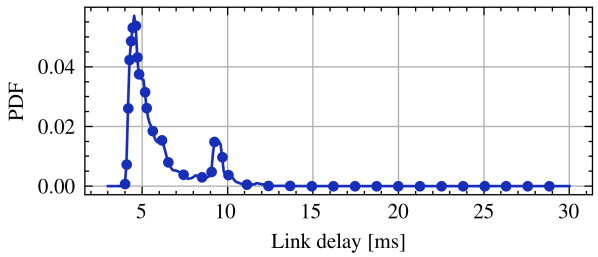
Data-Driven Latency Probability Prediction for Wireless Networks: Focusing on Tail Probabilities

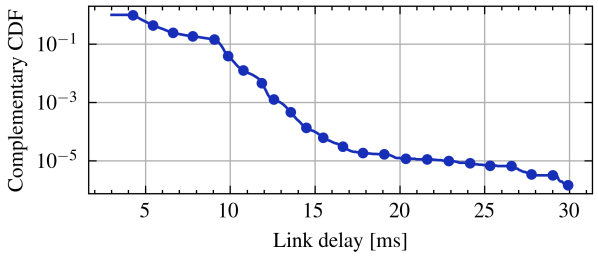
Samie Mostafavi, Gourav Prateek Sharma, James Gross
KTH Royal Institute of Technology,Stockholm, Sweden
{ssmos,gpsharma,jamesgr}@kth.se

***Abstract*—With the emergence of new application areas, such as cyber-physical systems and human-in-the-loop applications, there is a need to guarantee a certain level of end-to-end network latency with extremely high reliability, e.g., 99.999%. While mechanisms specified under IEEE 802.1AS time-sensitive networking (TSN) can be used to achieve these requirements for switched Ethernet networks, implementing TSN mechanisms in wireless networks is challenging due to their stochastic nature. To conform the wireless link to a reliability level of 99.999%, the behavior of extremely rare outliers in the latency probability distribution, or the tail of the distribution, must be analyzed and controlled. This work proposes predicting the tail of the latency distribution using state-of-the-art data-driven approaches, such as mixture density networks (MDN) and extreme value mixture models, to estimate the likelihood of rare latencies conditioned on the network parameters, which can be used to make more informed decisions in wireless transmission. Actual latency measurements of a commercial private and a software-defined 5G network are used to benchmark the proposed approaches and evaluate their sensitivities concerning the tail probabilities. Our benchmarks highlight how the proposed methods, aided by noise regularization, achieve an acceptable accuracy in the extreme 99.9999% latency probabilities.**

***Index Terms*—time-sensitive networking, ultra-reliable low latency, mixture density networks, extreme value theory**



(a) PDF linear-linear plot



(b) Complementary CDF log-linear plot

Fig. 1: Distribution of downlink latency, as observed from commercial off-the-shelf (COTS) private 5G system

I. Introduction

Traditionally, communication networks have been designed to provide best-effort connectivity between application endpoints without guaranteeing performance. In recent years, new application areas have emerged that require real-time, high-performance network communication, such as Cyber-Physical Systems (CPS) and Human-in-the-Loop (HITL) applications. These systems exchange information between components, including physical sensors, actuators, and computing devices, to support various applications, from industrial automation to virtual reality. To ensure that these applications function correctly and safely, it is necessary to guarantee a certain level of end-to-end network latency. The range of acceptable end-to-end latency varies from 1 to 50 ms, depending on the specific use cases, and necessitates an exceptionally high level of reliability ($\geq 99.999\%$), as noted in [1]. End-to-end latency for typical CPS and HITL applications refers to the time it takes for a packet of data to travel from the sensor source to the controller and from the controller back to the actuator. In such scenarios, delays in network communication can have serious consequences, such as delays in the response of a robotic system or human feedback. This requires a deep understanding of the underlying network infrastructure and developing novel techniques to bind latency with extremely high reliability. For switched Ethernet networks, these requirements can be achieved using mechanisms specified under IEEE 802.1 Time-Sensitive Networking (TSN). TSN is a suite of standards that specify mechanisms that enable time-critical traffic alongside best-effort traffic [2]. TSN employs various packet scheduling and traffic-shaping mechanisms, e.g., Time-aware Shaping (IEEE 802.1Qbv), to guarantee end-to-end latency. However, implementing TSN mechanisms for wireless networks is challenging due to their stochastic nature.

