Machine Learning for early HARQ Feedback Prediction in 5G

Nils Strodthoff, Barış Göktepe, Thomas Schierl, Wojciech Samek and Cornelius Hellge
Fraunhofer Heinrich Hertz Institute
Einsteinufer 37, 10587 Berlin
Email: [nils.strodthoff, baris.goektepe, thomas.schierl, wojciech.samek, cornelius.hellge]@hhi.fraunhofer.de

***Abstract*—We put forward Machine Learning methods to predict the decodability of a received message before the end of the actual transmission in an Early Hybrid Automatic Repeat reQuest (E-HARQ) feedback scheme. Here we focus on a single retransmission setting for ultra-reliable and low-latency communication (URLLC) and demonstrate how more elaborate classification approaches can significantly improve the effective bit error rate towards the URLLC requirement of 10^{-5} with only small latency overhead. We stress the importance of a careful determination of the classifier's working point and discuss appropriate ways of discriminating between different classifiers. We demonstrate the feasibility of our procedure for different signal-to-noise ratios as well as subcode lengths and discuss practical implications of our findings.**

***Index Terms*—Communication systems, ultra-reliable and low-latency communication, physical layer, Hybrid Automatic Repeat reQuest, Machine Learning, imbalanced classification**

I. Introduction

The next generation 5G wireless mobile networks is driven by new emerging use cases, such as Ultra-Reliable Low Latency Communication (URLLC) [1]. To mention a few URLLC applications, tactile internet, industrial automation and 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 emerged 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, such as LTE, is still unclear. In particular, the Hybrid Automatic Repeat reQuest (HARQ) procedure poses a bottleneck for achieving aforementioned latencies. HARQ is a physical layer mechanism that employs feedback to transmit at aggressive target Block Error Rates (BLERs), while achieving robustness of the transmission. However, it imposes an additional delay on the transmission, designated as HARQ Round Trip Time (RTT). This leads to the abandonment of HARQ for the 1 ms end-to-end latency use case of URLLC [5]. This decision implies that the code rate is lowered such that a single shot transmission is possible. 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 enable HARQ for URLLC becomes a critical issue.

The paper is organized as follows: In Sec. II, we give a short overview on state-of-the-art early HARQ enhancements based on the Variable Node Reliability (VNR) metric. In the following Sec. III, we present a novel Machine Learning method to exploit full VNR information for early HARQ prediction. In Sec. IV, we evaluate the Machine Learning based approaches and compare them to state-of-the-art early HARQ schemes. We summarize and conclude in Sec. V.

II. Early HARQ Feedback - Overview

Early HARQ (E-HARQ) approaches aim to reduce the HARQ RTT by providing the feedback on the decodability of the received signal at an earlier stage, e.g. during transmission of the transport block. 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 waits for the whole signal representing the total codeword to apply a decoder. 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 generation from the decoding process, which introduces a misprediction probability since the actual outcome is not known afore. By taking that step, it is possible to use only a portion of the transmission time interval (TTI), designated as sub-TTI, and thus reducing the time from initial reception to transmitting the feedback (T_1). The sub-TTI which is used for feedback prediction, contains a subset of OFDM symbols out of the total number of OFDM symbols making up the TTI which is then used for the actual decoding of the transmission. The outcome of the actual decoding of the TTI corresponds to the ground truth for the feedback prediction. In total, the retransmission is scheduled earlier, hence also reducing the HARQ RTT, see Fig. 1. The time T_2 which corresponds to the remaining time after feedback

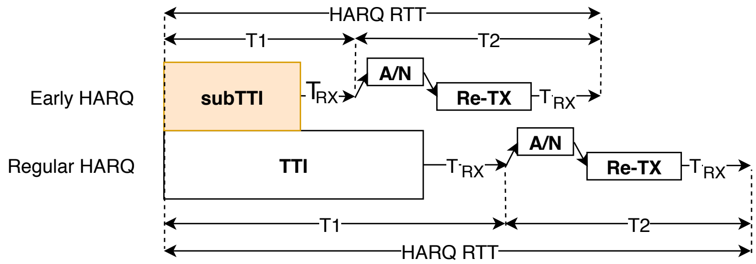


Fig. 1. Timeline of regular HARQ compared to early HARQ. (HARQ RTT: HARQ round trip time; TTI: transmission time interval; T_{RX}: processing time at the receiver; A/N: ACK/NACK feedback transmission; Re-TX: retransmission; T_1: time from initial reception to feedback transmission T_2: time from transmission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

generation, i.e. transmission time for feedback, scheduling and transmission of the retransmission, is assumed to be constant in this work. Hence, the improvement of HARQ RTT is limited to the reduction of T_1.

