Throughput Prediction Using Machine Learning in LTE and 5G Networks

Dimitar Minovski, Niclas Ögren, Karan Mitra, *Member, IEEE*, and Christer Åhlund

Abstract—The emergence of novel cellular network technologies, within 5G, are envisioned as key enablers of a new set of use-cases, including industrial automation, intelligent transportation, and tactile internet. The critical nature of the traffic requirements ranges from ultra-reliable communications, massive connectivity, and enhanced mobile broadband. Thus, the growing research on cellular network monitoring and prediction aims for ensuring a satisfied user-base and fulfillment of service level agreements. The scope of this study is to develop an approach for predicting the cellular link throughput of end-users, with a goal to benchmark the performance of network slices. First, we report and analyze a measurement study involving real-life cases, such as driving in urban, sub-urban, and rural areas, as well as tests in large crowded areas. Second, we develop machine learning models using lower-layer metrics, describing the radio environment, to predict the available throughput. The models are initially validated on the LTE network and then applied to a non-standalone 5G network. Finally, we suggest scaling the proposed model into the future standalone 5G network. We have achieved 93 and 84 percent R^2 accuracy, with 0.06 and 0.17 mean squared error, in predicting the end-user's throughput in LTE and non-standalone 5G network, respectively.

Index Terms—5G, LTE, network slice, throughput, QoS

1 Introduction

The evolution of LTE to the latest 5G technology gives rise to a wide range of commercial services with vastly different characteristics and requirements [1]. Some examples include augmented and virtual reality, remotely control industrial services [2]. The softwarization within cellular networks enables a 5G multi-tenant ecosystem, where each tenant can be an individual end-user or a commercial service with specific access rights and privileges over the shared cellular resources [3]. Network slicing is a key technology in delivering the 5G multi-tenancy [3]. 3GPP, as a standardization body, already defined three high-level slices to support dedicated use-cases: enhanced mobile broadband (eMBB), ultra-reliable low latency (URLLC), and massive Internet of Things (MIoT) [4].

Emerging 5G services, such as critical industrial services, will push the service owners to buy network resources, i.e., slices, from the network operators to ensure top-notch Quality of Service (QoS) [3]. Consider industrial mining service with a fleet of autonomous driving vehicles. Such vehicles typically consist of multiple vehicle-to-everything (V2X) communications, with unique network requirements regarding speed and latencies. For instance, the vehicles may be supervised and remotely-controlled via video live-stream involving multiple high-resolution cameras, requiring gigabit speeds. Whilst the communication among the vehicles, such as positioning, platooning, and warning messages, require low throughput and low latency network interfaces. Thus, each vehicle can be simultaneously connected to multiple network slices in enabling optimal and reliable performance [5].

Several research studies have explored the relationships between QoS, the objective network quality, and the end-users' quality perception [6], [7]. Typical QoS metrics include throughput, delay, jitter, and packet loss [7]. According to Chen *et al.* [8], throughput emerges as the most important metric in affecting the end-users' perception. Moreover, throughput directly influences the productivity when evaluating the Quality of IoT-experience (QoIoT) in critical industrial services [9]. For instance, in remotely-controlled vehicles, insufficient throughput may lead to completely shutting down the production as live-video cannot be streamed [9]. The state of the art on evaluating network quality relies on machine learning (ML) for predicting QoS metrics within the LTE network [8]. For instance, the authors in [10] and [11] directly predict the available throughput per User Equipment (UE) using historical throughput and lower-layer information; while Yao *et al.* [12] adds connectivity maps in improving the QoS. However, the state of the art is limited to data-sets from one particular environment and LTE settings, with small sample sizes, lacking versatility, and risking biased ML models as the main challenges. In reality, LTE suffers from issues due to dynamic wireless environment and competing traffic, especially in V2X case-studies, where ML models should be trained on diverse driving conditions that consider a variety of LTE settings, such as Carrier Aggregation (CA) and MIMO [13].

- *Dimitar Minovski is with the Luleå University of Technology, 931 87 Skellefteå, Sweden, and also with InfoVista Sweden AB, 931 62 Skellefteå, Sweden. E-mail: dimitar.minovski@ltu.se.*
- *Niclas Ögren is with InfoVista Sweden AB, 931 62 Skellefteå, Sweden. E-mail: niclas.ogren@infovista.com.*
- *Karan Mitra and Christer Åhlund are with the Luleå University of Technology, 931 87 Skellefteå, Sweden. E-mail: {karan.mitra, christer.ahlund}@ltu.se.*

Manuscript received 12 October 2020; revised 7 June 2021; accepted 8 July 2021. Date of publication 26 July 2021; date of current version 3 February 2023.
(Corresponding author: Dimitar Minovski.)
Digital Object Identifier no. 10.1109/TMC.2021.3099397

The scope of this work is evaluation of network quality in the LTE and upcoming 5G networks with respect to throughput. We aim to develop a non-intrusive throughput prediction model in the context of 5G, having LTE as an anchor. Herein, the key challenge is to study the impact of novel technologies, such as network slicing, on QoS. Network slices in 5G are already foreseen to suffer performance degradation [14]. Besides the legacy LTE challenges, such as capacity and resource sharing, there will be a competing interference among the slices due to the non-orthogonal multiple access (NOMA) [14]. Herein, Ericsson warns of the dynamic nature of the network slices, suggesting that each slice has to undergo continuous performance monitoring [15]. In addition, Elayoubi *et al.* [16] advocates monitoring individual network slice performance for validating Service Level Agreements (SLAs) via various of QoS metrics. Thus, in our study, we consider an industrial use-case with vehicles that can simultaneously communicate via multiple network slices. The goal is to benchmark the performances of each slice with respect to throughput.

The following list summarizes the main contributions of this study:

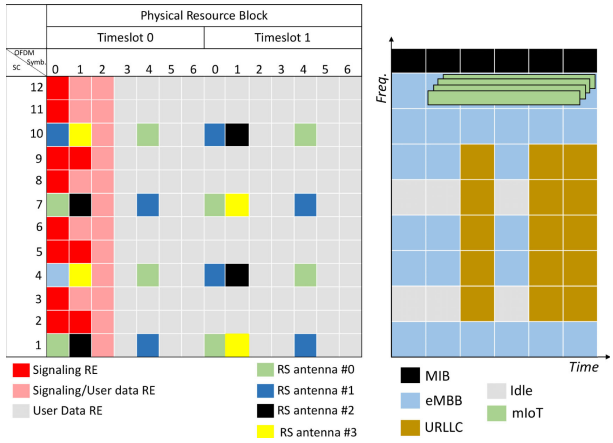
- Discussion on network slice requirements in IoT use-cases and possibilities for benchmarking individual slice performance.
- Analyzing the LTE and 5G cellular link throughput performance in use-case involving remotely-controlled vehicles. The data comes from real-life experiments from various driving conditions in urban, sub-urban, and rural areas, as well as stationary tests in crowded environments.
- Developing a network non-intrusive ML model-based for predicting LTE cellular link throughput, in both downlink (DL) and uplink (UL), tested in the above mentioned environmental condition.
- Extending and tuning the LTE ML model to a real-life non-standalone 5G environment, benchmarking cellular link throughput per network slice.

Our results show 93 percent accuracy (R^2) and 0.06 MSE in predicting the available throughput per UE, with a training and testing on various use-cases and LTE network settings. In addition, another ML model benchmarks the performance of a network slice in 5G non-standalone networks, predicting the available throughput per slice with 84 percent accuracy (R^2) and 0.17 MSE.

The rest of the paper is organized as follows: we review the background and related work in Section 2. Section 3 discusses the use-cases from this study, while Section 4 describes the experimental setup for conducting the experiments. Section 5 provides data-analysis and feature engineering. Section 6 presents the results, while Section 7 discusses the the results. Finally, Section 8 concludes the study.

2 Background and Related Work

This section consists of background information on cellular networks, LTE and 5G. Further, we discuss the state of the art studies utilizing ML to predict the available throughput per UE.



(a) LTE - distribution of resource elements (RE) per block, and reserved RSs for up to 4 antennas.

(b) 5G with Non-orthogonal multiple access among three network slices.

Fig. 1. Physical resource block schemes.

2.1 Background on Resource Sharing in LTE

In an LTE system, radio resources are divided into time and frequency units called Resource Blocks (RBs), each 180 kHz wide, with a duration of 0.5ms [17]. Fig. 1a illustrates a Physical RB (PRB) in frequency and time domains, 1ms long, constructed of two RBs. Each timeslot consists of 12 sub-carriers in the frequency domain and 7 OFDMA or SC-FDMA symbols in the time domain, in DL and UL, respectively [18]. In a simplified formula, the LTE throughput for a UE is determined by three factors, transmission time intervals (TTI) allocations to the UE, PRBs allocations per TTI, and modulation and coding scheme (MCS) [18]. The selected MCS to be used in a PRB is determined by the eNB, according to the UE's reported Channel Quality Indicator (CQI). The CQI calculation is vendor-specific, where the most influential metric is the SINR, alongside other metrics collected from the radio channel [17].

Fig. 1a shows the reserved reference signals (RSs) within each PRB, per antenna, whose purpose is to allow time and frequency alignment [18]. Note that in cases the UE supports only 2 antennas, the rest of the RSs are used to transmit user data. Some of the available RSs which we utilize in this study are shown in Table 1. A UE receives information from the serving and neighboring cells, which creates interference influencing the throughput when multiple end-users attempt to communicate simultaneously [17]. Further, throughput per UE is highly dependable on factors such as signaling overhead and current traffic load on the serving cell [19]. Shannon's channel capacity formula, shown in (1), utilizes RSs and compensating factors α and β to handle the static and average behavior of a specific system:

$$\begin{aligned} UE_1 &= 0.5 \times Bw \times \alpha \times Log_2(1 + \beta \times SINR_1), \\ UE_2 &= 0.5 \times Bw \times \alpha \times Log_2(1 + \beta \times SINR_2), \end{aligned} \tag{1}$$

where UE_1 and UE_2 are the capacities of two users competing for the same frequency in the same time. Bw is the assigned bandwidth in MHz; α takes into account the overhead loss generated from the guard spectrum and control signaling; while β is a value that caps the SINR according to the assigned MCS. Several research studies have utilized (1)

TABLE 1
Collected Parameters at the Source-Node

Parameter	Description
RSSI	*Carrier Received Signal Strength Indicator: comprises the linear average of the total received power observed only in OFDM reference symbols (in dB) [17].*
RSRP	*Reference Signal Received Power: the linear average over the power contributions of the resource elements that carry cell-specific reference signals (in dB) [17].*
RSRQ	*Reference Signal Received Quality: the ratio N*RSRP/(Carrier RSSI), where N is the number of resource blocks of the carrier RSSI measurement bandwidth (in dB) [17].*
SINR	*Signal to Interference plus Noise Ratio of the signal carrier best servings for the intervention seemed at all other sites/sectors, plus all the noise (in dB).*
RF-NC	*Summarized RSRP and RSRQ values measured on the neighboring cell (in dB) [17].*
p_a	*Energy reduction, compared to RS power, of user's data on OFDM symbols not carrying RS (in dB) [17].*
Band	*The frequency band on which the phone is connected to (in MHz).*
Num Carriers	*Number of utilized carriers per UE during connected mode (integer).*
RSPath Loss	*A function of RSSI and RSRP (in dB) [17].*
TA	*The cell measures the required timing advance based on the received UE signal arrival time (in sec) [17].*
Shannon	*Shannon's formula for measuring cell's capacity (in kbps), as measured in (1) in LTE and (3) in 5G [20].*
CellLoad	*PRBs, or REs, utilization at the cell, with respect to whole traffic (%), as measured in (2) [21].*
RF-RX0	*Aggregated RF values for UE's antenna 1 (in dB).*
RF-RX1	*Aggregated RF values for UE's antenna 2 (in dB).*
RF-RX2	*Aggregated RF values for UE's antenna 3 (in dB).*
RF-RX3	*Aggregated RF values for UE's antenna 4 (in dB).*
RI sum	*Summarized Rank Indicator value for all 4 UE's antennas (integer).*
CQI	*Reports the current channel quality, which is used to determine the transport block size [17] (measured on the 5G network).*
CRI	*A resource indicator for the channel state information [4] (available only on the 5G network).*

to calculate the available eNB resources [22], [23], from where the available UE throughput is extracted [21]. However, the cited studies [21], [22], [23], similarly as (1), lack integration of novel techniques, such as MIMO and CA, that can boost the available UE throughout. For instance, with CA, an UE can establish multiple carriers simultaneously, typically up to 5, raising the maximum bandwidth from 20 MHz to 100 MHz [18].