Wireless mobile communication links are susceptible to interference and potentially complicated channel scattering and fading patterns, unlike isolated wired links. Hybrid Automatic Repeat Request (HARQ) is used in most wireless communication schemes to ensure reliability by retransmitting lost or corrupted packets. It works by combining Forward Error Correction (FEC) with Automatic Repeat Request (ARQ) pro-

This work was supported by the European Commission through the H2020 project DETERMINISTIC6G (Grant Agreement no. 101096504).

978-1-979-8-3503-1090-0/23/$31.00 © 2023 European Union

tocols, allowing for error detection and correction at both the physical and link layers of the communication [3]. However, it introduces additional processing and retransmission time for lost or corrupted packets. This latency can depend on the specific HARQ implementation and the characteristics of the wireless channel, such as signal strength and interference. Ultimately, compared to wired medium, packets traversing the wireless link will experience a non-deterministic latency with 1) a higher average and 2) significant packet delay variation.

Despite the attempts made to reduce average latency in wireless communication technologies, such as Ultra-reliable Low-latency Communications (URLLC) improvements [4], there is still a considerable gap between the application latency requirements with extreme reliability levels of around 99.9999% and what state-of-the-art wireless networks offer. Conforming the wireless link to such a reliability level requires controlling the behavior of extremely rare outliers in the latency probability distribution or the tail of the distribution. The tail of the latency probability distribution refers to the portion that extends to the far right and contains low-probability latencies, which are less likely to occur than the bulk of the distribution. In other words, it represents the extreme values of the distribution as shown in Figure 1. By analyzing the tail of the latency distribution, it is possible to estimate the likelihood of rare latencies, which can be used to make more informed decisions in wireless transmission [5]. All in all, to achieve the requirements, latency must be analyzed using probabilistic models where the accuracy of the tail is of great importance.

A. Related Works

Latency probability prediction is a crucial component of network performance evaluation; especially, to support time-critical applications over wireless networks. In the literature, several approaches to characterize latency in communication networks have been proposed. These approaches could be essentially grouped into two categories: (i) model-based approaches and (ii) data-driven approaches. In model-based approaches, wireless networks and services are modeled, e.g., as queueing systems, and analytical tools (e.g., network calculus) are applied to derive bounds on latencies. For instance, stochastic network calculus was used to obtain probabilistic bounds on the end-to-end delay in a multi-hop wireless network in [6]. In contrast to model-driven approaches, data-driven approaches employ machine-learning methods to identify and learn relationships between latency and other variables using the measurement data from real systems. Khangura et al. identified bottleneck links and estimated the residual bandwidth via a neural network that was trained using a vector of packet dispersion values [7]. The output of the estimator produced a point estimate of the performance metric (i.e., available bandwidth). Point estimates (e.g., average throughput and average delay) are insufficient for scenarios where applications expect very high-quantile performance guarantees (e.g., 10 ms delay with a 99.999% guarantee). For such scenarios, complete probability distributions of performance metrics are necessary [8]. Within the data-driven approaches proposed in the literature, we focus on those that provide probability distributions for different performance metrics. Using the histogram-based approach, conditional round-trip time (RTT) probability distribution in IoT systems is estimated by Flinta et al. in [9]. In [10], Samani et al. estimated the conditional probability distributions of network performance metrics (i.e., frame rate and response time) based on the measurements taken from a test network. It was observed that mixture density networks (MDN) (Gaussian and Log-normal) and histograms could be leveraged to generate conditional distributions with their respective trade-offs. Similarly, authors in [11] proposed to use a mixture of Laplace distributions to characterize delay jitter in high-frequency and mobile communications. It is worth pointing out that these estimators are apt for predicting the bulk of latency distribution but not tail probabilities, which could not be ignored for applications demanding strict latency guarantees. In our previous work, the end-to-end latency in a multi-hop queuing system was characterized by exploiting Extreme-value Theory (EVT) in combination with MDNs [12]. The evaluation results showed the superiority of this method in predicting tail probabilities, over the traditional MDN approaches for conditional latency probability prediction. In this paper, we compare the performance of these estimators in predicting the latency probabilities of wireless networks.

