activation layer. Optimization is performed using the Adam optimizer [25] at learning rate 0.001. To stabilize training oversampling the minority class samples by a factor of 100 turned out to be beneficial.

APPENDIX C
SCHEDULING PROBABILITY OF A SYSTEM WITH FINITE RESOURCES

In (4), $P(T_{\text{j}} \leq T_{\text{c}})$ highly depends on the load of the system, since it is mainly a scheduling problem. Based on the resource distribution P_{res} which is discussed in Appendix E, we can formulate the probability $P(T_{\text{j}} \leq T_{\text{c}})$ of scheduling the initial transmission arriving at time slot $t_0 > 0$ and $j - 1$ retransmissions within a time constraint T_c as follows,

$$
\begin{aligned}
&P(T_{\text{j}} \leq T_{\text{c}}) \\
&= \sum_{k_0=0}^{T_c - j T_{\text{RTT}} - 1} P_1(t_0, k_0) \\
&\quad \times \sum_{k_1 = k_0 + T_{\text{RTT}}}^{T_c - (j-1) T_{\text{RTT}} - 1} P_1(t_0 + k_0 + T_{\text{RTT}}, k_1 - k_0 - T_{\text{RTT}}) \\
&\quad \cdots \sum_{k_j = k_{j-1} + T_{\text{RTT}}}^{T_c - 1} P_1(t_0 + k_{j-1} + T_{\text{RTT}}, k_j - k_{j-1} - T_{\text{RTT}})
\end{aligned}
\tag{18}
$$

where $P_1(t_0, \Delta_t)$ is the probability that a packet that has arrived at t_0 is scheduled in time slot $t_0 + \Delta_t$. Under the assumption that the resource distribution function is not diverging, the initial argument of P_1 in (18) is set to t_0. As mentioned before, P_1 is the scheduling probability for an additional transmission assuming that this single transmission does not affect the system probabilities. So, this means that from the slots t_0 till the slot $t_0 + \Delta_t - 1$ the system is fully loaded and the observed transmission is not scheduled (random scheduling). We allow only in slot $t_0 + \Delta_t$ a lower load or the random scheduler picks the observed transmission. Hence, this is expressed by,

$$
\begin{aligned}
&P_1(t_0, \Delta_t) \\
&\approx \sum_{k_0 = N_{\text{res}}}^{N_{\text{max}}} P_{\text{res}}(k_0, t_0) \left(1 - \frac{N_{\text{res}}}{k_0 + 1} \right) \\
&\quad \times \left(\sum_{k_1 = N_{\text{res}}}^{N_{\text{max}}} P_{\text{res}}(k_1 | N_{\text{res}}) \left(1 - \frac{N_{\text{res}}}{k_1 + 1} \right) \right)^{(\Delta_t - 1)} \\
&\quad \times \left(\sum_{k=0}^{N_{\text{res}} - 1} P_{\text{res}}(k | N_{\text{res}}) + \sum_{k = N_{\text{res}}}^{N_{\text{max}}} P_{\text{res}}(k | N_{\text{res}}) \frac{N_{\text{res}}}{k + 1} \right),
\end{aligned}
\tag{19}
$$

where P_{res} is the resource distribution function, which is discussed in more detail in Appendix E. The scheduling probability P_1 is discussed in further detail in Appendix D.

APPENDIX D
SCHEDULING PROBABILITY IN A MODERATELY LOADED FINITE SYSTEM

The scheduling probability P_1 as the probability that a transmission arriving at t_0 is scheduled after Δ_t TTIs is given as