E-HARQ approaches to reduce the HARQ RTT have been first discussed in [6] and [7]. This approach estimates the Bit Error Rate (BER) based on the Log-Likelihood Ratios (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 k^{th} bit b_k is defined as:

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

Having the LLRs of a subcode or the whole codeword allows to calculate an estimated BER for the received signal vector, as stated here:

$$B\hat{E}R = \frac{1}{M} \sum_{k=1}^{M} \frac{1}{1 + |L(b_k)|} \,, \tag{2}$$

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 [8] and [9]. The authors propose to exploit the code structure to improve the prediction performance. In case of Low-Density Parity-Check (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 defined iteratively via

$$\Lambda_k^{(j)} = \Lambda_k^{(j-1)} + \sum_{m \in \mathcal{M}(k)} \beta_{m,k}^{(j)} \,, \tag{3}$$

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 of the jth iteration. Here we use the superscript j in $\Lambda_k^{(j)}$ to denote the decoder iteration after which the posteriori LLRs were extracted. Again, the a posteriori LLRs are mapped to the same metric for each decoder iteration, designated as VNR:

$$VNR_j = \frac{1}{M} \sum_{k=1}^{M} \frac{1}{1 + |\Lambda_k^{(j)}|} \,, \tag{4}$$

where M is the length of the subcodeword and VNR_j is the VNR corresponding to a certain decoder iteration. Hence, VNR_0 corresponds to $B\hat{E}R$. In [8], the authors applied a hard threshold to the VNR of the last decoder iteration, i.e. , VNR_5.

However, these schemes loose significant information since only a single value is applied to a threshold. The evolution of VNRs is expected to provide improved prediction accuracy compared to the state-of-the-art schemes.

III. Machine Learning for Early HARQ

We aim to optimize E-HARQ by means of Machine Learning techniques. This includes more complex input features as well as more involved classification algorithms.

From a Machine Learning perspective we are dealing with an inherently imbalanced binary classification problem. The task is to predict the decoding result of a given transmission using data that is available after the first few decoder iterations. Here we focus on VNRs from the zeroth up to the fifth decoder iteration as input features in order to leverage information about the evolution of the sub-codeword during the decoding process. Concerning classification algorithms we also restrict ourselves to two well-established algorithms with fundamentally different underlying principles, regularized logistic regression and Random Forests leveraging [10]. Logistic regression is a commonly used classification algorithm, where the log-odds of a binary output variable is modeled as a linear combination of the classifier's input variable. A Random Forest is an ensemble method that relies on combining predictions from multiple decisions trees which in turn reach a classification decision by a sequence of binary decisions tracing a path from their root node down to one of their terminal nodes, see e.g. [11] for a detailed description of both algorithms. A more detailed treatment of the impact of both the choice of input features and the choice of classification algorithm is subject to future investigations.

The main complication arises from the heavily imbalanced setting of the problem that results from the typical base block error rates of the order of 10^{-3} or even smaller. A lot of research has been devoted to developing effective ways of dealing with imbalance, see e.g. [12] for a review, that can be broadly subdivided into cost-sensitive learning, rebalancing techniques and threshold moving. Here we focus in particular on the latter, see e.g. [13] and references therein, which allows to adapt any trained classifier that outputs probabilities for the predicted classes by modifying the decision threshold. If one varies the decision threshold rather than considering a single fixed value, one can obtain for example Receiver-Operation curves (ROC) or Precision-Recall (PR) curves. Here we focus on Precision-Recall curves as these have been shown to better reflect the classifier's performance for highly skewed datasets [14]. ROC- or PR-curves allow to discriminate between different classification algorithms consider their overall discriminative power rather than focusing on a single decision threshold.