B. Contributions

The main contribution of our work is a data-driven approach to characterize the latency of wireless networks in a probabilistic way with a focus on the extreme latencies, i.e., tail probabilities. Unlike model-driven approaches, we do not assume the wireless channel model (e.g., Rayleigh fading model) or any traffic profile (e.g., Poisson packet arrival). We investigate the performance of the state-of-the-art probability density estimation approaches, namely MDNs, specifically on the extremely rare latencies that occur with probabilities as small as 10^{-6}. In our previous work, we show how EVT models can be incorporated into MDNs to accurately relate the tail latency distributions to the conditions of the network. This approach is compared to the state-of-the-art methods through an extensive evaluation of two actual implementations of the 5G network: (i) COTS 5G and (ii) OpenAirInterface (OAI) 5G. The evaluations reveal the proposed approaches' advantages, disadvantages, and sensitivities when characterizing the tail latency behavior (e.g., at $> 99.999\%$ quantiles).

II. System Model and Problem Statement

As mentioned before, we strive to build a probabilistic understanding of the delay of the packets sent over the 5G network which is a bridge in the time sensitive networking (TSN) domain, or interfaces the end node with a controller. Assume packets $\{1, ..., N\}$ are being pushed to the wireless network with a fixed length B_s and at a fixed interval T_s that mimics the IEEE 802.1Qbv time cycles or sensor/actuator period packets arrival profile [13]. We denote the observed end-to-end delay of the packet n by Y_n and the transmission conditions by X_n. Due to the nature of the wireless medium,

packets may need to be re-transmitted to ensure reliability. Such mechanism is implemented in Hybrid-ARQ (HARQ) scheme which is widely used in the majority of wireless networks. These random re-transmissions which are the main contributors to the latency, could be related to the transmission conditions such as signal to interference and noise ratio (SINR) and modulation coding scheme (MCS) index that we model in X_n. Let the random variable Y with domain $\mathcal{Y} \subset \mathbb{R}_+$ model the packets' latency, its probability density could be expressed conditioned on the transmission conditions i.e. $\mathbb{P}[Y \mid X = \boldsymbol{x}]$ where $\boldsymbol{x}$ denotes the observed network state or condition.

Problem Statement: We are interested in predicting the probability density of latency from the network conditions, i.e., the probability that the packets face a certain delay, given the transmission conditions of the network, i.e.,

$$\hat{p}\left(Y \mid X = \boldsymbol{x}\right) \approx \mathbb{P}\left[Y \mid X = \boldsymbol{x}\right]. \tag{1}$$

In particular, the accuracy of $\hat{p}$ is of great importance for large values of Y or at the tail of the probability distribution. The tail probability is then characterized by $1 - \hat{p}\left(Y < y \mid X = \boldsymbol{x}\right)$.

III. Approach

The above task of estimating the wireless link's latency probabilities from the transmission conditions falls into the domain of conditional density estimation (CDE). In a non-conditional density estimation scenario, we can fit a parametric model such as a Gaussian mixture model (GMM) to the set of latency samples. This parametric density function is described by a finite-dimensional parameter θ [14]. For example, θ could be a vector of the weights, locations, and variances of the GMM density function. In CDE, we need to devise a function denoted by h_ω to map the values of X, i.e., conditions, into the space of the density function parameters θ in the form $\theta = h_\omega\left(\boldsymbol{x}\right)$. MDN is a well-known method that uses a fully connected neural network as h_ω to control the parameters of the conditional density estimate $\hat{p}_\theta$ [14]. In MDN, maximum likelihood estimation (MLE) is used for estimating the parameters ω, i.e., ω is chosen so that the conditional likelihood of the samples $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_N, y_N)\}$ is maximized. This is equivalent to minimizing the Kullback-Leibler divergence (KL-divergence) between the empirical data and the parametric density $\hat{p}_\theta$ [14].