$$
\begin{aligned}
&P_1(t_0, \Delta_t) \\
&= \sum_{k_0 = N_{\text{res}}}^{\infty} P_{\text{res}}(k_0, t_0) \left(1 - \frac{N_{\text{res}}}{k_0 + 1} \right) \\
&\quad \times \sum_{k_1 = N_{\text{res}}}^{\infty} P_{\text{res}}(k_1 | k_0) \left(1 - \frac{N_{\text{res}}}{k_1 + 1} \right) \\
&\quad \cdots \sum_{k_{(\Delta_t - 1)} = N_{\text{res}}}^{\infty} P_{\text{res}}(k_{(\Delta_t - 1)} | k_{(\Delta_t - 2)}) \left(1 - \frac{N_{\text{res}}}{k_{(\Delta_t - 1)} + 1} \right) \\
&\quad \times \sum_{k=0}^{N_{\text{res}} - 1} P_{\text{res}}(k | k_{(\Delta_t - 1)}) + \sum_{k = N_{\text{res}}}^{\infty} P_{\text{res}}(k | k_{(\Delta_t - 1)}) \frac{N_{\text{res}}}{k + 1}.
\end{aligned}
\tag{20}
$$

As obvious, P_1 crucially depends on the resource distribution function $P_{\text{res}}(N, t)$, which is the probability that N resources arrive at time slot t, and its probability distribution conditioned on the previous number of resource arrivals $P_{\text{res}}(k_t | k_{t-1})$. The properties and formulation of this distribution is evaluated more in detail in Appendix E.

However, $P_1(t_0, \Delta_t)$ the exact formulation of $P_1(t_0, \Delta_t)$ poses computational problems due to the infinite sums and the exponential growth of computation for increasing Δ_t.

Hence, we introduce Lemma 1 to simplify the computation of the scheduling probability.

Lemma 1: For a moderately loaded system with $\sum_{k=0}^{N_{\max}} P(k,t) \approx 1$ and $N_{\max} \gtrapprox N_{\mathrm{res}}$, the resource distribution function is approximated for sufficiently large time slots t by $P_{\mathrm{res}}(N_t,t) \approx \sum_{k=0}^{N_{\mathrm{res}}-1} P_{\mathrm{res}}(k,t-1)P_{\mathrm{res}}(N_t|k) + \sum_{k=N_{\mathrm{res}}}^{N_{\max}} P_{\mathrm{res}}(k,t-1)P_{\mathrm{res}}(N_t|N_{\mathrm{res}})$.

Proof: Assuming a converging behavior of the resource distribution function, there exits a time slot t_0 and a lower bound $N_{\min}$ and an upper bound $N_{\max}$, such that $\sum_{k=N_{\min}}^{N_{\max}} P_{\mathrm{res}}(k,t) \approx 1$ for all $t \geq t_0$. Additionally for an non-heavily loaded system which is required for URLLC traffic, we assume $N_{\max} \gtrapprox N_{\mathrm{res}}$. Also, the lower bound is assumed to be sufficiently large, $N_{\min} > N_{\max} - N_{\mathrm{res}}$.

The resource distribution function at time slot $t_1 > t_0$ is formulated as

$$P_{\mathrm{res}}(N_{t_1},t_1) = \sum_{N_{t_1-1}=0}^{\infty} P_{\mathrm{res}}(N_{t_1-1},t_1-1)P_{\mathrm{res}}(N_{t_1}|N_{t_1-1}). \tag{21}$$

The sum can be divided into two regions, below N_{res} and above. Since $P_{\mathrm{res}}(N,t) \to 0$ for any $N > N_{\max}$ and $N_{\max}$ is close to the number of resources of the system, we approximate the conditional function by assuming N_{res} resources in the previous time slot. For a moderately loaded system, this is a valid assumption, since the resource probability distribution function is decreasing fast for $N > N_{\mathrm{res}}$. Only for small arguments N_t close to 0 the deviation increases. However, the constraint regarding $N_{\min}$, which prevents underutilization, ensures that $P_{\mathrm{res}}(N_{t_1}|N_{t_1-1})$ is getting very small in that region anyway. Hence, we approximate the conditional resource distribution probability for $N_{t-1} > N_{\mathrm{res}}$ by

$$P_{\mathrm{res}}(N_t|N_{t-1}) \approx \sum_{n=0}^{N_t} P_{\mathrm{A}}(n) \cdot P_{\mathrm{H}}(N_t - n|N_{\mathrm{res}}). \tag{22}$$