Our study combines (1) with the metrics from Table 1 as input features to the ML models. The goal is to predict the available throughput per UE, both in UL and DL, under various dynamic wireless environments, considering MIMO and CA.

2.2 Predicting Throughput: State of the Art

Bui *et al.* [24] surveys the most recent prediction techniques in anticipating network performance, highlighting the throughput as essential contextual network information. Generally, the throughput prediction methods are classified into active and passive. The former requires UEs to be in *connected mode* and stream packets, while the latter produces predictions with minimal or no network intrusiveness [25]. Our study's scope is the passive methods, since active tests produce additional network load and UE energy consumption [25]. Moreover, the dynamic wireless environment would increase the sampling of the active tests in vehicular cases, resulting in overloading the network.

Yue *et al.* [10] conducts detailed correlation analysis among RSs describing the Radio Frequency (RF) and the labeled throughput. The study includes ubiquitous use-cases, ranging from stationery, walking, local, and highway driving scenarios. As part of their findings, the measured throughput almost linearly grows with the increase of RSRP, RSRQ, and CQI. The study focuses on random forest as a ML model for predicting the available throughput per device, with features such as RSRP, RSRQ, CQI, and past throughput. The proposed model is, thus, fully intrusive as collecting CQI and past throughput data requires the UE to be in *connected mode*. Replicating the same model and results is difficult to achieve due to lack of description. That is, regarding feature creation, the study defines a prediction window size of 1, 5, and 10 seconds representing the length of past historical information. Herein, the authors do not discuss the details of the conducted active tests, such as selecting a single labeled throughput vales during the data collection process. Moreover the authors provide no indications on the way features are measured with respect to their corresponding labeled throughput value and how they are mapped together in terms of time resolutions. Finally, the paper does not consider CA and MIMO, and neither analyse the sensitivity of the training and test data-sets on the predictions with respect to biases and over/underfitting of the ML model.

Yao *et al.* [12] proposed bandwidth maps, where past experienced throughput is used to predict the current throughput per location. The authors apply their model in adaptive video streaming and audio applications to retrieve mean opinion scores. Jomrich *et al.* [26] took a step further in combining positioning data with few RF metrics to predict the throughput for moving vehicles. Using geo-location has many benefits in understanding factors such as path-loss, cell load, and shadowing [12]. However, we will not consider geo-location as a feature in our study due to the following reasons: First, there is a risk of being biased towards specific network settings within one location. The risk comes with future changes to the network settings, topology, and physical changes in the environment that will out-date the ML model. Second, uncertainty on dimensions of the geo-area needed for the ML model. Third, it requires an expensive training process, as data must be collected from every geo-location over long periods. Forth, the privacy aspect of measuring the geo-location of end-users in a possible commercial deployment.

Raca *et al.* [11] built a ML model for forecasting the throughput in a future window based on RSRQ, CQI, SINR, and past throughput measurements. The study utilizes random forest, Support Vector Machine (SVM), and Neural Networks (NN) as ML model types. Similarly to [10], the

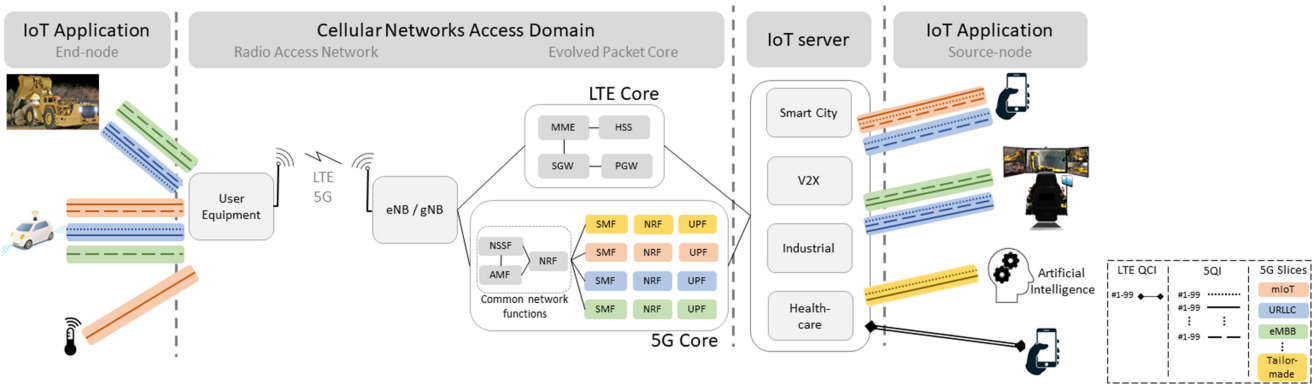


Fig. 2. Cellular network infrastructure supporting network slicing. The LTE network core, supporting QCI classes, includes serving gateway (SGW), packet data gateway (PGW), mobility management entity (MME) and home subscriber server (HSS), to illustrate the backward compatibility with 5G. While the 5G core architecture supports pre-configured and tailor-made slices for a specific application, with one or multiple 5QI [4]. The common network functions for each slice are access and mobility functions (AMF), network slice selection (NSSF), and network repository (NRF), while each slice has own configuration of the session management (SMF), NRF, and user plane (UPF).

study requires the UE to be in *connected mode* in order to perform the forecasting. The authors discuss a real-life deployment, where the ML model would forecast the application throughput while the end-user watches an online video. Raca *et al.* [11] analyze the size of the historical and forecasting windows, and their impact on the results. Their main approach for training consists of using percentiles of the raw features, particularly training on the 90th percentile of the data-set. Such an approach can be dangerous in cellular networks as it additionally excludes unique conditions that cannot merely be classified as outliers. Thus, the prediction accuracy is high, as in [11], but it can also lead into biased and overfitted ML models.

With our study, we take a step further from the state of the art in the following directions:

- Expand the use-cases by collecting real-life data in driving (urban, sub-urban, rural, underground) and crowded areas (stadiums and subways), with UL and DL traffic, while also experiencing CA, 2x2/4x4 MIMO, natural cell loads, and handovers.
- Statistical analysis and detailed feature engineering.
- Develop ML models for predicting the throughput in LTE, relying on non-intrusive features (Table 1).
- Benchmark the network slice performance in 5G non-standalone network with respect to throughput.
- Discuss a real-life deployment in a vehicular case.

2.3 Network Slicing in 5G

Heterogeneous services are allowed to coexist within the same 5G network architecture through network slicing [14]. Software Defined Networking (SDN) is a key technology enabling network slicing, where each slice gets a policy that governs the physical resource utilization [27]. A high-level view of the 5G architecture is shown on Fig. 2, consisting of UEs, access networks, core networks, and services.

Network slicing enables the cellular operators to slice the resource grid by reserving resources in time and frequency domain. Fig. 1b depicts an example of slice coexistence in the 5G resource grid. Therein, the eMBB slice spans in the time domain, while URLLC in the frequency domain. The anticipated large number of IoT devices and their intermittent traffic makes it unfeasible to allocate apriori mIoT resources [14]. Thus, a small portion of the resources is to be shared through random access [14]. Moreover, Fig. 1b illustrates the new way UEs will compete for the PRBs, through non-orthogonal multiple access (NOMA) schemas [28]. Therein, UEs with eMBB slice can compete for the same resources allocated for URLLC and mIoT slices, leading to higher spectral efficiency [14]. However, as pointed out by Popovski *et al.* [14], the mutual interferences in NOMA among eMBB, URLLC, and mIoT may significantly degrade the performance for all involved parties. Thus, we study the QoS monitoring methods, especially in a dynamic wireless environment where UEs already suffer from inherited issues, such as radio coverage, doppler effect, and scheduling [15]. Further, 3GPP changed the pattern of the RSs transmission compared to LTE, where, as shown on Fig. 1b, will not describe the whole bandwidth, but rather be UE specific and sent via the Master Information Block (MIB) [29]. For instance, one cell may assign eMBB UEs a 100 MHz bandwidth, while only 2 MHz for the MIoT UEs. Herein, an eMBB UE will receive RSs representative for the 100 MHz bandwidth, while a MIoT for the 2 MHz bandwidth. Hence, the reviewed state of the art in Section 2.2 [10], [11], [26] need to be revised for 5G applicability due to the modified nature of the RSs.

3GPP defines the life-cycle management of network slices, which includes admission control and configuration management [4], shown on Fig. 2. Regarding admission, Network Slice Broker (NSB) is proposed to perform on-demand resource negotiation, allocation, and charging, by considering the state of the network and signed SLA [3]. A single UE may be served by multiple network slices simultaneously, reported to the gNB via the Network Slice Selection Assistance Information (NSSAI) [4]. According to the SLA, the configuration of the NSSAI may have a default value or be dynamically modified by the service owner and network operator [16]. Further, the service owners and network operators may, in the future, design their own network slices [30] (Fig. 2). As a result, the NSB will have to monitor the network continuously and forecast the upcoming traffic to successfully reserve resources and ensure their availability in fulfilling the slice requirements [3]. However,

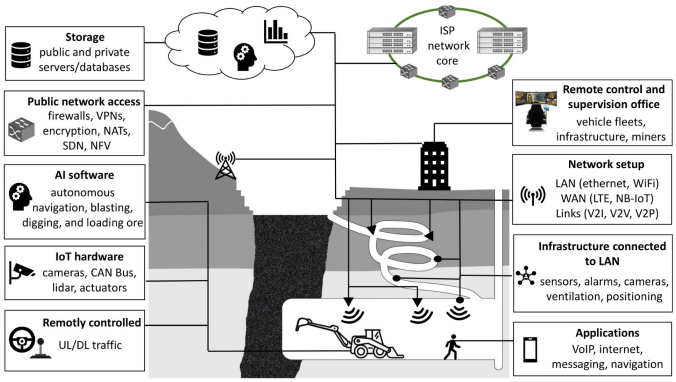


Fig. 3. Network infrastructure in the mine [9]. Network communication is required during self-driving and remote-controlling, but also for collecting sensory data, VoIP, internet, and data storage/retrieval.

the network monitoring and forecasting methods are susceptible to errors and biases [25], which is crucial in validating the SLAs for private customers paying for network slices.