The subsequent procedure involves determining the parametric distribution $\hat{p}_\theta$ employed in the MDN. In this regard, we primarily use GMM, a widely recognized and established parametric distribution, in addressing the requirements of this particular problem domain. However, our use case requires high accuracy at the tail of the fitted probability density. In our previous work [12], we investigated the use of Extreme value theory (EVT) models for the tail estimation in addition to the Gaussians for this task. Since the output of the GMM-based solutions is a density function whose tail probability decreases exponentially fast, it can lead to significant probability error for low probability events, e.g., in the order of 10^{-5} if the empirical distribution is not light-tailed. we use the following parametric mixture function that encapsulates the generalized

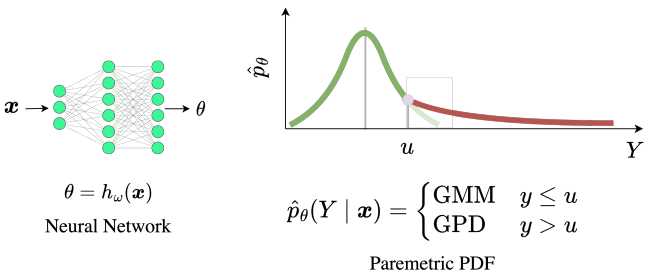


Fig. 2: Block diagram of the predictor, showing inputs, components, and outputs.

Pareto distribution (GPD) tail model in combination with the GMM to be fused with the neural network of a MDN in the form:

$$\hat{p}(y|\theta = h_\omega\left(\boldsymbol{x}\right)) = \begin{cases} f(y|\phi) & y \le u \\ [1 - F(u|\phi)]g(y|\beta, \xi, u) & y > u, \end{cases} \tag{2}$$

$f(y|\phi)$ and $F(y|\phi)$ represent the probability density function (PDF) and cumulative distribution function (CDF) of the GMM respectively, where ϕ stands for the parameter vector of the GMM. $g(y|\beta, \xi, u)$ denotes the GPD's PDF, where u is the tail threshold, ξ is the tail index, and β denotes the tail scale. For simplicity, we define θ as the collection of all parameters ϕ, β, ξ, and u. The structure of the predictor is illustrated in Figure 2. In this extreme value MDN scheme, tail parameters of the distribution (threshold u, tail index ξ, and scale β) are being estimated by the neural network in addition to the bulk distribution parameters.

IV. Methodology

To evaluate the proposed approach, we attempt to predict the latency of a real 5G network, i.e., Y, under different network conditions, i.e., X. For this task, first, we collect latency samples from the network to form a dataset that can be used to train and evaluate the MDN model. Then, we compare these models in terms of their accuracy in predicting the tail latency distribution. Other aspects such as the sensitivity to the number of training samples, noise regularization, and generalization accuracy are investigated as well. The following subsection introduces all the steps for benchmarking the predictors and conducting the experiments. [1].

Measurements Setup: The experimental setup consists of 2 nodes connected using a 5G network as shown in Figure 3. The latency samples are collected from 10ms, 172 bytes periodic transmission of timestamped packets from the end-node to the server node (uplink) and back (downlink). Since the nodes reside on different machines, we synchronized their clocks using precision time protocol (PTP) protocol on a dedicated separate interface. The desired network condition is also recorded for each latency sample. We conduct the measurements on two different 5G systems: a private COTS 5G, and an end-to-end

[1]The reproducible results and the datasets could be found at: https://github.com/samiemostafavi/wireless-pr3d

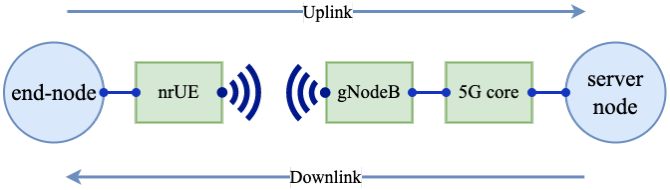


Fig. 3: Illustration for the setup used to gather measurement data from both COTS 5G and OAI 5G.

software-defined radio (SDR) 5G developed by OAI software alliance [15]. The SDR 5G system offers maximum flexibility in recording and manipulating the parameters of all layers. Both 5G networks operate in band 78, in time division duplex (TDD) mode, with 106 physical resource blocks (PRB) which occupies 40 MHz of bandwidth at 3.5 GHz.