□

Using Lemma 1 for $\Delta_t > 0$, the scheduling probability is approximated by

$$
\begin{aligned}
&P_1(t_0, \Delta_t)\\
&\approx \sum_{k_0=N_{\mathrm{res}}}^{N_{\max}} P_{\mathrm{res}}(k_0,t_0)\left(1 - \frac{N_{\mathrm{res}}}{k_0+1}\right)\\
&\times \left(\sum_{k_1=N_{\mathrm{res}}}^{N_{\max}} P_{\mathrm{res}}(k_1|N_{\mathrm{res}})\left(1 - \frac{N_{\mathrm{res}}}{k_1+1}\right)\right)^{(\Delta_t-1)}\\
&\times \left(\sum_{k=0}^{N_{\mathrm{res}}-1} P_{\mathrm{res}}(k|N_{\mathrm{res}}) + \sum_{k=N_{\mathrm{res}}}^{N_{\max}} P_{\mathrm{res}}(k|N_{\mathrm{res}})\frac{N_{\mathrm{res}}}{k+1}\right).
\end{aligned}
\tag{23}
$$

APPENDIX E
RESOURCE DISTRIBUTION FUNCTION OF A SYSTEM WITH FINITE RESOURCES

The resource distribution function describes the probability of having a specific number of resources N to be scheduled at a time slot t. With the previously mentioned system setup mainly three components contribute to resource allocations. The first are the packet arrival processes of the individual UEs. These pose the main component. Additionally, there are the HARQ retransmissions, which depend on the error probability of the underlying channel code for a specific channel. However, to simplify analysis a uniform BLER has been assumed for each of the transmissions. The last component is the overload of the previous time slot due to resource shortage, which is then transfered to the next time slot. Hence, the resource distribution is described as

$$P_{\mathrm{res}}(N,t) = \sum_{n,m,o \in \mathcal{S}} P_{\mathrm{A}}(n)P_{\mathrm{H}}(m,t-T_{\mathrm{RTT}})P_{\mathrm{OL}}(o,t-1), \tag{24}$$

with $\mathcal{S} = \{n,m,o \in \mathbb{N}_0 : n+m+o = N\}$, $N \in \mathbb{N}_0$ and $t \in \mathbb{Z}$ and $P_{\mathrm{A}}(n)$ being the probability of having n arrival processes, $P_{\mathrm{H}}(m)$ being the probability of having m HARQ retransmissions in time slot t and $P_{\mathrm{OL}}(o,t)$ being the probability of having o resources overload in the time slot t to be transferred to the next time slot.

The probability of arrival processes for N_{UE} UEs is described straightforwardly as a binomial distribution for $n \leq N_{\mathrm{UE}}$ i.e.

$$P_{\mathrm{A}}(n) = \binom{N_{\mathrm{UE}}}{n}(P_{\mathrm{A,UE}})^n(1-P_{\mathrm{A,UE}})^{N_{\mathrm{UE}}-n}, \tag{25}$$

and otherwise $P_{\mathrm{A}}(n) = 0$, where $P_{\mathrm{A,UE}}$ is the probability of packet arrival of one UE at one time slot. This modeling implicitly assumes that one UE can only have at most one new transmission per time slot.

Formulating P_{H} is a bit more intricate since for a limited allowed number of HARQ retransmissions initial packet transmissions have to be distinguished probability-wise from HARQ retransmissions. This would require to distinguish initial transmissions and first, second up to n retransmissions as separate dependencies in P_{res} and would require to specify scheduling rules, which would considerably complicate the whole analysis. However, this limitation is overcome by allowing unlimited HARQ retransmissions. This implies that this approach cannot be used to analyze for example single-retransmission HARQ since the HARQ retransmission term assuming an infinite number of retransmissions as implemented below would drastically overestimate the system load from HARQ retransmissions hence punishing FPR too much. Hence, P_{H} is given for $t \geq 0$ and $n \leq N_{\mathrm{res}}$ as