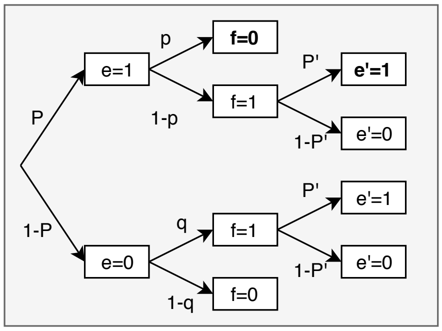


Fig. 2. Probabilistic model for single-retransmission HARQ (terminal nodes marked in bold face lead to an effective block error). The random variable $e(e')$ designates a block error in the initial transmission or retransmission, respectively. Similarly the sent feedback is quantified using the random variable f.

However, during the implementation a particular decision threshold has to be fixed, which requires a careful consideration between the classifier's false negative rate (FNR) and false positive rate (FPR) that counteract each other and that are the key output figures of the classifier from the system point of view. Here we follow the nomenclature in the Machine Learning community designating minority class i.e. non-decodable examples as positive examples, even though the opposite assignement is often used in the Communications literature. Hence the FNR quantifies the error made by sending ACK feedback given a non-decodable input, which has more severe consequences from the system's point of view compared to the error arising from sending a NACK feedback for decodable input. The reason is that the former results in a failure to deliver the message within the latency constraint whereas the latter represents an unnecessary retransmission, which only degrades the spectral efficiency of the HARQ scheme. The impact of FNR and FPR on the system performance is most easily illustrated in a simple probabilistic model, see Fig. 2. Designating the base code block error rate (BLER) by P, the block error probability after retransmission by $P' \simeq P$ and the conditional probability $P(f = 0|e = 1)$, which is to be identified empirically with the classifier's FNR, by p one straightforwardly derives the following relation for the effective block error probability for single-retransmission HARQ,

$$p_{\text{BLE,eff}} = P \cdot (p + (1-p)P') \,, \tag{5}$$

where we work with the simplifying assumption of a perfect feedback channel. However, effects of an imperfect feedback channel can be simply accommodated in this framework by means of effective FNRs/FPRs. The restriction to a single retransmission arises from the hard latency threshold of 1 ms that allows at most one retransmission [8]. Soft latency or reliability constraints might allow multiple retransmissions and therefore further improvements of the effective error probability but will not be considered in our setting. The FPR primarily impacts the spectral efficiency, which we characterize for simplicity by the expectation value $\langle T \rangle$ of total transmissions for single-retransmission HARQ relative to the baseline of regular HARQ with $1+P$ retransmissions. Although, spectral efficiency does not impact the reliability and latency directly, in a practical system with limited amount of available resources the latency constraint is hurt in the overloaded regime. In the same model as before and designating the conditional probability $P(f = 1|e = 0)$, which is to be identified empirically with the FPR, by q one finds

$$\langle T \rangle = 1 + (P(1-p) + (1-P)q) \,, \tag{6}$$

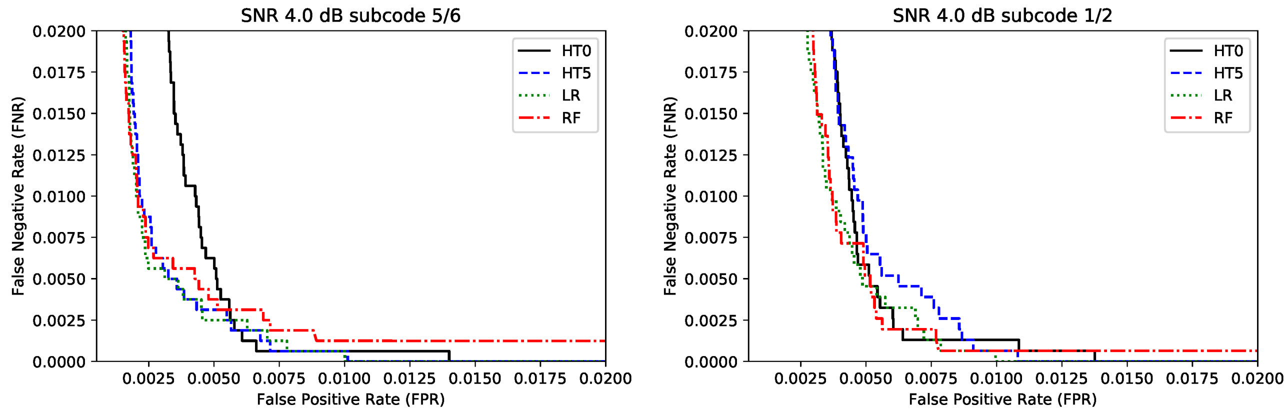
with leading order contribution $1 + P + q$.