In the COTS 5G experiment, we gathered 4 million uplink latency measurements in 11 hours, from 20 distinct locations within a 50 m^2 5G-covered conference room containing metallic chairs, whiteboards, and screens. The 5G base station was located in one corner of this rectangular-shaped room. The investigation did not reveal any variations in the tail behavior across diverse locations. Therefore, we analyze non-conditional latency prediction on the COTS 5G measurements. This approach is viable in cases where there is no information about the communication conditions, or there is no relation between the available conditions and the latency distribution. Examining the impact of sample size and the parametric model on the fitting accuracy remains essential, irrespective of the neural network's inherent bias-variance trade-off. In the SDR 5G experiment, we gathered 5 million uplink latency measurements in 15 hours. In the configuration employed, we located the base station and user equipment (UE) 1 meter apart and in line of sight. We were able to set the MCS index of the transmissions to a fixed value of 3, 5, or 7 throughout an entire round, in contrast to the link adaptation algorithm, which assigns a value of 15 in the absence of such a constraint. In this case we analyze conditional latency prediction approach.

Data Preprocessing: We observed that normalization and standardization of the data play a crucial part in achieving the best accuracy in training. For instance, the delay values are scaled to the millisecond scale and moved to around zero by subtracting the average delay. Conditions are normalized to be between 0 and 1. Moreover, applying noise regularization in mixture density networks is suggested by [16] to improve the accuracy of the fit. We investigate that by comparing trained models on two noise levels of 1ms and 3ms with the raw data.

Training: The MDN model is implemented using four fully connected hidden layers with [10, 100, 100, 80] neurons, respectively. It controls parameters of 15 Gaussian centers and optionally a GPD for modeling the tail. The output size of the neural network is equal to the number of parameters of the parametric density function $\hat{p}_\theta$, which is 48 when GPD is incorporated and 45 without GPD. We name the model with GPD component, Gaussian mixture extreme value model (GMEVM). Training the model is done by Adam optimizer in 4 rounds, each with 200 epochs and different learning rates: $[10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}]$, respectively. The batch size is set to 1/8 of the training dataset size. We observed that including training rounds with minimal learning rates, e.g., 10^{-4} or 10^{-5} is necessary to avoid underfitting in the tail region. In Table I, the training durations for different sizes of the training dataset are mentioned. MDN models are implemented and evaluated using CPU-only Tensorflow on a system equipped with Intel(R) Core(TM) i9-10980XE CPU operating at 3.00GHz.

TABLE I: Training durations for 1000 epochs

Training dataset size	1M	256k	64k
Batch size	128k	32k	8k
Epoch duration	4s	1s	300ms
Step duration	500ms	100ms	27ms
Total training time	66.6m	16.6m	5m

The MDN model estimates the PDF of the latency distribution from the observed latency measurements over time. In a slowly changing wireless environment, we can periodically collect new latency measurements and use them to re-train the MDN model. Therefore, it is crucial to carefully monitor the model's performance concerning the retraining frequency, which is limited by the training duration and dataset size.

Evaluation: In our study, we assess the efficacy of our trained models by conducting training sessions for ten different models per configuration. Each model is trained independently using a randomly chosen set of latency samples but similar hyper-parameters and MDN type to assess the performance deviations. We then use the obtained results to compute the minimum, maximum, and average predictions. Subsequently, we compare these predictions to the ground truth, which we established by measuring the system's latency. We collect many samples to ensure the accuracy of our ground truth measurements. To visualize this, we use a colored area between the minimum and maximum curves instead of only one. We consider a predictor to be effective when its average prediction aligns closely with the ground truth and exhibits lower variance. This approach enables us to gain a deeper understanding of the performance of our models and the extent of the variations.

V. Numerical Results

This section focuses on analyzing the performance of latency predictors in various practical scenarios and wireless communication technologies. The primary objective is to evaluate the sensitivity of these predictors to the number of samples and the complexity of the data.