Besides an NSSAI, the network also sets QoS classes to network slices. This approach is inherited from LTE, where the network can assign multiple QoS Class Identifier (QCI) to a UE by means of EPS Bearer [17]. In 5G, this parameter is known as 5G QoS Identifier (5QI) [4]. Similarly to LTE's QCI, the role of 5QI is to standardize a set of QoS rules for specific traffic, such as scheduling weights, admission thresholds, queue management thresholds, and link layer protocol configuration [4]. A 5QI value defines priority level, delay budget, PER, and achievable throughput [4]. In practice, as shown on Fig. 2, each network slice may have multiple 5QI, accommodating different traffic requirements in the SLA. In our study, QCI and 5QI are the least common denominator when transitioning from LTE to 5G network slice benchmarking. The 5QI and NSSAI values are the key features in a supervised ML to give an accurate prediction of the throughput per slice. That is, the model will learn the throughput distribution based on the 5QI and NSSAI values. Thus, our study first attempts to build a model utilizing QCI in LTE for throughput predictions per QoS class. Second, the model will then be applied in the 5G network via 5QI and NSSAI values for network slice benchmarking.

3 IoT Slicing Use-Case and Requirements

The imperfections of the sensor technology within autonomous vehicles lead the research community into developing advanced inter-network communication methods to improve the service reliability [31]. The idea is to enable vehicles to become IoT devices and communicate among themselves. Hence, sharing information for predicting events before they occur [32]. Standardized communication approaches by 3GPP appear as suitable option due to infrastructure reusability, offering wide coverage, QoS, and high-speed movement support [31]. An LTE-Vehicle (LTE-V) standard was issued with 3GPP Release 15, while the new 5G-V2X standard takes part in Release 16 [33]. Fig. 3 shows the IoT use-case in our study, involving autonomous mining vehicles that can be remotely-controlled. The V2X interfaces, depicted on Table 2, are defined by 3GPP as: (i) V2I, where a vehicle communicates via the existing network infrastructure, such as base stations; (ii) V2V, covering the direct communication among the vehicles; and (iii) V2P, vehicles communicating with the miners. The mining IoT setup consists of vehicles with a variety of sensors, actuators, and cameras, streaming V2I data in UL to a remote-control station (Fig. 3). An expert driver can supervise and control vehicles from a distance, with DL data from joysticks, buttons, and pedals. Generally, the V2X communication shown on Fig. 3 consists of critical and non-critical packets, with substantially and lightly weighted data. Hence, the network requirements on Table 2 are rather dynamic and should be validated per application [3].

TABLE 2
Network Slice Requirements for the Mining Use-Case

V2X links	IoT applications	Slice requirements		
		Thrput.	Latency	PER
Vehicle-to-Infrastructure (V2I)	Video Live Stream	+1 Gbps	150 ms	10^{-3}
	Software Updates	+1 Gbps	300 ms	10^{-3}
	Advanced driving [33]	5 Mbps	60 ms	10^{-5}
	Sensor Stream	1 Mbps	200 ms	10^{-5}
	Remote control stream	1 Mbps	5 ms	10^{-6}
	Infotainment	+1 Gbps	300 ms	10^{-3}
Vehicle-to-vehicle (V2V)	Platooning	1 Mbps	5 ms	10^{-6}
	CAMs data sharing [33]	1 Mbps	60 ms	10^{-5}
	DENMs data sharing [33]	1 Mbps	200 ms	10^{-4}
Vehicle-to-people (V2P)	Positioning	1 Mbps	200 ms	10^{-4}
	Coordination	1 Mbps	200 ms	10^{-4}
	Warnings	1 Mbps	60 ms	10^{-6}
	Emergency	5 Mbps	5 ms	10^{-6}

The V2X requirements from the stakeholder, for our use-case, match the network slicing requirements defined by 3GPP [4] and are illustrated on Table 2. For instance, the sensor stream consists of vehicle data shared on the network (e.g., lidar, radar, engine control unit, transmission, fuel, and battery levels), with requirements of 200ms latency, 1 Mbps, and 10^{-5} PER. While the remote control stream, coming from the expert-driver, has stringent requirements regarding latency and PER, 5ms and 10^{-6}, respectively [4]. The advanced driving includes coordination, safety, emergency data, and heavier information, such as trajectory mappings. Hence, it requires 5 Mbps, 60ms latency, and Packet Error Rate (PER) of 10^{-5} [4]. The abundance of traffic streams with unique requirements (Table 2) pose a heavy burden on cellular networks to deliver high QoS [32]. We envision a scenario where the stakeholder will have multiple network slices per vehicle, vertically slicing the IoT applications from Table 2. In fact, there are multiple ways of assigning a network slice to IoT devices [3]:

- UEs are pre-configured with a static NSSAI, representing the slice requirements.
- Both the network operator and service owner may dynamically change an NSSAI per device, as well as changing the resources assigned to a particular NSSAI.
- NSSAIs are assigned according to a destination/source address of each packet sequence.

Benchmarking the network slice performance goes inline with delivering high QoIoT in the mining case-study [9]. Real-time prediction of the available throughput per

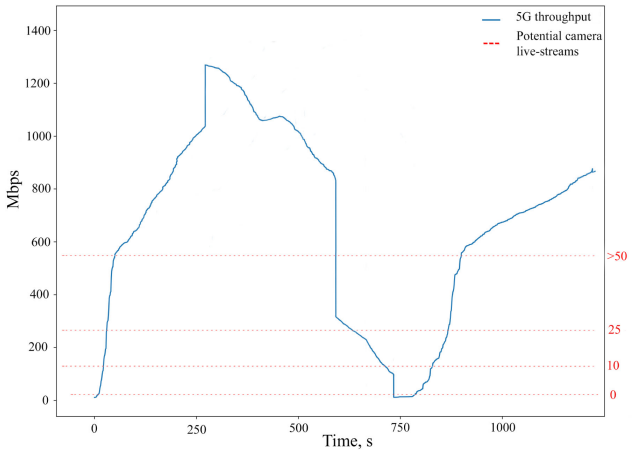


Fig. 4. Measured throughput on 5G mine's network. The potential cameras supported by the network is extracted from the stakeholder's requirements (Table 2). At last 10 vehicles should be supported on the network, ideally with 5 live-cameras per vehicle, along the other applications.

TABLE 3
Statistics of the Conducted Experiments

Use-cases	Expr	RSSI	RSRP	RSRQ	SINR	RF-NC	RI sum	p_a (CQI in 5G)	TA (CRI in 5G)	Shannon (kbps)	Cell Load	True Thrput. (kbps)
City drive	mean	-55.5	-83.9	-7.6	15.9	-107.3	123.9	0.53	1.5	52,926	0.90	71,851
	std	9.4	9.6	2.0	7.0	13.0	36.7	0.49	12.1	26,958	0.14	27,176
Subway ride	mean	-77.2	-103.7	-5.6	15.6	110.8	100.1	0.17	0.5	52,464	0.83	45,384
	std	6.0	6.9	1.9	6.1	5.3	2.5	0.37	1.8	19,273	0.2	12,788
Rural drive	mean	-85.1	-111.6	-9.0	7.3	-116.6	100	0.0	7.1	10,926	0.9	7,970
	std	2.0	1.5	0.9	1.4	3.0	0.0	0.0	0.3	2,070	0.08	5,122
Sub-urban drive	mean	-56.4	-85.4	-8.3	17.0	-96.6	168.6	0.01	0.4	70,327	0.84	76,280
	std	5.1	5.1	1.1	4.2	4.9	29.8	0.11	7.1	24,265	0.1	18,494
Office /home	mean	-61.5	-87.1	-4.7	27.0	-123.4	100	0.08	0.1	66,063	0.84	67,335
	std	10.0	10.2	1.8	5.8	17.8	0.0	0.27	0.04	15,287	0.17	10,680
Crowd areas	mean	-39.1	-66.6	-6.8	26.9	-115.4	151.6	0.0	21.5	102,534	0.92	130,674
	std	4.0	4.3	1.2	3.0	27.55	44.7	0.0	45.5	36,766	0.13	16,763
Mining	mean	-54.8	-81.3	-5.9	26.8	-123.4	107.9	0.2	0.1	70,763	0.98	48,995
	std	13.2	13.0	0.7	5.1	19.1	20.6	0.4	0.08	16,383	0.02	26,643
Total LTE	hours	**93**	**93**	**93**	**93**	**93**	**93**	**93**	**93**	**93**	**93**	**93**
	mean	**-62.9**	**-90.5**	**-7.4**	**22.1**	**-118.2**	**145.8**	**0.1**	**13.3**	**78,089**	**0.88**	**98,659**
	std	**7.4**	**7.4**	**1.5**	**4.6**	**14.7**	**43.2**	**0.3**	**33.7**	**36,947**	**0.15**	**42,856**
No-CA (LTE)	hours	64	64	64	64	64	64	64	64	64	64	64
	mean	-52.8	-81	-7.5	20.1	-115.7	149	0.1	6.4	75,307	0.87	88,146
	std	11.5	11.7	1.7	6.8	18.6	41.3	0.3	26.5	33,566	0.13	33,500
CA (LTE)	hours	29	29	29	29	29	29	29	29	29	29	29
	mean	-73.1	-100	-7.4	24.1	-120.8	142.7	0.1	20.3	80,872	0.90	109,172
	std	3.4	3.1	1.3	2.5	10.9	45.2	0.2	40.9	40,328	0.17	52,212
Total 5G	hours	*NaN*	**23.4**	**23.4**	**23.4**	*NaN*	**23.4**	***23.4***	***23.4***	*NaN*	*NaN*	**23.4**
	mean	*NaN*	**-85.2**	**-10.8**	**21.1**	*NaN*	**3.4**	***12.9***	***3.1***	*NaN*	*NaN*	**802,763**
	std	*NaN*	**8.1**	**4.2**	**3.4**	*NaN*	**1.1**	***3.5***	***1.7***	*NaN*	*NaN*	**500,509**

network slice can directly impact some of the QoIoT metrics, such as productivity, safety, and reliability [9]. Fig. 4 shows a snippet of the 5G throughput experienced in the underground mine. The red-dashed lines illustrate the number of potential live-streaming cameras supported on the network due to the quality requirements on Table 2. Further, the requirement is to have at least five vehicles operating simultaneously, while one vehicle has five 1080p cameras live-streaming. On Fig. 4, towards the middle, the vehicle goes out of coverage and slowly loses connectivity to a point when video can no longer be live-streamed. During such periods, the mining company would experience a shutdown of the vehicle, impacting the overall productivity. Benchmarking the throughput can be used for root-cause analysis and perform recoveries. For instance, a predicted low throughput can warn the expert driver when approaching a blind spot, directly lower resolution or switch off multiple cameras.

4 Experimental Setup

Within this section, we provide a high-level description of the conducted experiments. In addition, we introduce the utilized software to perform the experiments and collect the data-sets.

4.1 Experimental Description

Fig. 2 depicts the high-level cellular infrastructure used to conduct the experiments. A mobile phone resides as a source node, naturally communicating via a commercial cellular networks (LTE and non-standalone 5G). The first objective of the experiments is to measure the maximum available throughput via active testing. Section 4.2 describes the software and servers utilized for active testing. The second objective is to collect the features from Table 1 using a separate software, a process described in Section 4.3.

