A Machine-Learning-Based Framework for Optimizing the Operation of Future Networks

Claudio Fiandrino, Chaoyun Zhang, Paul Patras, Albert Banchs, and Joerg Widmer

The authors develop a general machine-learning-based framework that leverages artificial intelligence to forecast future traffic demands and characterize traffic features. This makes it possible to exploit such traffic insights to improve the performance of critical network control mechanisms, such as load balancing, routing, and scheduling.

Abstract

5G and beyond are not only sophisticated and difficult to manage, but must also satisfy a wide range of stringent performance requirements and adapt quickly to changes in traffic and network state. Advances in machine learning and parallel computing underpin new powerful tools that have the potential to tackle these complex challenges. In this article, we develop a general machine-learning-based framework that leverages artificial intelligence to forecast future traffic demands and characterize traffic features. This makes it possible to exploit such traffic insights to improve the performance of critical network control mechanisms, such as load balancing, routing, and scheduling. In contrast to prior works that design problem-specific machine learning algorithms, our generic approach can be applied to different network functions, allowing reuse of existing control mechanisms with minimal modifications. We explain how our framework can orchestrate ML to improve two different network mechanisms. Further, we undertake validation by implementing one of these, mobile backhaul routing, using data collected by a major European operator and demonstrating a 3× reduction of the packet delay compared to traditional approaches.

Introduction

Recent advances in machine learning (ML) enable optimization at levels of complexity that were previously unaffordable. This has led to dramatic performance improvements, fostering the use of ML algorithms like neural networks across a wide range of fields.

Harnessing ML to enhance the performance of wireless networks started with 5G and will be essential to promote zero-touch configuration and management, thereby enabling self-configuration and self-optimization envisioned for 6G networks [1]. Wireless network operation depends on many variables that are not always known at the time decisions need to be made and which cannot be easily forecast or inferred. Furthermore, wireless networks are increasingly complex and heterogeneous, as they comprise many different radio access technologies and modules that mutually interact, need to satisfy diverse evolving requirements, and have to adapt quickly to changes. This renders the problem of real-time performance optimization of wireless systems prohibitive for traditional techniques. In contrast, the ability of ML tools to handle very complex systems makes them suitable for managing highly dynamic wireless networks and make more intelligent decisions (e.g., based on predicted future traffic patterns) [2].

Stemming from these observations, this article proposes a modular ML-based wireless network optimization framework, which enables plug-and-play integration of machine intelligence into new, as well as existing, network functions. Specifically, we leverage ML to forecast future traffic volume, and characterize traffic features. We then feed this information into network control mechanisms to improve their performance. The advantage of our approach is two-fold: it is sufficiently general and allows instantiating ML pipelines across different network elements and functions, thus being compliant with the recent International Telecommunication Union — Telecommunication Standardization Sector (ITU-T) Recommendation Y.3172 for integrating ML in future networks [3], and it permits retrofitting ML to legacy architectures and reusing existing network control mechanisms with minimal or no modifications.

Previous works embed ML into the design of specific algorithms, focusing on network functions including:

- Mobility management, resource management and orchestration, and service provisioning [4]
- Detection and channel estimation in massive multiple-input multiple-output (MIMO) systems [5]
- Routing [6]
- Resource scaling of virtual network functions (VNFs) [7]

Their main drawback is precisely that they are mechanism-specific, that is, each network control mechanism requires a purpose-built ML approach and cannot easily be reused.

In contrast, we use ML to make accurate traffic predictions that can be straightforwardly used as input to well-established algorithms and decision modules. Traffic forecasting and characterization using ML has received significant research interest [8, 9]. However, previous work largely focuses on traffic analysis to optimize specific network operations, for example, routing (see [10] for a survey of ML techniques applied to software defined networking, SDN) or VNF resource scaling [7], while our approach relies on traffic analytics to improve the performance of generic network control mechanisms.

Digital Object Identifier:
10.1109/MCOM.001.1900601

Claudio Fiandrino, Albert Banchs, and Joerg Widmer are with IMDEA Networks Institute; Chaoyun Zhang and Paul Patras are with the University of Edinburgh.

0163-6804/20/$25.00 © 2020 IEEE

In addition, we incorporate an ML orchestrator that is responsible for managing and monitoring resources, but also for deriving suitable configurations for training ML models. We expect that instantiating an ML pipeline for a specific function with the aid of our framework will bear similar costs to those incurred by purpose-built ML algorithms serving the same purpose. The one-off signaling cost associated with the orchestration of a function in our approach is a small price to pay for the flexibility it enables.