A. Non-Conditional COTS 5G Latency Prediction

In a relatively simple approach, prediction of the latency probability could be carried out only from the samples and not incorporating any condition. Figure 4 shows the tail probability estimation of the COTS 5G latency for different models trained with different number of samples. It can be observed that the

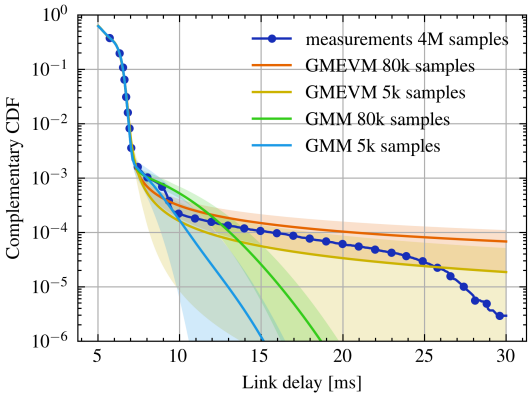


Fig. 4: COTS 5G uplink latency measurements vs parametric density fits with different number of samples and models

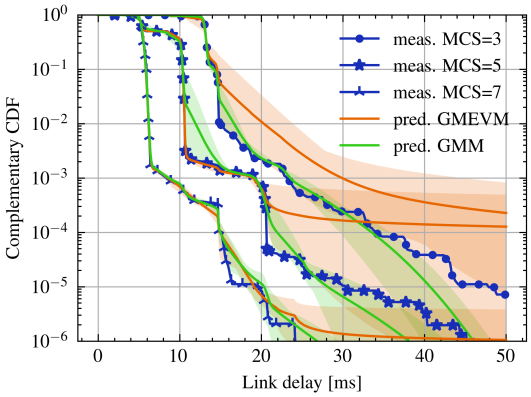


Fig. 5: Performance of MDN models trained without noise regularization. Number of samples: 1M (20%)

tail latency distribution of the GMM predictor falls exponentially with the received delay and therefore diverges from the empirical distribution whereas, the GMEVM predictor closely follows the empirical distribution. Next, we compared the impact of training data size on the accuracy of the fit for GMM and GMEVM. It can be expected for both predictors that their performance improves with the increasing number of training data samples. Our observations indicate that the effect of decreasing the sample size from the magnitude of 10^5 to 10^4 on the GMEVM is relatively minor. However, the accuracy is significantly compromised when reducing the sample size to 10^3, as depicted in the figure. The emergence of a distinct trend in the measurement curve as the tail probability approaches the order of 10^{-4} could explain the observed behavior. Our findings reveal that GMEVM can effectively capture this trend, whereas GMM cannot. In the next section, we assess the performance of the MDNs in the context of conditional scenarios and a more challenging latency profile.

B. Conditional SDR 5G Latency Prediction

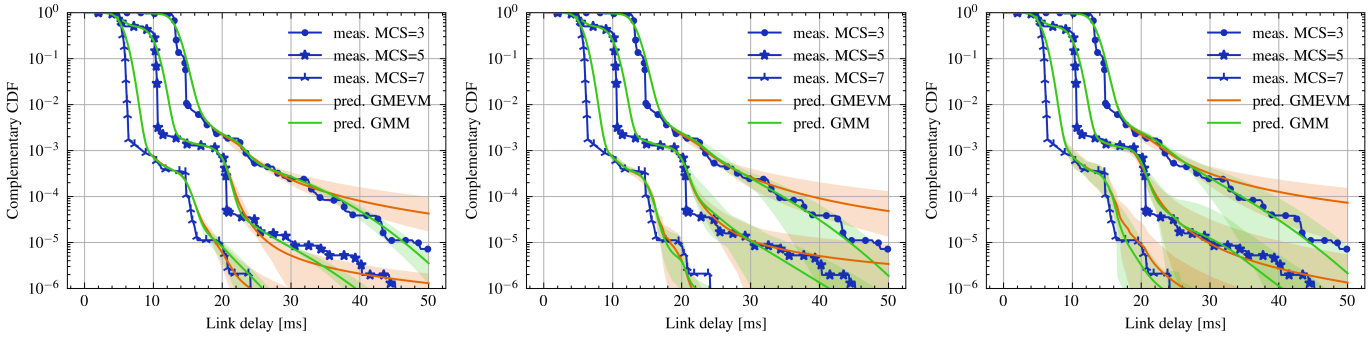
This section investigates the performance of MDNs in the context of conditional latency prediction. Condition vector X is related to the parametric distribution parameters θ through the multi-layer perceptron (MLP). It can potentially extrapolate the observed behavior of the training conditions to previously unseen ones. As a result, it may be possible to leave certain conditions unsampled and rely on the MLP's ability to predict their profile, further reducing the number of training samples while achieving a desirable level of accuracy.