The underground mine has two pre-installed cellular networks, LTE and non-standalone 5G. Compared to commercial, above-earth, cellular networks, the mining wireless environment has vastly different properties. For instance, there is different signal propagation pattern due to the wet environment, minimal noises and competing frequencies. The network load and resource utilization are also considerably lower and more stable, discussed in Section 6.2. However, the mining vehicles drive in-and-out of the mine and experience the two different radio environments. Thus, we extend the scope of the experiments also to cover above-earth wireless conditions. Along with the mine, we conduct experiments in other use-cases, depicted on Table 3. The following list summarizes the environment of the conducted experiments:

- Drive tests on LTE networks in urban cities, sub-urban, and rural areas, with an average vehicle speed of 50, 80, and 60 km/h, respectively. The urban city experiments include individual vehicle rides and public transportation (i.e., subway and bus) in three cities, Stockholm, Luleå, and Skellefteå (Sweden). The sub-urban city includes motorway and minor road rides around the mentioned cities.
- Static and walking tests in environments such as office and home in Stockholm and Luleå (Sweden). These tests include LTE and 5G networks.
- Static and walking tests in crowded areas on the LTE network, with around 5000 people (live hockey match) and a large shopping mall in Stockholm (Sweden).
- Static and drive tests in underground mine, on LTE and 5G networks, with average drive speed of 20 km/h.

The LTE network experiments were performed simultaneously with two phones, each with different SIM card from two Swedish cellular operators that have own radio networks, without any speed and data restrictions. The brand of the utilized mobile phones varied per test in order not to be phone-bias as much as possible. Thus, the following phones were used per use-case: Sony XZ (City, Mining, Rural, Subway), Samsung S10 (Mining, Office, Crowd), Huawei Mate 30 Pro (Rural, Crowd, Sub-urban, Subway), and Samsung S20 (City, Mining, Office, Sub-urban). The core networks of the two operators resided in Stockholm. The experiments experience the natural behavior of the cells

during real-life usage. Thus, natural cell load and handovers. The following cells' settings were observed during the experiments: 10 and 20 MHz bandwidth, 2x2 and 4x4 MIMO, appearance of CA, one static QCI value (assigned by the operators), and frequency bands such as 3, 7, 8, and 20. Section 5 covers a more detailed analysis of the statistics of the wireless conditions.

The non-standalone 5G network, owned by one Swedish cellular operator, is installed in the mining environment, as well as in an indoor and outdoor area of a university, with a core network in Stockholm. A special SIM card was required to perform the experiments, without any speed and data restrictions. Note that Huawei Mate 30 Pro was the only phone capable of doing the 5G non-standalone tests. Both 5G non-standalone network deployments consist of multiple cell nodes, configured on 3.5 GHz frequency with 100MHz bandwidth, 30 kHz sub-carrier spacing (SCS), and only one available beam and slice.

The goal of conducting experiments in such different use-cases is three folded. First, the ML model will learn the impact of different RF conditions (noise, interference, coverage, and speed) on the throughput. Second, the ML model will understand how congested cells and network load impact the throughput. For instance, minimal competing traffic in the mining network against crowded areas. Third, the ML model will capture the impact of novel network technologies, such as CA and MIMO, on the achievable throughput.

4.2 Generating Cellular Traffic

The active testing has a goal of creating the labeled data for the ML model. Several software tools, such as iperf and netperf, can periodically stream a bulk of packets to achieve the maximum available throughput. However, for this study, we develop our own software tool for generating cellular traffic. The main motivation is the ability to fully control the network traffic, packet handling at the server-side, and servers' location. For instance, a service that requires multiple slices will have unique packet sizes and traffic patterns for each slice, with different time gaps between the packets [3]. In addition, the use of multiple server locations will eliminate the biases to a particular public internet structure. As a result, a software tool was developed based on the TWAMP library, an RFC standard with open implementation [34], capable of streaming continuous short bursts of packet trains. TWAMP is also used by Ericsson and Rohde&Schwarz to perform active testing [35], [36]. In our case, TWAMP streams UDP traffic interchangeably in UL and DL. The aim is to maximize the resources and push as many packets as possible to the cell. Hence, the client continuously adapts the packet trains' structure based on the response from the server, allowing asymmetric traffic. For instance, the client/server can modify the packet size, the time intervals between packet trains, and the number of packets per train in order to adapt to current conditions. The idea is to most effectively achieve maximum UL and DL throughput under diverse network conditions. The client part was deployed on multiple phones brands, as mentioned in Section 4.1, where two phones simultaneously did the active tests with SIMs from two different operators. The destination servers resided in Stockholm, Frankfurt, and Dublin, which we observed to have no significant impact on the measured throughput.

The measured throughput range from 1 Mbps to 150 Mbps in LTE and from 50 Mbps to 1.4 Gbps in 5G (Table 3). The packet trains' size is 1500ms for both UL and DL, with a time gap between the packet trains of 500ms [34]. Typically, a cell gradually assigns and removes resources to an end-node; thus, we observe 1500ms to be enough per packet train in achieving the maximum available throughput. From there, the labeled throughput value per packet train is the peak measured value during 1500ms, as observed on Fig. 5. The 500ms waiting time between packet trains is recommended by the TWAMP library as the optimal time for the server to wait for reconstruction of missing packets after a packet train [34]. Our tool was previously used in [37], where 15 phones simultaneously generated congestion on a commercial LTE network. The phones were able to generate network load, in a controlled manner, in ranges of $< 5\%$, 10-20, 35-50, 50-65, and 75-95 percent. The load was validated by the cell's logs from the network operator.

4.3 TEMS Pocket

A mobile application, TEMS Pocket, was used to record the features [38], running in parallel with the TWAMP software. TEMS Pocket is a commercial state-of-the-art phone-based tool, which monitors the performance of wireless networks on a millisecond level basis. The parameters of interest within this study, shown in Table 1, are monitored by scanning the RF and signals from the surrounding cells. The main motivation behind selecting the listed features is their *passive* nature. Namely, the commercial phones registered on the cellular network can measure those features while in *idle* mode, without generating any network intrusiveness. Hence, Table 1 features were the only metrics available from TEMS Pocket that fit the study's scope. The listed features can also be measured with other competing software, such as Qualcomm QXDM.[1] The reviewed state of the art in Section 2.2 only considers a handful of the parameters from Table 1. For instance, [11] considers only RSRQ, SINR, and CQI, while [10] considers RSRP, RSRQ, and CQI. In that way, we extend the state of the art in bringing up an additional set of features and study their influence on the throughput. Section 5.1 describes feature engineering during *idle mode* while performing active testing.

5 Data Analysis and Feature Engineering

As mentioned in Section 4, the active tests are performed with time intervals of 1500ms with 500ms silence in between, interchangeably in UL and DL. Fig. 5 shows the typically observed throughput pattern during the active tests. The DL/UL starts slowly, with a couple of hundreds of milliseconds required before it reaches the peak value. Thus, the labeled throughput value is the peak value for each packet train. In this section, we further discuss feature engineering, as well as perform statistical and correlation analysis on the recorded data-sets.

1. Available at www.qxdm-professional.software.informer.com/3.2/. Accessed Sept. 2020.

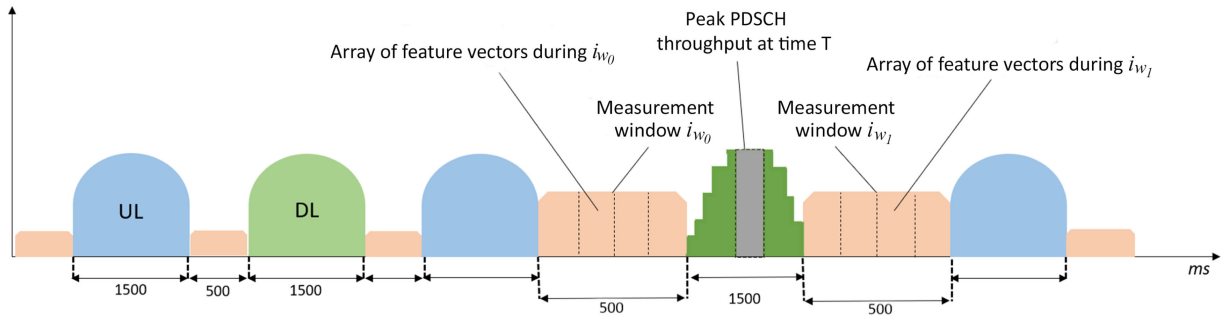


Fig. 5. Feature engineering. Creating two measurement windows for each peak throughput value, where each window has an array of feature vectors. The two windows represent the prior and future RF conditions, with respect to the peak throughput at time T.

5.1 Creating Feature Vectors

Due to the millisecond level time-resolution of TEMS Pocket, the features have to be aggregated for mapping to the labeled value. The RF measurements (Table 1) suffer from deterioration during active tests, especially when sending bulk of UL/DL data traffic [21]. This is because the serving cell in *connected mode* utilizes more resources and, thus, receives multiple fluctuating values of, for instance, RSRP and RSRQ [21]. Hence, the RF measurements have different patterns in *connected* and *idle* mode. Fig. 5 shows the creation of feature vectors, which capture the radio conditions during the silent period between the UL/DL data. The goal is to collect RF measurements during *idle* mode and train them to predict their corresponding peak throughput value. This is an extension of the state of the art, since [10], [11], [12], [26] do not provide information on feature engineering, considering the difference among the RF features in *idle* and *connected mode*.

The feature vectors are defined as

$$[i_{T-w_0}, j_{x,t}]_z$$

$$[i_{T+w_1}, j_{x,t}]_z,$$

where the measurement windows i spread before and after the time T of the measured peak throughput (Fig. 5). The initial length of the measurement windows i_{w_0} and i_{w_1} is 500ms each. While $j_{x,t}$ denotes the pool of features from Table 1, where x iterates the features at time t within the window size. The value z, where $z \in \{1, 2, \ldots, n\}$, creates an array of feature vectors within the measurement windows i_{w_0} and i_{w_1}, respectively. The idea is to have feature vectors, per measurement window, that are closest and furthest to the peak throughput, as depicted on Fig. 5. The number of feature vectors per measurement window, i.e., z, is chosen empirically, depending on the quality of the radio conditions during measurement. That is, in certain cases there can be hundreds of values per feature x during 500ms time window, while at other times only a few tens of values would be observed as the phone receives less radio measurements in poor radio conditions. Via trial-and-error we have tested a range of z values from 1 to 10, where the achieved overall accuracy (R^2) peaks at the value $z = 3$ from where it gradually decreases. Thus, we have 3 vectors per measurement window i_{w_0} and i_{w_1}, respectively, where each vector aggregates the mean value of each feature x. An example of a feature vector, with real values, is provided on Table 4 containing the recorded features before and after the measured throughput. Note that only a subset of the features is shown on Table 4 from the whole set as represented on Table 1.

To illustrate the training process, each feature vector z is trained against its corresponding peak throughput value. Then, we combine and average the predicted throughput values to produce one final prediction per packet train. An algorithmic representation of this process is discussed in Section 6. Note that the complexity of the model increases as the z values gets lager. An array of feature vectors per $[i_{T-w_0}, j_{k,t}]_z$ and $[i_{T+w_1}, j_{k,t}]_z$, each predicting a throughput value per packet train, gave on average a 4 percent increase ($z = 3$) in accuracy compared to having one aggregated feature vector ($z = 1$). One reason for this could be that the relationships in the latter case are destroyed when there are non-linear behaviors of the features for specific time-frames. Also, feature vectors recorded closest to the peak have a more significant influence on the throughput than furthest vectors.