$$
\begin{aligned}
P_{\mathrm{H}}(n,t) = \sum_{k=n}^{\infty} P_{\mathrm{res}}(k,t)&\binom{\min(k,N_{\mathrm{res}})}{n}\\
&\cdot P_{\mathrm{r}}^n(1-P_{\mathrm{r}})^{(\min(k,N_{\mathrm{res}})-n)},
\end{aligned}
\tag{26}
$$

and otherwise $P_{\mathrm{H}}(n,t) = 0$ except for $P_{\mathrm{H}}(0,t<0) = 1$, where N_{res} is the number of system resources per time slot, T_{RTT} is the HARQ RTT and the single-retransmission probability $P_{\mathrm{r}} = (1-P_{\mathrm{fn}})P_{\mathrm{e}} + P_{\mathrm{fp}}(1-P_{\mathrm{e}})$ as in (3). Because of notational reasons, we chose using an infinite sum, which can be easily replaced by splitting the sum at N_{res} and replacing the part from $N_{\mathrm{res}}+1$ to ∞ by $\left(1 - \sum_{k=0}^{N_{\mathrm{res}}} P_{\mathrm{res}}(k,t-T_{RTT})\right)\binom{N_{\mathrm{res}}}{n}(P_{\mathrm{r}})^n(1-P_{\mathrm{r}})^{(N_{\mathrm{res}}-n)}$.

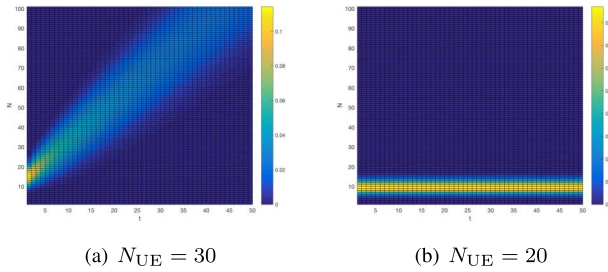(a) $N_{\mathrm{UE}} = 30$ (b) $N_{\mathrm{UE}} = 20$

Fig. 8.   Non-converging and converging resource distribution functions over time of an overloaded system (left) and a balanced system (right).

Still, this way of evaluating the HARQ-contributions in the system still overestimates the load from retransmissions and therefore underestimates the system performance.

The last component P_{OL} is simply defined by a back reference to the resource distribution function in the previous slot i.e.

$$P_{\mathrm{OL}}(n,t) = \begin{cases} P_{\mathrm{res}}\left(N_{\mathrm{res}} + n, t\right), & \text{if } t \geq 0 \wedge n > 0 \\ \sum_{k=0}^{N_{\mathrm{res}}} P_{\mathrm{res}}(n,t), & \text{if } t \geq 0 \wedge n = 0 \\ 1, & \text{if } t < 0 \wedge n = 0 \\ 0, & \text{otherwise} \end{cases} \quad (27)$$

For the sake of simplicity, we may assume $T_{\mathrm{RTT}} = 1$. This assumption makes the resource distribution function at time slot t only dependent on the previous time slot $t-1$ and is a valid assumption for the evaluated early HARQ schemes.

Here, the interesting question is, if the resource distribution converges for $t \to \infty$. By simulating the propagation of $P_{\mathrm{res}}(N,t)$ over t, we gain an insight on that question, as presented in Figure 8. As obvious in Figure 8(a), choosing the parameters such that the system is massively overloaded results in divergence of the resource distribution function. However, in case of a balanced system the resource distribution function shows a strong convergence behavior, as noticeable in Figure Figure 8(b). From (24), the conditioned resource distribution function for $t > 0$ and $N_t \geq N_{t-1} - N_{\mathrm{res}}$ follows as

$$P_{\mathrm{res}}(N_t | N_{t-1}) = \sum_{n=0}^{N_{\mathrm{up}}} P_{\mathrm{A}}(n) P_{\mathrm{H}}(N_{\mathrm{up}} - n | N_{t-1}), \quad (28)$$

where $N_{\mathrm{up}} = N_t - \max(N_{t-1} - N_{\mathrm{res}}, 0)$ and for $m \leq \min(N_{\mathrm{res}}, N_{t-1})$

$$P_{\mathrm{H}}(m | N_{t-1}) = \binom{\min(N_{t-1}, N_{\mathrm{res}})}{m}$$
$$\cdot (P_{\mathrm{r}})^m (1 - P_{\mathrm{r}})^{(\min(N_{t-1}, N_{\mathrm{res}}) - m)}, \quad (29)$$

otherwise $P_{\mathrm{H}}(m | N_{t-1}) = 0$.