In this context, it is expected that the link capacity will increase with the increase in the MCS index when the index is less than 15. Furthermore, an increase in the link capacity resulting from using a higher MCS index is anticipated to decrease the delay. Figure 5 depicts the measured tail probabilities for different MCS indices using all 5 million samples (1.6 million samples for each condition). Moreover, we trained GMM and GMEVM models using 1 million samples, including all three MCS conditions. In the presented results, the GMEVM scheme exhibits high variance and poor prediction accuracy across all three cases, while GMM performs well, as evidenced in the figure. We hypothesize that the measurements' bumpy and non-smooth tail profile is the reason for the observed struggles of GMEVM. To address this, we propose a noise regularization approach by adding random Gaussian noise with a variance of 1 millisecond to the latency samples. As shown in Figure 6, this technique significantly improves the performance of GMEVM even with lower training samples by smoothing out the tail. However, introducing noise regularization comes with a cost, which is lower accuracy in the bulk of the latency distribution. It is crucial to select the appropriate variance for the added noise. Figure 7 demonstrates the impact of using a larger variance of 3 milliseconds. The noise has degraded the low-tail regions' prediction accuracy but improved the high-tail predictions. Therefore, choosing the correct variance value is essential for noise regularization in GMEVM. Figure 6 also shows the models' accuracy concerning the training dataset's size. We observe higher uncertainty and error from the models in the case of less number of training samples.

In Figure 8, we evaluate the generalization capability of MDN models. For this purpose, we trained the models on a dataset that does not include latency samples with MCS=5. Consequently, MCS=5 is an unseen condition for the predictors. Both models follow the empirical tail but with a relatively high variance. GMM exhibits better performance than GMEVM with a lower variance. One can see an opportunity for further improving the performance of the models in handling unseen conditions in future work.

VI. Conclusions

This study uses state-of-the-art data-driven approaches, such as mixture density networks, to predict the latency of wireless links, particularly for extreme latencies that impact time-critical applications. Specifically, we analyze Gaussian mixture models and a novel approach that integrates extreme value models into the mixture of parametric distributions. Through our investigation, we examine the impact of the number of training samples, the complexity of the tail profile, and the generalization capabilities of these approaches. Our results



(a) Models trained with 1M samples (20%) (b) Models trained with 250k samples (5%) (c) Models trained with 64k samples (1.3%)

Fig. 6: SDR 5G uplink latency tail probability measurements conditioned for different MCS indices with 5M samples vs predictions of MDN models trained with different numbers of samples and added Gaussian noise

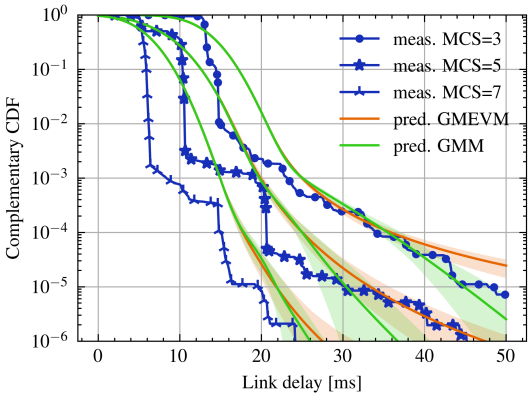


Fig. 7: Performance of MDN models trained with added Gaussian noise of variance 3ms. Number of samples: 250k samples (5%)

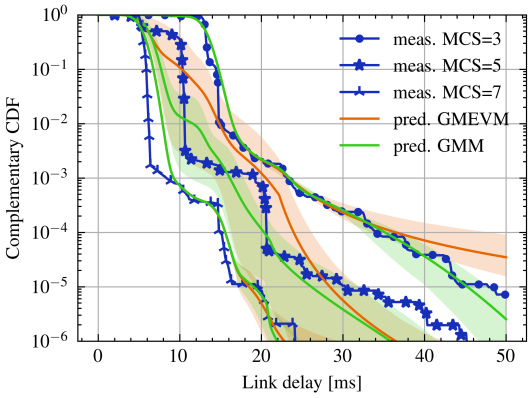


Fig. 8: Generalization capability of MDN models trained without MCS=5 samples. Number of samples: 0.6M (20%)

demonstrate that both approaches achieve acceptable accuracy with sufficient training samples. Additionally, we find that noise regularization improves the accuracy of the fit, particularly in the case of GMEVM when the tail profile is non-smooth. There is room for improvement in the generalization capabilities which can be addressed in future research. Overall, our study sheds light on the effectiveness of data-driven approaches in predicting wireless link latency and highlights areas for further improvement.