5.2 Feature Extraction

The use of RF measurements as features (Table 1) is rather universal, as discussed in Section 2. For instance, the legacy Shannon formula (1) utilizes RF parameters, such as RSSI, RSRP, and RSRQ, to form SINR value that can calculate the communication channel capacity. Making a feature from (1) is beneficial since the ML model will eventually learn the dependencies and limitations present in the Shannon formula. For instance, the impact of CA on the throughput. The two compensating factors in (1), α and β, handle the average behaviour of an LTE cell, but that is obviously not as accurate as to dynamically handle specific events, such as RS overhead loss and available cell resources [21]. Complementary to (1), Chang *et al.* [21] proposes serving cell load measurement based on RF measurements

$$CellLoad = 1 - \frac{5 - \rho_b}{6 - min(2, NumAntennas)} \times \left(\frac{1}{8 \times n \times RSRQ} - \frac{3}{2 \times n \times SINR} - \frac{1}{4}\right). \quad (2)$$

Although the authors did not thoroughly validate (2), we will utilize the formula as a feature in our study, since we observe that it catches the right trends. The idea is to enable the ML model to understand the impact of the cell load on

TABLE 4
An Example of One Feature Vector With Respect to Labeled Throughput During the City Drive Tests, Where the Features are Measured Before and After the Label so That the ID 3 and 4 are Closest to the Label While 1 and 6 Furthest, Respectively

ID	PeakThrp-SeqNum	Before-After	Throughput (Kbps)	RSSI	RSRP	SINR
1	1	0	82836	-62.25	-89.11	21.3
2	1	0	82836	-63.59	-90.78	20.6
3	1	0	82836	-63.90	-90.43	17.7
4	1	1	82836	-64.68	-92.68	15.8
5	1	1	82836	-64.68	-92.68	15.8
6	1	1	82836	-65.93	-92.68	19.8

the throughput. The expectation is that (2) can be used in conjunction with (1) in order to more accurately predict the throughput.

5.3 Correlation Analysis and Understanding the Data

Fig. 6 depicts Spearman correlation graph among the features and label. Such a correlation does not assume that data is from a specific distribution, so it is a non-parametric correlation measure. At first glance, we can remove the feature representing QCI (and 5QI as a 5G equivalent), since only one value was available. RS path loss is a function of RSSI and RSRP and, thus, can be removed as it is almost entirely correlated, with 0.95 and 0.98 (Fig. 6), to those features. RF parameters were also aggregated for all four transmission antennas separately, thus the features from RF-RX0 to RF-RX3. However, it seems that such information does not carry any significant information, as it is highly correlated to SINR. Arguably the most surprising results from Fig. 6 are the obtained correlation for Shannon and Cell load features. One would expect a high correlation with the RF values, as (1) and (2) are calculated based on the RF values. However, the static parameters in each of the formulas seem to play a large role in extracting the calculated values from the RF data.

Table 3 illustrates the statistics of the data-sets, showing the experienced wireless conditions per use-cases in terms of mean and standard deviation (std). The highest fluctuation values from the mean, regarding the labeled throughput, were recorded during the city drive and mine tests, with *std* of around 27 Mbps and mean of 71 and 48 Mbps, respectively. Interesting point is that within those two use-cases the RF values also show the highest *std* compared to the other use-cases. However, although the *std* of the labeled throughput is similar for the two mentioned cases, the RF values in the mining test observe noticeably higher *std* compared to the city drive. Higher *std* is due to the perfect radio conditions when the vehicle is right below a cell in the mine and being relatively easy to drive out of coverage, which rarely happens in the city environment. On another note, the mine tests also experience the lowest cell load of around 0.2 percent, which was expected since we were mostly alone on the network. Hence, the ML model can understand the

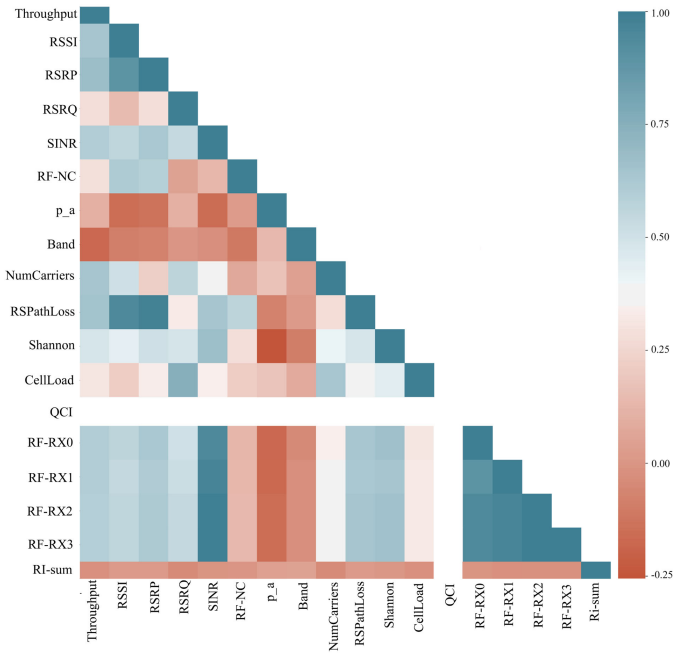


Fig. 6. Spearman correlation graph of the features recorded on the LTE network.

way cell load impacts the throughput. The highest cell load was experienced during the subway ride tests, where the operators may focus more on increasing the coverage in the underground tunnels where the trade-off is being loaded cells.

Algorithm 1. Algorithm for Throughput Prediction

Input: Feature set X; Label set Y; Time T of a measured label; Measurement window i, with two sizes w_0 and w_1, respectively; and $model$ is a selected ML model.
Output: FV_1 as feature vector for window i_{w_0}, FV_2 as feature vector for window i_{w_1}, Predicted label $\hat{Y}$.

for each Y **do**
  for $(i = T - 1; i > w_0; i - -)$ **do**
    $FV_1 \leftarrow aggregate(Xi)$
  for $(i = T + 1; i < w_1; i + +)$ **do**
    $FV_2 \leftarrow aggregate(Xi)$
function MLmodel ▷ Select ML model
function Rand_Split_80_20$[FV_1, FV_2], Y$
function ML.train$[FV_{train1}, FV_{train2}], Y_{train}$
function $\hat{Y} \leftarrow$ ML.test$[FV_{test1}, FV_{test2}], Y_{test}$
return $\hat{Y}$

Another takeaway from the Table 3 is the performance of the (1) formula. As expected, the calculated throughput is the closest to the true recorded throughput during static tests, where the RF fluctuate the least. However, (1) largely over-performs in cases of low throughout values, such as rural and mining tests, while under-performing in areas with high throughputs, such as crowded and city drive tests. Overall, (1) achieved 67 percent of accuracy compared to the measured throughput value. One reason for under-performing is because (1) does not capture the effect of CA, MIMO, and novel spatial multiplexing schemes.

Finally, crowded areas are a special case as they observe, on average, the best RF values, and hence the largest

throughput performance. The reason is that the operators heavily invest in cellular equipment to prepare the cells to handle the load and offer multiple carriers for the traffic.

6 Machine Learning Modeling and Results

In this section, we develop ML models using supervised learning for predicting the maximum available throughput per UE. Specifically, we perform two concrete regression tasks, first to utilize the features from Table 1 to directly predict LTE throughput, and second, to predict throughput in 5G settings, per network slice. Herein, we motivate the selection of the ML models, coupled with reasoning on their hyperparameters optimization. Then, we provide the results of the two regression tasks and discuss the outcomes. Algorithm 1 summarizes the proposed approach.

6.1 Models Types and Metrics

Regarding ML model types, we select multiple approaches typically recommended for tabular regression tasks [25]. For instance, models using decision trees (DTs) are beneficial for understanding the decision process for root cause analysis. Thus, we select two different DT approaches, random forest and XGBoost. As the most widely used DT approach, the first one averages the result over a series of independent DTs. While the latter arguably improves the random forest's method by implementing an extreme gradient boosting. XGBoost creates sequential trees that try to reduce the error term and search for the minima of the cost function [39].

Deep Learning (DL) methods, specifically neural networks (NN), appear to be suitable for large data-sets and regressions tasks [40]. From the family of the NN techniques, we select Artificial NN (ANN) as a preferable option for handling tabular data-sets, specifically an Multilayer perceptron (MLP). The other DL techniques, Convolutional and Recurrent NN, are typically recommended for image and time-series data, respectively [40]. Finally, we select Support Vector Regression (SVR), as it has been shown to achieve better results when predicting edge cases for some tabular data-sets [41]. SVR is different from the mentioned models, where instead of minimizing the error with each iteration, SVR defines an acceptable error and uses it to find an appropriate hyperlane to fit the data, avoiding falling into local optima [41].

Each of the selected models has a unique set of hyperparameters that can be tuned-in to achieve higher accuracy and stop possible over/underfit. In the following, we show the set of selected hyper-parameter for each model, for which we used a grid search with 10-fold cross-validation (CV) to empirically find the best suited values. In LTE and 5G cases, the CV is autonomously performing the data split, instructed to divide it as 70 percent train, 10 percent validation, and 20 percent test data. As suggested by Russell and Norvig [42], the hyper-parameter optimization is performed on the validation set. For evaluation, we use two common statistical metrics, R^2 and MSE. In the following, we present the selected hyper-parameters for each ML model for LTE and 5G (values in brackets), since the models are trained separately:

- *RF^2 = Estimators 70 (75), MaxDepth 15 (16), MinSampleSplit 0.7 (0.9), MinSamplesLeaf 0.5 (0.4), MaxFeatures 0.5 (0.5)*
- *$XGBoost^3$ = Estimators 100 (100), MaxDepth 19 (20), Alpha 1.2 (1.2), Lambda 1.2 (1.2), SubSample 0.8 (0.5), MinChildWeight 3 (5), ColsampleBytree 0.9 (0.6), Eta 0.1 (0.1)*
- *SVR^2 = Kernel rbg (rbf), Gamma 0.6 (0.5), C 1 (1), Epsilon 0.1 (0.1)*
- *Multilayer perceptron $(MLP)^2$ = HiddenLayerSize 11 (20), Activation relu (relu), Solver sgd (sgd), Alpha 0.0006 (0.0009)*

6.2 Prediction Results in LTE

Table 5 details the prediction results for each model type. It is important to note that the evaluation of *Total LTE* combines the data-sets from each of the uses-cases, where CA is both active and inactive. Regarding *Total LTE*, XGBoost achieves the highest R^2 of 93 percent and lowest MSE of 0.06, followed by MLP with 92 percent R^2 value and 0.08 MSE. Note that the results are obtained using a 10-fold CV, with 70 percent train, 10 percent validation, and 20 percent test data, as suggested by Russell and Norvig in order to eliminate potential issues regarding peeking and data leakage [42]. Further, in Section 6.2.3 we justify the selection of 70 percent train data through a gradual increase of the training set, where the overall accuracy saturates after the 70 percent of the training set size. To better understand the results regarding *Total LTE*, it is necessary to have a closer look at the results per use-case. In the following, we validate the accuracy of throughput predictions per use-case.

6.2.1 Prediction Results per Use-Case

Table 5 details the evaluation results for each of the use-cases separately. Note that we use a modified CV in order to produce the per use-case results. First, we randomly select 50 percent test data from a particular use-case. Second, we combine the other 50 percent with the rest of use-cases to create the train data, from were 80 percent is randomly chosen to train the ML model. The training process is repeated 10 times to act as a CV.