In a system with unlimited resources it might be desirable to choose an arbitrarily small FNR, which lets the effective block error rate approach the lower bound $P \cdot P'$ corresponding to perfect HARQ feedback in the sense that $p = 0$. However, this comes in turn with a high FPR, which in the extreme case amounts to predicting NACK for every single transmission. This is obviously not a viable procedure in realistic setting with limited resources. It even turns out that the system characteristics determine an optimal working point for the prediction algorithm. At this point we restrict ourselves to the comparison of different FNR-FPR curves to discriminate between different classifiers.

We can summarize the problem setting as follows: Given the decoder outputs of the first five decoder iterations i.e. $(\Lambda^{(0)}_{i,k=1\ldots M}), \ldots, (\Lambda^{(5)}_{i,k=1\ldots M})$, where M denotes the length of the subcode under consideration and the subscript $i = 1 \ldots N$ enumerates the training samples, we aim to predict the final outcome of the decoding process l_i of the full codeword. Our solution involves the following steps:

1) For a given training sample we calculate the classifier's input features $x_i = (VNR_0(\Lambda^{(0)}_{i,k})), \ldots, VNR_5(\Lambda^{(5)}_{i,k})))$ using Eq. 4.
2) We train a binary classifier (regularized logistic regression/Random Forest) on tuples (x_i, l_i), where $i = 1 \ldots N$.
3) We assess classification performance on unseen test data and estimate FNR p and FPR q empirically from the classifier's confusion matrix.
4) Finally, we analyze the system performance in terms of effective block error rate $p_{\text{BLE,eff}}$ and expected number of transmissions $\langle T \rangle$ as defined by Eqs. 5 and 6.

We stress again that the advantage of the proposed solution lies in decoupling the feedback generation from the final



(a) 4.0 dB subcode 5/6

(b) 4.0 dB subcode 1/2

Fig. 3. Evaluating prediction performance based on FNR-FPR curves: Comparing improved predictors (LR,RF) to baseline results (HT0,HT5).

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

Transport block size	360 bits
Channel Code	Rate-1/5 LDPC BG2, Z = 36, see [15]
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 ns, 2.9 GHz, 3.0 km/h
Equalizer	Frequency domain MMSE
Decoder type	Min-Sum
Decoding iterations	50
VNR iterations	5

decoding process. As a consequence, the task of estimating the outcome of the decoding process for a given codeword based on the outcome of the first few decoder iterations applied to a corresponding subcodeword is entirely different from the actual decoding process itself for which very efficient algorithms exist. For a given SNR and channel model the classifier has to be trained only once and there are no additional signaling overheads compared to regular HARQ during runtime.

IV. RESULTS

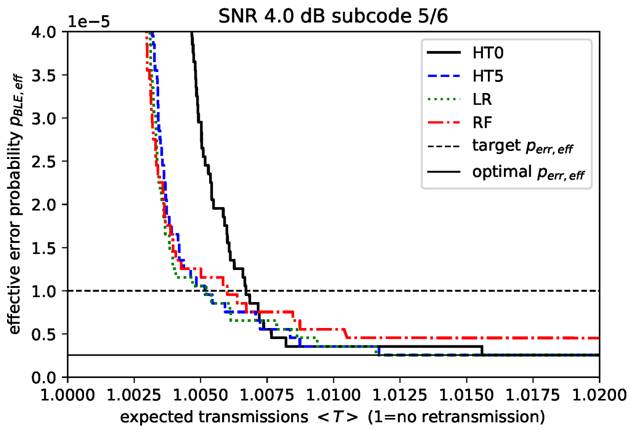
A. Prediction Performance

We compare classification performance of different classifiers based on the area under the PR curve (AUC-PR), where a perfectly discriminative classifier would amount to a score of 1, and FNR-FPR curves as discussed above. As external parameters we vary the SNR between 3.5 dB and 4.0 dB and subcodes 1/2 and 5/6 for a pedestrian channel model, see Tab. I for detailed parameters. The two SNR-values were chosen as the corresponding baseline BLERs are small enough to allow us to reach effective BLERs of the order 10^{-5} as required for URLLC services. On the other hand the baseline BLER decreases and hence the imbalance in the classification problem further increases with increasing SNR requiring even larger training and test sets. For this reason we decided not to consider SNRs beyond 4.0 dB in this study. The simulation setup used to produce training and test data follows the one reported in [8]. In all cases we use 1M transmissions with independent channel realizations for training and evaluate on a test set comprising 1M transmissions. Hyperparameter tuning was performed once at 4.0 dB at subcodelength 5/6 on a separate evaluation set comprising 1M records.