To demonstrate the feasibility and performance gains achievable with our framework, we show how to orchestrate two ML pipelines: traffic-driven VNF scaling and routing in mobile backhaul networks. We practically evaluate the latter use case. By feeding a state-of-the-art routing scheme with city-scale forecasts of future traffic consumption, obtained with a deep learning structure, our framework attains 3× reductions in packet delay.

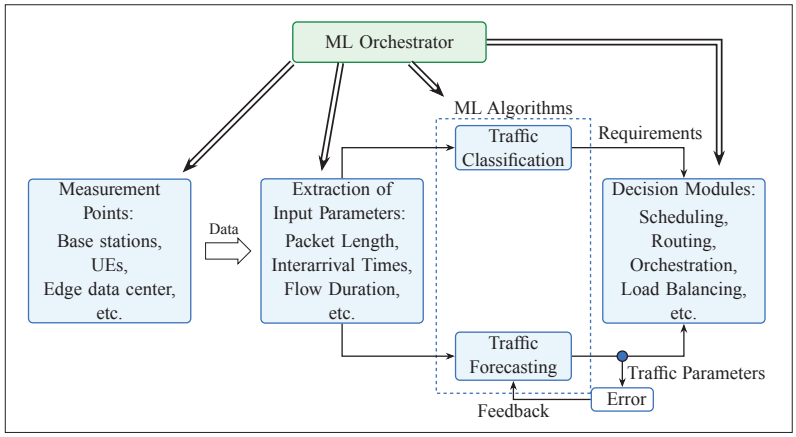

Figure 1. Building blocks of the proposed framework. ML algorithms used to characterize and forecast traffic based on measurements and flow metadata. Knowledge extracted is fed to modules implementing network functions.

ML-Based 5G Network Optimization

We propose an ML-based framework for network optimization and explain how to incorporate it within 5G networks.

ML-Based Framework

Figure 1 illustrates the building blocks of our framework, which comprise:

- The ML orchestrator
- Modules to measure mobile network traffic
- ML algorithms that process this data
- Modules performing specific network optimizations based on the output of the ML algorithms

According to the specific network function to optimize, the orchestrator defines in the form of a template the set of collector nodes, the duration and the aggregation level of traffic measurements, and ML pipeline-specific parameters, such as number of epochs, layers, and possibly a custom loss function as in [11]. Different functions require different inputs; for example, while routing needs to monitor traffic from a set of base stations to decide optimal routes, scaling computational resources of VNFs executing core services requires monitoring control traffic from the same set of base stations. The orchestrator thus coordinates the instantiation of an ML pipeline accordingly, along with the mechanisms to update the decisions for each network function (e.g., by interacting with the VNF orchestrator) and ensures sufficient computing capacity is provisioned to train ML models in either a centralized or a distributed manner.

Gathering measurements requires direct access to flow information available at, for example, SDN switches or base stations. Rather than defining a limited set of input features, the measurement modules extract sequences of packets of each flow, along with their lengths, inter-arrival times, direction (uplink/downlink), and possibly even parts of the payload. The key advantage of working with such comprehensive data as input is that abstract features can be extracted automatically during training, instead of relying on a restricted and manually identified set. Indeed, feature engineering is costly and may lead to poor performance [12]. Further, new use cases may require different features. Thus, our approach is future-proof.

Our framework is general enough to allow for different learning techniques. We focus on deep learning (DL) because:

- DL algorithms scale better than ML approaches as the volume of data grows.
- In network settings where inferences must be made based on a large number of input parameters, DL produces highly accurate outputs.
- Advances in parallel computing enable complex neural networks to be trained rapidly and applying them in different settings without re-training.

We are especially interested in DL structures that can identify distinct types of flows within large aggregates in order to accommodate specific requirements, such as latency and reliability (traffic classification), and predict essential characteristics of future network traffic, such as average and peak data rates, level of burstiness, and so on (traffic forecasting). Depending on the target task, different DL structures can be employed [12]. Auto-encoders are particularly effective in traffic classification based on TCP traffic flow information. Structures typically used for image segmentation (e.g., convolutional neural networks, CNNs) are also effective in classification. Optimization of network functions like scheduling and load balancing depends on the ability to accurately classify traffic.

Traffic forecasting depends significantly on temporal features. Long short-term memories (LSTMs) work well with time series. Similarly, the convolution operation can be extended to the temporal dimension to construct a 3D-CNN, thereby extracting spatio-temporal features that are characteristic of mobile traffic [13]. Figure 2 illustrates a deep learning pipeline tailored to mobile traffic forecasting. City-level traffic measurements are fed into stacks of such 3D-CNNs and ConvLSTMs to extract spatio-temporal features within traffic snapshots that a group of fully connected layers uses to make future traffic predictions per eNB.