Such throughput prediction, per use-case in isolation, shows the differences in evaluation accuracy from case-to-case when using one general model for training. Complementary to Table 5, Fig. 7 extracts snippets of the recorded and predicted throughput in isolation per use-case. In the following, we discuss the individual results, per use-case, in more detail.

City and Sub-Urban Driving. The results from Fig. 7 and Table 5 regarding urban and sub-urban (Stockholm) and small city (Skellefteå) drive are rather similar in accuracy, in general for each ML model. The reasoning is that even though the driving conditions are different in terms of speed, traffic jams, and buildings, the radio conditions from Table 3 are almost identical. Moreover, the mean recorded throughput for these use-case has a similar mean value of around 76 Mbps, hence similar prediction results. The

2. Parameter description at https://scikit-learn.org
3. Parameter description at https://xgboost.readthedocs.io

TABLE 5
Prediction Results and Feature Importance

	Model	City drive	Subway ride	Rural drive	Sub urban drive	Home/office	Crowd areas	Mining	Total LTE	Non-CA	CA	5G Total
R^2	RF	0.91	0.90	0.88	0.91	0.96	0.89	0.90	0.91	0.94	0.87	0.82
	SVR	0.89	0.87	0.91	0.89	0.93	0.90	0.88	0.89	0.89	0.88	0.80
	XGBoost	0.94	0.90	0.92	0.95	0.96	0.91	0.95	**0.93**	**0.95**	0.90	0.83
	MLP	0.93	0.90	0.92	0.92	0.95	0.92	0.90	0.92	0.94	**0.90**	**0.84**
MSE	RF	0.1	0.1	0.13	0.11	0.07	0.12	0.1	0.1	0.07	0.13	0.19
	SVR	0.15	0.16	0.12	0.14	0.09	0.13	0.14	0.15	0.13	0.15	0.20
	XGBoost	0.07	0.1	0.08	0.05	0.02	0.09	0.04	**0.06**	**0.06**	**0.08**	0.17
	MLP	0.07	0.11	0.09	0.08	0.04	0.06	0.1	0.08	0.06	0.09	**0.17**

		Feature Importance (XGBoost)									
		RSSI	RSRP	RSRQ	SINR	RF-NC	RI-sum	p_a (CQI in 5G)	TA (CRI in 5G)	Shannon formula	Cell Load estimate
Global	**Total LTE**	0.11	0.09	0.10	0.11	0.10	0.09	0.12	0.06	0.10	0.12
	Total 5G	*NaN*	0.16	0.17	0.20	*NaN*	0.18	0.15	0.13	*NaN*	*NaN*
Local	**Total LTE**	0.08	0.06	0.11	0.15	0.11	0.10	0.15	0.04	0.08	0.12
	Total 5G	*NaN*	0.12	0.10	0.31	*NaN*	0.21	0.18	0.08	*NaN*	*NaN*

relative high R^2 values (XGBoost) in the city and sub-urban driving conditions demonstrate the ability to capture the effects of the doppler effect and the randomness generated by various present issues, such as reflections from buildings, scattered signals, and interferences.

Subway Ride and Mining Tests. On average, those two cases experience the same recorded throughput, with a mean value of around 46 Mbps. Both use-cases are in an underground environment with a similar cell setup with directional antennas to cover more ground in the tunnels. However, as discussed in Section 5.3, the mining tests experience the largest *std* on average, due to the distinct radio conditions and the way cells are positioned (Table 3). The R^2 is lower for the subway case, compared to mining, due to several unique contextual events. Namely, the snippet of the subway ride on Fig. 7 shows a sudden increase in the recorded throughput, particularly when the subway reaches the subway station. In addition, there are accuracy glitches during periods when the subway suddenly meets with another subway in the tunnels or station (Fig. 7). The lower accuracy in such cases indicates that more training data is necessary to fully capture the impact of the RF features on the label.

Home and Office Tests. As expected, the static and walking tests from the home and office areas achieve the highest accuracy and lowest MSE on average for each ML model. The reasoning for this behavior is the relative constant RF condition, i.e., lowest *std*, and highest accuracy by the Shannon Formula (1). The mean value recorded by (1) is 66 Mbps, which is the closest to the real recorded throughput (mean of 67 Mbps) compared to the other use-cases.

Rural Driving and Crowded Areas. The hockey match's snippet on Fig. 7 is during the last 40 minutes of the match, showing results from two Swedish operators. Operator A has apparent dips in the recorded throughput. The first significant throughput dip, with recorded 60 Mbps, is during the break of 18 minutes - associated with increased traffic on the network, as people start to use their cell phones. The second dip of around 100 Mbps is when the match ended, and people were leaving the area, again associated with heavy network load. However, Operator A record higher throughput when the arena is empty (approx. 145 Mbps) compared to Operator B (approx 120 Mbps). Noticeably, the ML model under-performs during the highest recorded throughput for the Operator A and over-performs during the lowest recorded throughput for the Operator B. In general, the hockey match achieves the highest throughput, opposite of the rural driving tests, which record the lowest (Fig. 7). A common outcome for both rural and crowded area use-cases is the lowest R^2 accuracy and highest MSE, on average, compared to the rest of the cases (Fig. 5). Table 3 can explain such behavior, as the mentioned use-cases contain the most edge cases or outliers in the data-set. For instance, the crowded areas experience the best RF conditions and, hence, the highest throughput measurements, while the rural areas have the lowest recorded RF conditions and throughput (Table 3). To remedy this, one may argue that additional experiments are necessary for the two use-cases to strengthen the models' weights that correspond to the experienced RF conditions.

6.2.2 Prediction Results for Carrier Aggregation

Table 5 also extracts the LTE results based on whether CA was present or not. To produce these results, we split the data-set based on the *Number of Carriers* feature (Table 1). XGBoost and MLP produce the same accuracy of 90 percent; however, there is a noticeable 5 percent decrease in accuracy than predicting non-CA cases. The large difference in accuracy of CA and Non-CA is due to the uncontrollable nature of CA, as reported in [17]. Namely, the CA's occurrence and its impact in the total throughput are relative to the cell load and other UEs competing for the same carriers. Thus, the ML models struggle to fully capture the impact of the CA on throughput. Although the lowest overall scores, it is noticeable that SVR produces similar results when predicting CA and Non-CA cases.

Note that the feature *Number of Carriers* is measurable only in *connected mode*, i.e., when the UE is sending or

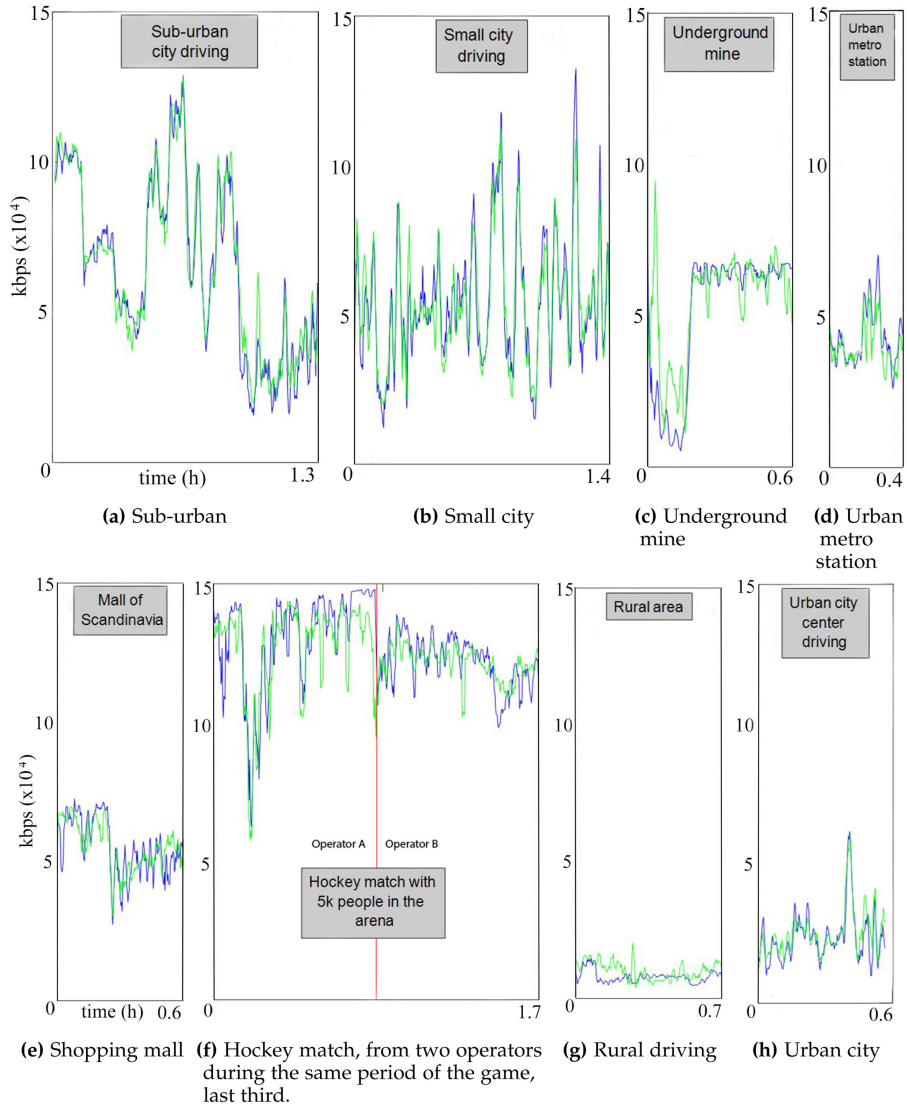


Fig. 7. A snippet of the recorded (blue) versus predicted (green) throughput, per use-case, using the XGBoost model on LTE data-sets.

receiving data. A UE in *idle mode* will not know whether CA is available at that particular point in time. Thus, there are three deployment options for the proposed ML models in covering CA on LTE network:

- *Intrusive mode*, where *Number of Carriers* is measurable when, for instance, a vehicle starts streaming its own application data.
- *Low-intrusive mode*, where, for instance, a vehicle is forced to stream a minimal size packet train, upon a cell change, in order to trigger a CA and measure the *Number of Carriers* feature.
- *Non-intrusive mode*, where the ML model does not consider the *Number of Carriers* feature, relying entirely on the rest of the features from Table 1. To prove that this theory works, we create a separate ML classification model to predict the occurrence of CA, described in the following.

The motivation for a classification model predicting *Number of Carriers* is to analyze whether the idle features from Table 1 could predict the CA's occurrence. Herein, the goal is to prove that the ML models predicting *Total LTE* throughput will not be confused when mixing the CA and Non-CA data-sets. We utilize the same ML types and techniques for the classification problem to find the most optimal hyper-parameters as before. However, due to space limitations, we do not discuss the details. The results show that the MLP classification model could, with 92 percent of accuracy and 0.11 MSE, predict the occurrence of CA based on the features from Table 1.