REFERENCES

[1] R. El-Hattachi and J. Erfanian, "NGMN 5G white paper," Next Generation Mobile Networks, Frankfurt, Germany, Tech. Rep. 1.0, Feb. 2015.
[2] T. Fehrenbach, R. Datta, B. Göktepe, T. Wirth, and C. Hellge, "URLLC services in 5G low latency enhancements for LTE," in *Proc. 88th IEEE Veh. Technol. Conf. (VTC Spring)*, Aug. 2018, pp. 1–6.
[3] *IMT Vision—Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*, document ITU-R M.2083-0, 2015.
[4] K. Takeda, L. H. Wang, and S. Nagata, "Latency reduction toward 5G," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 2–4, Jun. 2017.
[5] MCC Support, "Report of RAN1#92b v1.0.0," 3GPP, Tech. Rep. R1-1805801, Apr. 2018, pp. 127–128.
[6] G. Berardinelli, S. R. Khosravirad, K. I. Pedersen, F. Frederiksen, and P. Mogensen, "Enabling early HARQ feedback in 5G networks," in *Proc. 83rd IEEE Veh. Technol. Conf. (VTC Spring)*, May 2016, pp. 1–5.
[7] G. Berardinelli, S. R. Khosravirad, K. I. Pedersen, F. Frederiksen, and P. Mogensen, "On the benefits of early HARQ feedback with non-ideal prediction in 5G networks," in *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, Sep. 2016, pp. 11–15.
[8] N. Strodthoff, B. Göktepe, T. Schierl, W. Samek, and C. Hellge, "Machine learning for early HARQ feedback prediction in 5G," in *Proc. IEEE Globecom (GC) Workshops*, Dec. 2018, pp. 1–6.
[9] B. B. Göktepe, S. Fähse, L. Thiele, T. Schierl, and C. Hellge, "Subcode-based early HARQ for 5G," in *Proc. IEEE Int. Conf. Commun. (ICC) Workshops*, May 2018, pp. 1–6.
[10] *Study on Channel Model for Frequency Spectrum Above 6 GHz*, document 3GPP, TR 38.900 v15.0.0, Jun. 2018.
[11] *Agressive Early Hybrid ARQ for NR*, document TR1-1700647 3GPP RAN1-NR#1 Spokane (US), Fraunhofer HHI, Jan. 2017.
[12] P. Branco, L. Torgo, and R. Ribeiro, "A survey of predictive modeling under imbal-anced distributions," *ACM Comput. Surv.*, vol. 49, no. 2, pp. 1–31, 2016.
[13] G. Collell, P. Drazen, and P. Kaustubh, "Reviving threshold-moving: A simple plug-in bagging ensemble for binary and multi-class imbalanced data," 2016, *arXiv:1606.08698*. [Online]. Available: https://arxiv.org/abs/1606.08698
[14] B. Kiran, T. Dilip, and P. Ranjith, "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos," *J. Imag.*, vol 4, no. 2, p. 36, 2018.
[15] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, 2006, pp. 233–240.
[16] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.
[17] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2008, pp. 413–422.
[18] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[19] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS) Workshop Autodiff*, 2017, pp. 1–4.
[20] *MCC Support*, document 3GPP TS 38.212 v1.0.1, Sep. 2017, pp. 19–30.
[21] B. Zong *et al.*, "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent. (ICML)*, 2018, pp. 1–19.
[22] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1096–1103.
[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.
[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