Finally, our decision modules (Fig. 1) are based on existing algorithms that only need to be updated to take as input the predictions made by the

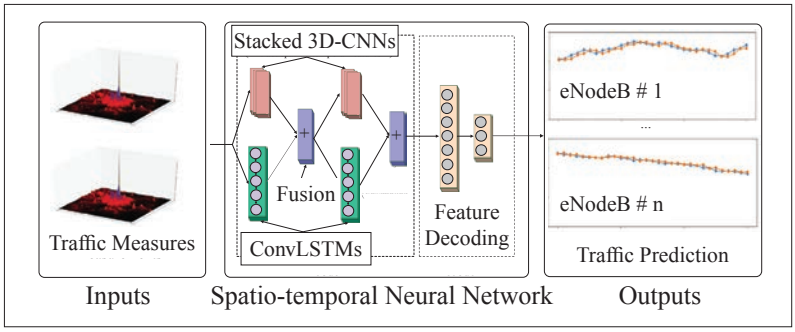

Figure 2. Example of a deep learning pipeline for mobile traffic forecasting adapted from [13]. City-level measurements are fed into stacks of 3D-CNNs and ConvLSTMs, which extract spatio-temporal features to predict future traffic demands at the eNodeB level.

DL algorithms. Thus, with our framework the basic operation of these algorithms remains unmodified. This provides much greater control over their operation, unlike fully ML-based solutions whose decisions are made directly by ML algorithms.

5G Architecture Integration

The design of our framework adheres to the ITU-T guidelines [3]. Although it is tailored specifically for (beyond) 5G networks, the solution is backward-compatible with 4G networks (except with limited range of network mechanisms that can be optimized). Our framework can instantiate measurement modules in the core, backhaul/fronthaul, and radio access network (RAN), while exploiting standard interfaces to extract measurements from the network equipment.

Figure 3 shows a realization of our architecture. The sources (src) are user equipments (UEs) and base stations (eNodeB/eNBs and gNBs according to 4G and 5G terminology respectively) with diverse radio access technology (RAT). Our framework can also interact with other sources, such as network data analytics functions (NWDAFs), which 3GPP has introduced in Release 15, and the RAN data analytics function (RAN-DAF — not within 3GPP's umbrella) to extract user location, cell/slice ID, cell/slice load, channel quality, amount of transmitted/received data, and so on. This allows data collection (collector c) close to where it can undergo pre-processing, so as to expose properly formatted inputs to deep learning algorithms by applying a set of transformations to map the measurements into a tensor format that learning algorithms can process.

According to the specific optimization objective, the training and inference phases take place either in edge clouds or at the level of individual base stations/routers/UEs, depending on the required computing capabilities. Traditionally, training an ML model requires moving all the data into a central location, which is the result of trading privacy and bandwidth (e.g., backhaul links) to obtain accurate models. With federated learning, an ML model is trained in a distributed manner, which better suits scenarios with the UE in the loop and entails minimal communication overhead, for example, with data compression or reduction on the number of updates per node. This allows the computing resources of the federated nodes to be harnessed, while partial model parameter updates affect the model's convergence speed only marginally. Once the output parameters are determined (e.g., an update of the scheduling or routing policy), they are distributed to the sink nodes (e.g., the RATs in Fig. 3).

Use Cases for Network Optimization

We now cover a diverse set of use cases that our framework can serve.

Semi-Persistent, Elasticity-Aware, and Latency-Aware Scheduling

Different types of traffic have different levels of latency requirements, ranging from extremely small (e.g., ultra-reliable low-latency communications, URLLC), to medium (interactive voice or video), all the way to slack latency requirements (media streaming). ML can extract specific flow characteristics and the associated latency requirements, and feed this information to schedulers.

With semi-persistent scheduling, periodic uplink flows can be assigned transmission slots without notifying the scheduler of the UE's queue occupancy. Such allocations will change dynamically over time to adapt to changes in the modulation and coding scheme (MCS) that reflect the perceived channel quality. Reserving resources in advance brings significant advantages, that is, reduced control overhead and signaling load (which is critical in dense networks) and notifications to neighboring cells of these reservations, to better coordinate transmissions and limit intercell interference. While this technique has many advantages, it requires flow classification, that is, inferring whether a given flow is amenable to this type of scheduling, its periodicity, and the resources required for each period.