6.2.3 Feature Importance and Learning Curves

Table 5 depicts the features that were chosen by the XGBoost to include in the model for both LTE and non-standalone 5G. From the original 17 features on Fig. 6, XGboost selected 10 via feature exclusion and grid search regarding LTE. Table 5 also covers two methods for determining feature importance - local and global. The global feature importance is a standard method from the scikit-learn ensemble library that evaluates the importance after the fitting and prediction of the model. The local feature importance is a separate process that uses SHapley Additive explanations (SHAP) to test how the model is performing

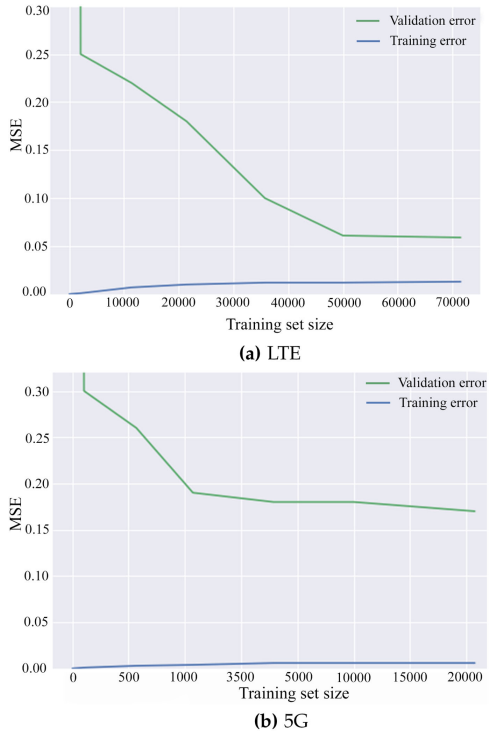


Fig. 8. Learning curves that observe the bias and variance of the predictions by increasing the training set size.

with and without a particular feature [43]. Table 5 show the normalized SHAP feature importance, illustrating the average impact on model output magnitude. Although the selected features almost equally share their contribution, it is worth mentioning that p_a and SINR emerge as the most important features on average. First, as expected, the SINR has a direct effect on the throughput since it measures the interference of the signal. Moreover, SINR has a relationship to the cell load, as it is used in the static formula provided by Chang *et al.* [21] Second, p_a measures the energy reduction of user's data symbols compared to the RSs, which gives valuable information about the relevance of the whole set of RS features.

One popular validation method for over/under-fitting is bias and variance observations over the train and validation errors [44]. Fig. 8 plots the learning curve for XGBoost regarding (a) LTE and (b) 5G. The idea is to observe the rate of change in train and validation errors while iteratively increasing the training data size. For instance, the model first starts with only 1 training sample in the training set, computes the predictions, and produces the validation and training error. As expected, in such a case, the model fits almost perfectly the training data, thus low training error. However, it yields high validation error as it performs poorly on unseen data, which indicates over-fitting. Further, as the training set size increases, the models learn new patterns in the data-set and gradually increase the training and decrease the validation error, reaching an arguably acceptable MSE of 0.06 for LTE (Fig. 8a). As a rule of thumb, high training error indicates high bias, associated with under-fitted model, while a large difference among train and validation error signals high variance and over-fitted model [44]. Note that the validation error, especially in the LTE case, saturates after 50.000 training data points, which would suggest that adding more data-sets will not benefit the model [44]. Moreover, due to the validation error saturation after 50.000 points, one may further consider splitting the training data into creating additional data-set for independently testing the accuracy of the model, as suggested by Russell and Norvig [42]. In the following, Section 6.3 addresses the higher obtained variance in the 5G results (Fig. 8b).

6.3 Prediction Results in 5G

The experimental results from the 5G cellular network, as mentioned before, were conducted on a non-standalone deployment, having LTE as an underlying core network. The UE stays on LTE unless it begins using 5G supported applications, such as UDP streaming. Thus, non-standalone 5G is only available in *connected mode*. It is important to note that commercial standalone 5G networks are currently unavailable. As a result, in the following, we describe the achieved prediction results in a non-standalone 5G network, while also discussing the possible implementations in a standalone network.

6.3.1 Non-Standalone 5G

There are few key differences in applying the LTE ML models, from Section 6.2, into the 5G settings. The feature set is limited due to the non-standalone 5G network. That is, not every feature from Table 1 is measurable in 5G settings, such as p_a, p_b, and RSSI. Instead, TEMS Pocket was able to measure other metrics of interest, such as CRI and CQI. Since the 5G cells supported only 100 MHz bandwidth and 30 kHz SCS, the features (Table 1) still represent the whole cell's bandwidth, and thus, can be used to directly predict the throughput. In addition, the tested 5G cells offer a single slice and 5QI options only. Hence, a single static NSSAI value was available, with 5QI of 84, which represents real-time streaming [4]. The ML model excludes the NSSAI and 5QI values as they are constant. In future deployments with many pre-configured NSSAI and 5QI, we suggest using the same method that predicts throughput with and without CA, as in Section 6.2.2. That is, NSSAI and 5QI, as features, can be used to split the data-set when benchmarking each network slice.

The results from Table 5 show that MLP and XGBoost achieve similar accuracy of around 84 percent (R^2) when predicting the available throughput per network slice, per given 5QI value, using the same procedure data splitting as for LTE. Fig. 8b depicts the bias and variance observation for the results on the 5G network. Although the training error is relatively low, the training and validation difference is larger than in LTE, referring to high variance. Also, the constant validation error of around 0.17 MSE suggests that adding new data-points will not improve the model. High variance and a stable validation curve can indicate an over-fitted model [44]. A conclusion is that the 5G model overfits with a training set size of above 3500. Different meta learning methods, such as meta-regressors and model-tuning, were tested for reducing the variance without any significant improvement, resulting with increased model complexity. For instance, meta-regressors were utilized which stack multiple models while combining their predictions, as

well as model-tuning as meta-learning technique proposed by Hutter *et al.* [45]. One reason for the high variance is that the 5G network experiments were limited to two use-cases - walking at a university campus and low-speed drive tests in the underground mine. Moreover, the large difference among the *Total LTE* and *Total 5G* prediction accuracy may be due to the reduced number of features in the 5G case, which also contributes to the outlook of the Fig. 8b.

6.3.2 Standalone 5G

In this part, we consider a future deployment of standalone 5G network, with full implementation of the 3GPP requirements regarding network slicing [4]. The goal of 5G in covering various traffic requirements creates multiple connectivity options regarding bandwidth, MCS, SCS, beam-forming, NSSAI, and 5QIs. Each combination of these options will dramatically change the available throughput and, thus, should be considered as features in the future ML model predicting the available throughput per NSSAI. In the following, we will discuss the impact of each of these features on the throughput.

The research community focuses on understanding the throughput limits per networking slice [14], [20], [46]. Motivated by the legacy Shannon formula (1), the cited studies attempted to statically model the standalone 5G available throughput. A direct implementation of (1) is impossible in the context of 5G due to the nature of NOMA. For instance, NOMA enables the cell to serve multiple slices at the same time and frequency, following Fig. 1b, and whose transmissions will interfere with each other [14]. Within NOMA, successive interference cancellation (SIC) was proposed to enable the receiver to iteratively decode signals of distinct UEs [46]. Thus, SIC has to be taken into account when developing a feature in predicting the throughput per NSSAI. For instance, Ding *et al.* [20] discuss an extension of (1) by measuring the data-rates, in terms of bits per channel use (bpcu), of two UEs sharing the same resources in time/frequency domain

$$UE_1 = Bw_{eff} \times Log_2\left(1 + \frac{\rho \times \alpha_A \times |h_A|^2}{1 + \rho \times \alpha_B \times |h_A|^2}\right), \qquad (3)$$
$$UE_2 = Bw_{eff} \times Log_2(1 + \rho \times \alpha_B \times |h_B|^2).$$

Herein, ρ denotes the transmit SINR, α_A and α_B are the power allocation coefficients, while $|h_A|$ and $|h_B|$ denotes the channel gains for UE_1 and UE_1, respectively. From (3) it is noticeable that the power allocations for each UE greatly affects the user throughput performance and, thus, the MCS used for data transmission of each UE [46]. Another large influence on the throughput is BW_{eff}, representing the bandwidth efficiency. Compared to LTE, with 6 different channel bandwidth options (1.4, 3, 5, 10, 15, 20 MHz) and static RB size of 180 KHz, 3GPP seeks to increase the channel bandwidth options for 5G with 50, 100, 200, and 400 MHz, with SCS of 15, 30, 60, and 120 [4], which dramatically increases the RB size.

In addition, Popovski *et al.* [14] points out that it is important to consider the error probability in (3), that is known for each slice. The expected error probability, or PER, as in Table 2, can be utilized to determine which UE's packets should be decoded first. For instance, URLLC slices with 10^{-6} will get higher priority than the eMBB slices with 10^{-3} PER. In such cases, in H-NOMA with SIC, the URLLC transmissions should be decoded first while treating eMBB signals as additional noise.

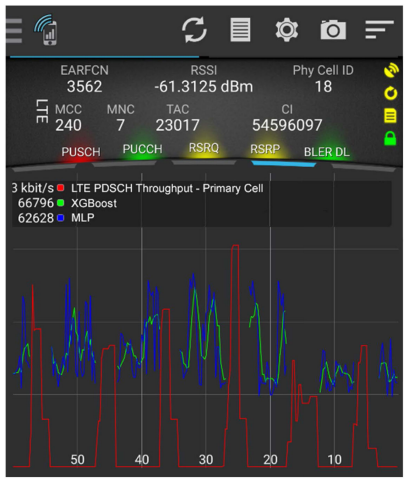


Fig. 9. A screenshot of TEMS Pocket, with deployed ML models running on a mining vehicle. The predicted values (green, blue) follow the pattern of the true measured throughput (red line).

Due to several options of channel bandwidth and SCS, the RSs will represent measurements only for a specific channel bandwidth and SCS, sent from the gNB via the MIB [4]. Thus, compared to LTE, the RSs will not describe the whole spectrum, resulting with inability to develop an equation similar to (2). However, motivated by the results from Section 6.2, we can conclude that utilizing the RSs will still be beneficial for the ML model, even in standalone 5G. For instance, RS will capture the impact of network coverage and interference on the throughput, especially in a dynamic environment [19]. In addition, RSs measure other unique conditions of the 5G network that may affect the throughput, such as the radio conditions per beam, translated into beam performance. However, we expect the RSs to have lower influence to the throughput in 5G compared to the LTE networks.

7 Discussion and Future Work

Two of the ML models within this study were chosen for real-life deployment on vehicles that are part of the industrial mining automation use-case (Fig. 3). To do so, an exported, pre-trained version of the XGBoost and MLP models were integrated on TEMS Pocket, shown on Fig. 9.

The ML models retrieve the features (Table 1) via TEMS Pocket and perform feature engineering, as described in Section 5. The length of the feature window size stays the same as in the training process, i.e., two 500ms windows. Thus, the model produces a prediction of the maximum available throughput per 1 second. The models produce the predictions (blue and green lines) with measurements during idle periods, thus, lack of predictions when the UE sends/receives traffic (red line). Note that this is a trade-off between the ability to perform the predictions and having an accurate model. Namely, as discussed in Section 5.1, the features' values have completely different patterns in *idle*

and *connected mode*, due to the way networks are designed. In addition, we observe that the features' values during *connected mode* are vendor-specific, i.e., have different patterns from phone to phone. Thus, the decision to train the model based on in-between idle measurement windows of 500ms. Such an approach can lower the overall accuracy of the predictions, as we do not consider the features' values during the *connected mode* (1500ms). However, given the high accuracy, the ML model can understand what is happening during the 1500ms from the in-between RF measurements.