VII. Acknowledgements

We acknowledge the support of the European Commission for this research through the H2020 project DETERMINISTIC6G (Grant Agreement no. 101096504). We would also like to extend our thanks to Rishi Nandan, Núria Flores Espinosa, and Bixing Yan for their assistance in conducting the COTS 5G latency measurements.

References

[1] M. A. Lema, A. Laya, T. Mahmoodi, M. Cuevas, J. Sachs, J. Markendahl, and M. Dohler, "Business case and technology analysis for 5G low latency applications," *IEEE Access*, vol. 5, pp. 5917–5935, 2017.

[2] A. Nasrallah, A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein, and H. ElBakoury, "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 88–145, 2018.

[3] 3GPP, "Medium Access Control (MAC) protocol specification," Technical Specification (TS) 38.321, 3rd Generation Partnership Project (3GPP), 05 2019. Version 15.5.0.

[4] J. Ansari, C. Andersson, P. de Bruin, J. Farkas, L. Grosjean, J. Sachs, J. Torsner, B. Varga, D. Harutyunyan, N. König, *et al.*, "Performance of 5G trials for industrial automation," *Electronics*, vol. 11, no. 3, p. 412, 2022.

[5] G. P. Sharma, D. Patel, J. Sachs, M. D. Andrade, J. Farkas, J. Harmatos, B. Varga, H.-P. Bernhard, R. Muzaffar, M. K. Atiq, F. Duerr, D. Bruckner, E. Montesdeoca, D. Houatra, H. Zhang, and J. Gross, "Towards deterministic communications in 6G networks: State of the art, open challenges and the way forward," 2023.

[6] J. P. Champati, H. Al-Zubaidy, and J. Gross, "Transient Analysis for Multihop Wireless Networks Under Static Routing," *IEEE/ACM Transactions on Networking*, vol. 28, pp. 722–735, Apr. 2020.

[7] S. K. Khangura, M. Fidler, and B. Rosenhahn, "Machine learning for measurement-based bandwidth estimation," *Computer Communications*, vol. 144, pp. 18–30, Aug. 2019.

[8] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, 2018.

[9] C. Flinta, W. Yan, and A. Johnsson, "Predicting Round-Trip Time Distributions in IoT Systems using Histogram Estimators," in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–9, Apr. 2020.

[10] F. S. Samani, R. Stadler, C. Flinta, and A. Johnsson, "Conditional Density Estimation of Service Metrics for Networked Services," *IEEE Transactions on Network and Service Management*, vol. 18, pp. 2350–2364, June 2021.

[11] A. Sawabe, Y. Shinohara, and T. Iwai, "Delay Jitter Modeling for Low-Latency Wireless Communications in Mobility Scenarios," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 2638–2643, Dec. 2022.

[12] S. S. Mostafavi, G. Dán, and J. Gross, "Data-driven end-to-end delay violation probability prediction with extreme value mixture models," in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 416–422, 2021.

[13] S. S. Craciunas, R. S. Oliver, M. Chmelík, and W. Steiner, "Scheduling real-time communication in IEEE 802.1 Qbv time sensitive networks," in *Proceedings of the 24th International Conference on Real-Time Networks and Systems*, pp. 183–192, 2016.

[14] C. Bishop, "Mixture density networks," workingpaper, Aston University, 1994.

[15] F. Kaltenberger, G. d. Souza, R. Knopp, and H. Wang, "The OpenAir-Interface 5G new radio implementation: Current status and roadmap," in *WSA 2019; 23rd International ITG Workshop on Smart Antennas*, pp. 1–5, 2019.

[16] J. Rothfuss, F. Ferreira, S. Walther, and M. Ulrich, "Conditional density estimation with neural networks: Best practices and benchmarks," *arXiv:1903.00954*, 2019.