Nils Strodthoff received the Dr.rer.nat. degree (Hons.) *(summa cum laude)* from the Technische Universität Darmstadt in 2012. He joined the Machine Learning Group, Fraunhofer Heinrich Hertz Institute, as a Research Scientist, in 2017. He studied physics at Georg-August-Universität Göttingen and Imperial College London from 2005 to 2009. He held postdoctoral appointments at Ruprecht-Karls-Universität Heidelberg from 2012 to 2015 as well as the Lawrence Berkeley National Laboratory, USA, from 2016 to 2017, supported by the DFG Research Fellowship. He has authored more than 25 peer-reviewed articles presented at numerous international conferences. He is a machine learning enthusiast and particularly interested in applications of deep learning in communications, biomedical signal processing, and proteomics. He has received the HGS-HIRe Excellence Award in 2011. He has received scholarships from Deutsche Studienstiftung and HGS-HIRe.

Barış Göktepe received the B.Sc. and M.Sc. degrees in electrical engineering from the Berlin University of Technology (TUB), Berlin, Germany, in 2014 and 2017, respectively. Since 2013, he has been with the Fraunhofer Heinrich Hertz Institute (HHI). He joined the Multimedia Communications Group, Video Coding and Analytics Department, in 2015. He has authored more than 10 peer-reviewed journal and conference papers in the area of next-generation mobile communication. He is actively participating and contributing to 3GPP RAN1 5G standardization in the field of HARQ feedback, NR for unlicensed, and URLLC. His research interests include channel coding, information theory, and feedback in wireless communication.

Cornelius Hellge received the Dipl.Ing. degree in media technology from the Ilmenau University of Technology, Germany, in 2006, and the Dr.Ing. degree in electrical engineering and computer science from the Berlin University of Technology, Germany, in 2013. As a student, he was a Visiting Researcher at the Network Coding and Reliable Communications Group, Massachusetts Institute of Technology. He has authored more than 60 journal and conference papers, predominantly in the areas of video communication and next generation mobile networks. He holds more than 30 internationally issued patents and patent applications. He is currently the Head of the Multimedia Communications Group, Fraunhofer Heinrich Hertz Institute, where he is responsible for various scientific as well as industry-funded projects. Together with his team, he is regularly contributing to 3GPP (RAN1, RAN2, and SA4), ISO/IEC MPEG, JCT-VC, and JVET.

Thomas Schierl received the Dipl.Ing. degree (Hons.) in computer engineering from the Berlin University of Technology (TUB), Germany, in December 2003, and the Dr.Ing. degree (Hons.) in electrical engineering and computer science from the Berlin University of Technology (TUB) in October 2010. He is currently the Head of the Department of Video Coding and Analytics, Fraunhofer Heinrich Hertz Institute, Berlin, Germany. The department is covering all research topics related to video coding, communications, and machine learning. He is the coauthor of various IETF RFCs and ISO/IEC MPEG standards. His research interests include system integration of video codecs as well as delivery of real-time multimedia data over mobile networks such as 4G and 5G. Together with the ISO/IEC JCT1/SC29/WG11 Moving Picture Experts Group (MPEG) he received the Technology and Engineering Emmy Award from the National Academy of Television Arts and Sciences for the Development of the MPEG-2 Transport Stream in 2014.

Wojciech Samek (M'13) received the Diploma degree in computer science from the Humboldt University of Berlin in 2010 and the Ph.D. degree in machine learning from the Technische Universität Berlin in 2014. In the same year, he founded the Machine Learning Group, Fraunhofer Heinrich Hertz Institute, where he is currently the Director. He is associated with the Berlin Big Data Center. He was a Scholar with the German National Academic Foundation and a Ph.D. Fellow with the Bernstein Center for Computational Neuroscience Berlin. He was visiting Heriot-Watt University in Edinburgh and the University of Edinburgh from 2007 to 2008. In 2009, he was with the NASA Ames Research Center, Intelligent Robotics Group, Mountain View, CA, USA. His research interests include interpretable deep learning, neural networks compression, federated learning, and computer vision.