A complete remote-control service would require fulfilment of the traffic requirements from the Table 2. Therein, network slices have unique requirements for the throughput regarding video streaming, sensor stream (UL), remote-control stream (DL). The main benefit of the proposed ML model during a future real-life deployment can be associated with the following events:

- Monitor the overall status of the achievable throughput in real-time in order to optimize the scheduling of competing traffic within a vehicle. For instance, the vehicle can prioritize the critical traffic, such as the remote-control stream, when low throughput is predicted.
- Map the throughput predictions to various service quality metrics, such as productivity, efficiency, safety, and MOS. For instance, extend Fig. 9 to plot the productivity of the remote-control service retrieved from a throughput utility function.
- Develop a separate ML model that, in real-time, loads a history of the predicted throughput, a time-series data, and forecast the throughput into the next few seconds in the future.
- Possibility to conduct a root cause analysis on a produced prediction. For instance, we have previously analyzed each of the decisions by random forest trees to produce a range of values for each feature that may lead to low predictions [37].

8 Conclusion

Benchmarking the available throughput, as a metric, is especially important in the era of network slicing. 5G will cover multiple use-cases involving industry automation, traffic safety, and tactile internet, each of which will require unique tailor-made network slices. For instance, a connected vehicle may be simultaneously connected on multiple slices, each with various traffic requirements. The critical nature of the vehicle's communications, combined with a dynamic wireless cellular environment, requires rigorous network monitoring in fulfilling SLAs, where throughput emerges as a key performance metric. This paper proposes a non-network intrusive machine learning model that predicts the available throughput in a non-standalone 5G network, benchmarking a network slice. Due to the limited deployment options for the 5G network, we first develop and validate the model on a conventional LTE network. The aim is to collect data from various real-life conditions during driving in urban, sub-urban, and rural areas, as well as stationary and walking tests in home/office and crowded areas, such as shopping malls and live sports events. Then, we apply and validate the model in a real-life 5G non-standalone network on an university campus and underground mine. The achieved results show 93 and 84 percent accuracy (R^2), 0.06 and 0.17 MSE, when predicting the throughput in LTE and 5G network respectively.

Acknowledgments

The authors would like to thank the reviewers from Luleå University of Technology and Infovista Sweden AB for improving the quality of paper and to Per Johansson, Irina Cotanis, and Ulf Marklund, all from InfoVista Sweden AB, for their contributions to this work.

References

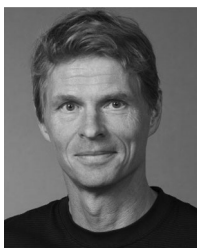
[1] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.
[2] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for vehicular communications," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 111–117, Jan. 2018.
[3] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2429–2453, Jul.–Sep. 2018.
[4] 3GPP, "System architecture for the 5G System. TS 23.501, release 16," 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/16.06.00_60/ts_123501v160600p.pdf
[5] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5G network slicing for vehicle-to-everything services," *IEEE Wireless Commun.*, vol. 24, no. 6, pp. 38–45, Dec. 2017.
[6] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *IEEE Netw.*, vol. 24, no. 2, pp. 36–41, Mar./Apr. 2010.
[7] K. Mitra, A. Zaslavsky, and C. Åhlund, "Context-aware QoE modelling, measurement, and prediction in mobile computing systems," *IEEE Trans. Mobile Comput.*, vol. 14, no. 5, pp. 920–936, May 2015.
[8] Y. Chen, K. Wu, and Q. Zhang, "From QoS to QoE: A tutorial on video quality assessment," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 1126–1165, Apr.–Jun. 2015.
[9] D. Minovski, C. Åahlund, and K. Mitra, "Modeling quality of IoT experience in autonomous vehicles," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3833–3849, May 2020.
[10] C. Yue, R. Jin, K. Suh, Y. Qin, B. Wang, and W. Wei, "Linkforecast: Cellular link bandwidth prediction in LTE networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 7, pp. 1582–1594, Jul. 2018.
[11] D. Raca *et al.*, "On leveraging machine and deep learning for throughput prediction in cellular networks: Design, performance, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 3, pp. 11–17, Mar. 2020.
[12] J. Yao, S. S. Kanhere, and M. Hassan, "Improving QoS in high-speed mobility using bandwidth maps," *IEEE Trans. Mobile Comput.*, vol. 11, no. 4, pp. 603–617, Apr. 2012.
[13] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, Jul.–Sep. 2019.
[14] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *IEEE Access*, vol. 6, no. 1, pp. 55 765–55 779, Sep. 2018.
[15] E. Fersman *et al.*, "AI for 5G slicing: From a reactive to proactive approach," *Ericsson, Kista, Stockholm, Sweden, Whitepaper*, 2019. [Online]. Available: https://www.mobileworldlive.com/ai-for-5g-slicing-from-a-reactive-to-proactive-approach
[16] S. E. Elayoubi, S. B. Jemaa, Z. Altman and A. Galindo-Serrano, "5G ran slicing for verticals: Enablers and challenges," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 28–34, Jan. 2019.
[17] 3GPP, "LTE: Physical layer measurements. TS 36.214, release 16," 3GPP, Mar. 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/136200_136299/136214/14.02.00_60/ts_136214v140200p.pdf
[18] G. Ku and J. M. Walsh, "Resource allocation and link adaptation in LTE and LTE advanced: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1605–1633, Jul.–Sep. 2015.

[19] S. Chen, J. Hu, Y. Shi, and L. Zhao, "LTE-V: A TD-LTE-Based V2X solution for future vehicular network," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 997–1005, Dec. 2016.
[20] Z. Ding *et al.*, "Application of non-orthogonal multiple access in LTE and 5G networks," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 185–191, Feb. 2017.
[21] K. Chang and R. P. Wicaksono, "Estimation of network load and downlink throughput using RF scanner data for LTE networks," in *Proc. Int. Symp. Perform. Eval. Comput. Telecommun. Syst.*, 2017, pp. 1–8.
[22] R. P. Wicaksono, S. Kunishige, and K. Chang, "Scanner based load estimation for LTE networks," in *Proc. Int. Conf. Inf. Commun. Technol. Convergence*, 2015, pp. 413–418.
[23] E. Takahashi, T. Suzuki, T. Onishi, and K. Satoda, "Autonomous off-peak data transfer by passively estimating overall LTE cell load," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf.*, 2017, pp. 754–759.
[24] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, "A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1790–1821, Jul.–Sep. 2017.
[25] J. Xie *et al.*, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 393–430, Jan.–Mar. 2018.
[26] F. Jomrich, A. Herzberger, T. Meuser, B. Richerzhagen, R. Steinmetz, and C. Wille, "Cellular bandwidth prediction for highly automated driving-evaluation of machine learning approaches based on real-world data," in *Proc. 4th Int. Conf. Veh. Technol. Intell. Transp. Syst.*, 2018, pp. 121–132.
[27] I. F. Akyildiz, P. Wang, and S. -ChunLin, "Softair: A software defined networking architecture for 5G wireless systems," *Comput. Netw.*, vol. 85, pp. 1–18, May 2015.
[28] I. Budhiraja, S. Tyagi, S. Tanwar, N. Kumar, and J. J. P. C. Rodrigues, "Tactile internet for smart communities in 5G: An insight for NOMA-based solutions," *IEEE Trans. Ind. Inform.*, vol. 15, no. 5, pp. 3104–3112, May 2019.
[29] S. Timotheou and I. Krikidis, "Fairness for non-orthogonal multiple access in 5G systems," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1647–1651, Oct. 2015.
[30] J. Ni, X. Lin, and X. S. Shen, "Efficient and secure service-oriented authentication supporting network slicing for 5G-enabled IoT," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 644–657, Mar. 2018.
[31] J. Wang *et al.*, "Networking and communications in autonomous driving: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1243–1274, 2018.
[32] X. Cheng, R. Zhang, and L. Yang, "Wireless toward the era of intelligent vehicles," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 188–202, Feb. 2019.
[33] G. T. 22.185, "Service requirements for v2x services," 2018. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/122100_122199/122185/14.03.00_60/ts_122185v140300p.pdf
[34] K. Hedayat *et al.*, "A two-way active measurement protocol (twamp)," Tech. Rep. RFC 5357, Oct., 2008.
[35] S. Baillargeon, C. Flinta, and A. Johnsson, "Ericsson two-way active measurement protocol (twamp) value-added octets," *IETF RFC 6802*, IETF, vol. 1, no. 1, pp. 1–17, Nov. 2012.
[36] I. G. recommendation, "Latency measurement and interactivity scoring under real application data traffic patterns," ITU-T, vol. 1, no. 1, pp. 1–20, Aug. 2020.
[37] D. Minovski *et al.*, "Analysis and estimation of video QoE in wireless cellular networks using machine learning," in *Proc. 11th Int. Conf. Qual. Multimedia Experience*, 2019, pp. 1–6.
[38] InfoVista, "Tems pocket," Jun. 2020. [Online] Available: https://www.infovista.com/tems/pocket
[39] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.
[40] A. Courville, I. Goodfellow, and Y. Bengio, *Deep learning*. Cambridge, MA, USA: MIT Press, 2016.
[41] J. Nalepa and M. Kawulok, "Selecting training sets for support vector machines: A review," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 857–900, Jan. 2019.
[42] S. Russell and P. Norvig, "Artificial intelligence: A modern approach," Harlow, Essex, U.K.: Harlow Pearson Education Limited, 2002.
[43] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *in Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4768–4777.
[44] C. M. Bishop, *Pattern Recognit. and Mach. Learn.* New York, NY, USA: Springer, 2006.
[45] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated Machine Learning: Methods, Systems, Challenges.* Cham, Switzerland: Springer, 2019.
[46] Y. Saito, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, "Non-orthogonal multiple access (NOMA) for cellular future radio access," in *Proc. IEEE 77th Veh. Technol. Conf.*, 2013, pp. 1–5.

Dimitar Minovski is currently working toward the PhD degree with the Luleå University of Technology, Sweden, and InfoVista Sweden AB. His research interests include quality of experience, IoT, machine learning, and wireless access networks.

Niclas Ögren received the BSc degree in electrical engineering emphasized on digital signal processing from the Blekinge Institute of Technology, Karlskrona, Sweden in 1995. Since 1995, he was with measurement solutions for radio access of mobile telecommunication systems from 2G to 5G, and currently works as a specialist of Network and Protocols. His research interests include radio access technology, radio measurement, QoS, and QoE.

Karan Mitra (Member, IEEE) received the BIS degree (Hons.) from Guru Gobind Singh Indraprastha University, New Delhi, India, in 2004, the MIT (M.T.) and PGradDipDigComm degrees from Monash University, Melbourne, V.I.C., Australia, in 2006 and 2008, respectively, and the Dual-Badge PhD degrees from Monash University and Luleå University of Technology, Skellefteå, Sweden, in 2013. He is currently an assistant professor with the Luleå University of Technology. His research interests include quality of experience modeling, measurement and prediction, context-aware computing, cloud computing, and mobile and pervasive computing systems. He is a member of ACM.



Christer Åhlund received the PhD degree from the Luleå University of Technology, Skellefteå, Sweden, in 2005. He is currently a chair professor of pervasive and mobile computing with the Luleå University of Technology. He is also the scientific director of excellence in research and innovation named Enabling ICT. He has 12 years of industry experience in the ICT area. His research interests include Internet of Mobility, wireless access networks, Internet of Things, and cloud computing.