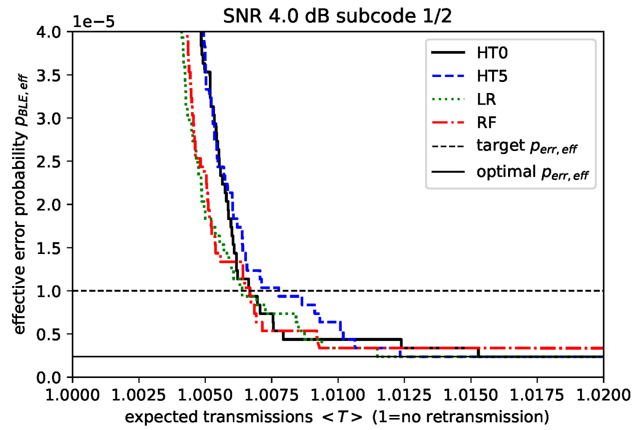
TABLE II
COMPARING CLASSIFICATION PERFORMANCE BASED ON AUC-PR.

model	SNR SC	BLER	AUC-PR
HT0	4.0dB 5/6	0.001601	0.818
HT5	4.0dB 5/6	0.001601	0.900
LR	4.0dB 5/6	0.001601	0.903
RF	4.0dB 5/6	0.001601	**0.905**
HT0	4.0dB 1/2	0.001540	0.806
HT5	4.0dB 1/2	0.001540	0.806
LR	4.0dB 1/2	0.001540	0.835
RF	4.0dB 1/2	0.001540	**0.837**
HT0	3.5dB 5/6	0.002843	0.844
HT5	3.5dB 5/6	0.002843	0.921
LR	3.5dB 5/6	0.002843	0.921
RF	3.5dB 5/6	0.002843	**0.924**
HT0	3.5dB 1/2	0.002842	0.828
HT5	3.5dB 1/2	0.002842	0.821
LR	3.5dB 1/2	0.002842	**0.858**
RF	3.5dB 1/2	0.002842	0.855

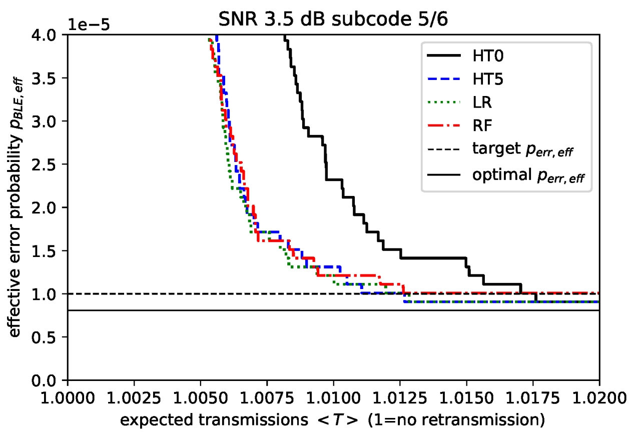
We benchmark hard threshold classifiers on the zeroth (HT0) [6] and fifth (HT5) decoder iteration [8] compared to (L_2-regularized) logistic regression (LR) and Random Forests (RF) using VNRs from the zeroth to fifth decoder iteration as input features. The results for test-set AUC-PR are compiled in Tab. II. As reported in [8] for subcode 5/6 the usage of the fifth decoder iteration rather than the initial VNR values leads to an improved classification performance. Interestingly, the converse applies to the case of subcode 1/2, where using the VNRs from the fifth decoder iteration even worsens the classification performance compared to the zeroth iteration baseline. In all considered cases the algorithms operating on the full set from zeroth to fifth decoder iteration show an improved classification performance compared to the current



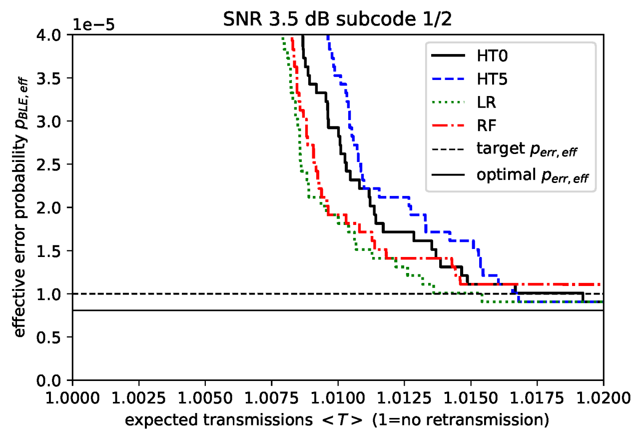
(a) 4.0 dB subcode 5/6



(b) 4.0 dB subcode 1/2



(c) 3.5 dB subcode 5/6



(d) 3.5 dB subcode 1/2

Fig. 4. Evaluating system performance based on $p_{\mathrm{BLE,eff}}$-$\langle T\rangle$ curves: Improvement from using more complex predictors (LR,RF) compared to baseline results (HT0,HT5). Target $p_{\mathrm{err,eff}}$ designates the URLLC target value of 10^{-5} and optimal $p_{\mathrm{err,eff}}$ denotes the effective error probability for perfect feedback in the sense of a vanishing FNR.

state-of-the-art HT5. Logistic Regression and Random Forests reach a similar level of performance in all cases. The largest performance gains for more complex prediction methods to single VNR baselines are obtained for the subcode 1/2 cases.

Even though these AUC-PR results reflect the overall discriminative power of the different classifiers on the given dataset, the classifiers' behavior in the FNR-region below 0.01 is of primary practical interest. In Fig. 3 we compare the classification performance for SNR 4.0 dB at subcodelength 5/6 and 1/2. Fig. 3 reflects a similar ranking that already emerged in Tab. II. However, the ranking based on FNR-FPR curves allows a more finegrained discrimination between different classifiers in dependence of the chosen working point. This is illustrated for example in Fig. 3(a), where for small FPR ranges HT5 is favored over HT0 whereas for larger FPRs HT0 outperforms HT5.

Unfortunately in the extremely unbalanced regime it is difficult 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. Apart from increasing sample sizes this can only be circumvented by fitting appropriate parametric descriptions to the FNR-curve which could then be used to extrapolate to very small FNR values. As a matter of fact the FNR-curve in the small FNR range is well-described by a power-law behavior.

B. System Performance

For the evaluation of the system performance we focus on the effective block error rates, as obtained from Eq. 5 that can be reached in combination with different classification algorithms, see Fig. 4. This is a valid consideration as long as the system resources are able to properly accommodate the retransmission overhead introduced by the FPR. To produce Fig. 4 we approximated the conditional probabilities p and q by the empirical FNR/FPR as extracted from the classifier's confusion matrix, which summarizes the classification performance in terms of false/true positive/negative classification results. The baseline block error probability P was also estimated from test set statistics. For simplicity we assumed that $P' = P$ i.e. that the block error probability for the retransmission coincides with that of the original transmission. Fig. 4 shows the expected behavior, the effective error probability decreases with increasing latency and eventually approaches the effective error probability for regular HARQ feedback.

A qualitatively consistent picture arises from the comparison in Fig. 4: On the one hand, for the larger subcode 5/6 the fifth decoder iteration HT5 is favored over the zeroth iteration HT0 over most of the parameter range. The more complex algorithms LR and RF represent a slight improvement over the current state-of-the-art HT5, where LR seems to perform slightly better than RF in particular in low FNR-regime that is most relevant for applications. On the other hand, for the smaller subcodelength 1/2 over most of the parameter range HT0 is favored over HT5. Interestingly the HT0 results is only marginally influenced by the subcodelength, which is most likely related to the fact that it only estimates an effective error rate across the codeword that is most likely even for subcode 1/2 long enough to give a reasonable estimate. In this setting the largest improvement from using more complex prediction methods is observed. This is consistent with expectations in the sense that decreasing the subcodelength leads to a more difficult prediction problem that can profit more from more complex methods. At 4.0 dB an effective error probability is achievable with the presented methods, whereas for 3.5 dB, where the gap between the target and the optimal block error probability is already much smaller, it is still achievable albeit requiring considerably more HARQ overhead as the effective error probabilities already start to flatten at this point. However, it is worth stressing that the target error probability of 10^{-5} should primarily seen as fixing the order of magnitude rather than a hard threshold value. For example an effective error probability of $2 \cdot 10^{-5}$ is easily achievable at 3.5 dB. Finally we can use Fig. 4 to estimate the HARQ overhead we have to accept when decreasing the subcodelength from 5/6 to 1/2 while maintaining the same effective error probability. Using the LR result and a threshold value of $1{\cdot}10^{-5}$ or $2{\cdot}10^{-5}$ at SNR 4.0 dB and SNR 3.5 dB we find that the expected number of retransmissions increases by approximately 40%.

Our results clearly demonstrate that effective block error rates of the order of 10^{-5} are achievable with a small loss compared to the optimal block error rate at both investigated SNR values and subcodelengths. In all cases and independently of the algorithm the classifier profits from the information of the VNR evolution with the decoder iterations. The evaluation both for Random Forests as well as logistic regressions is amenable to efficient implementations on hardware and are therefore not only of theoretical but potentially also of practical interest.

V. Conclusion

In this work we explored the prospects of improving subcode-based early HARQ as introduced in [8] by using more elaborate classification methods to predict the decoding result ahead of the final decoder iteration. We demonstrated that quantitative improvements over baseline results are possible by considering a larger set of input features in conjunction with more complex classification algorithms. Furthermore we showed that URLLC requirements can be satisfied for 3.5 and 4.0 dB at different subcodelengths and for different channel models with only small latency overheads.

This work represents a first exploratory study towards quantitatively improved prediction methods in this setting. The most obvious extension from the system point of view is to incorporate implications of limited system resources that eventually singles out an optimal working point on the FNR-FPR curve that is determined by system properties. Furthermore, exploiting more elaborate input features like intra-message features such as LLRs or history features incorporating the receiver's knowledge about channel properties from earlier transmissions might prove useful to further improve the classification performance, in particular on vehicular channels. These issues are currently under investigation.

References

[1] R. El-Hattachi and J. Erfanian, "NGMN 5G White Paper," tech. rep., Next Generation Mobile Networks (NGMN), 02 2015.

[2] T. Fehrenbach, R. Datta, B. Göktepe, T. Wirth, and C. Hellge, "URLLC Services in 5G Low Latency Enhancements for LTE," in 2018 IEEE 88th Vehicular Technology Conference (VTC Spring), August 2018.

[3] ITU, "IMT Vision Framework and overall objectives of the future development of IMT for 2020 and beyond ," tech. rep., ITU, 2015.

[4] K. Takeda, L. H. Wang, and S. Nagata, "Latency reduction toward 5G," in 2017 IEEE Wireless Communication, June 2017.

[5] MCC Support, "Final Report of 3GPP TSG RAN WG1 #92b v1.0.0," tech. rep., 3GPP, 04 2018.

[6] G. Berardinelli, S. R. Khosravirad, K. I. Pedersen, F. Frederiksen, and P. Mogensen, "Enabling early harq feedback in 5g networks," in 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring), pp. 1–5, May 2016.

[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 2016 International Symposium on Wireless Communication Systems (ISWCS), pp. 11–15, Sept 2016.

[8] B. Göktepe, S. Fähse, L. Thiele, T. Schierl, and C. Hellge, "Subcode-based early HARQ for 5G," in 2018 IEEE International Conference on Communications Workshops (ICC), May 2018.

[9] Fraunhofer HHI, "Agressive Early Hybrid ARQ for NR," TDoc R1-1700647, 3rd Generation Partnership Project (3GPP), Jan 2017.

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

[11] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning. Springer Series in Statistics, New York, NY, USA: Springer New York Inc., 2001.

[12] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modelling under imbalanced distributions," CoRR, vol. abs/1505.01658, 2015.

[13] G. Collell, D. Prelec, and K. R. Patil, "Reviving threshold-moving: a simple plug-in bagging ensemble for binary and multiclass imbalanced data," CoRR, vol. abs/1606.08698, 2016.

[14] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," in Proceedings of the 23rd International Conference on Machine Learning, ICML '06, (New York, NY, USA), pp. 233–240, ACM, 2006.

[15] MCC Support, "3GPP TS 38.212 v1.0.1," tech. rep., 3GPP, 09 2017.