Elastic flows can (within limits) adapt to the available capacity; for example, HTTP video streaming applications can switch between codecs of different rates. Dropping packets of a flow directly reduces its rate by the corresponding fraction of packets in the case of UDP. When TCP is employed, packet dropping indirectly reduces the rate of that flow by triggering congestion control. Further, dropping packets of elastic flows reduces the quality of experience (QoE) of the corresponding users significantly less than reducing the rate of inelastic traffic. In contrast, URLLC traffic of industrial automation and tactile applications poses extreme requirements in terms of latency and reliability (1 ms target latency and 10^{-5} reliability as per 3GPP Rel. 15 specifications).

A prompt and correct classification of latency requirements enables the prioritization and scheduling of different flows to meet their deadlines, if necessary at the expense of serving less important traffic with higher delay. The classification of the level of elasticity allows intelligently dropping packets of specific flows so as to minimize QoE degradation and shaping the traffic of known applications early on, thereby preventing congestion.

Load Balancing

Advance knowledge of behavior and characteristics of different flows, such as the average/peak rate and level of burstiness, allows intelligent assignment of different flows to base stations and scale in and out VNF resources.

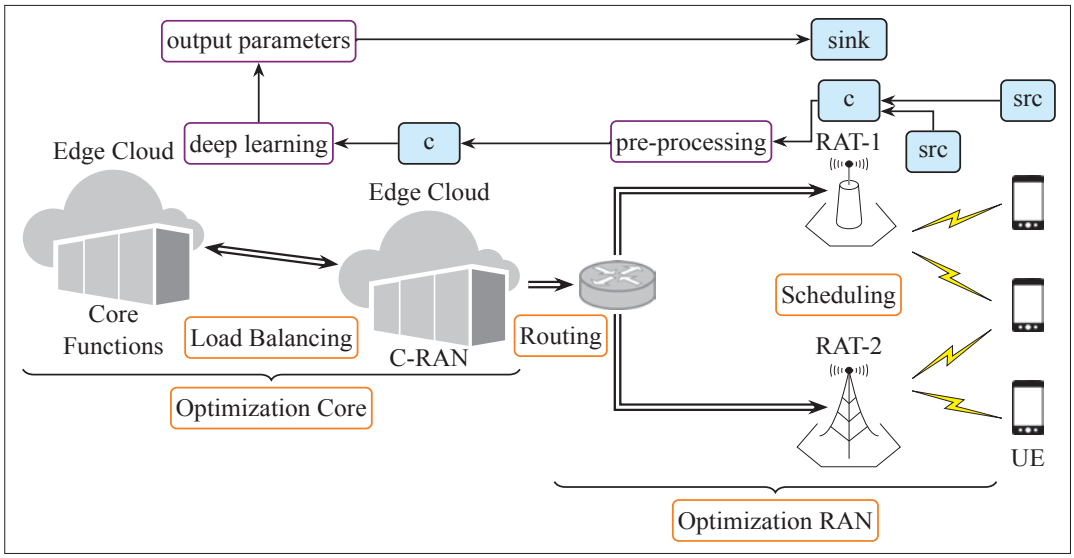

Figure 3. Integration of our framework in a 5G architecture. Workflow execution shown by highlighting measurement points and functions benefiting from ML.

In a 5G core network with control-user plane separation, the user plane function (UPF) takes care of the data path, while AMF and SMF cover the control path. AMF performs functions related to user authentication and authorization, and handles mobility management. SMF is in charge of all session management functions, allocates IP addresses, and selects the UPF.

Load balancing may be based on average load or the statistical behavior of the different flows and the probability that they congest the corresponding base station. More importantly, prediction-based load balancing can occur proactively rather than reactively (i.e., when some base stations have already become congested). In 5G scenarios, users may benefit from being simultaneously attached to conventional LTE and New Radio (NR) base stations. NR will typically operate in millimeter-wave (mmWave) bands, which offer much higher data rates but are more susceptible to blockage. Early prediction of blockage on NR links allows steering traffic over proper fallback LTE links.

Load balancing is also essential for VNF orchestration. The placement of the different VNFs serving a given flow depends on the traffic generated by that flow and the latency requirements of the application. Since moving or start-up new VNFs incur high overheads, it is important to predict these features as far in advance as possible, thus minimizing the disruption of their operation. For example, co-location in the same rack of VNFs executing RAN and core services that exchange a significant amount of signaling traffic, for example, access and mobility function (AMF) and session management functions (SMFs) is beneficial to reduce load peaks, queuing delays, and energy consumption. Additionally, traffic prediction is an enabler for joint optimization of radio configuration (MCS) and CPU allocation of VNFs executing RAN services [11]. Unlike core services, for which statistics about control traffic are required to optimize resource allocation and orchestration of VNFs, RAN services and flow assignment require data traffic statistics.

Mobile Traffic Routing

Efficient route management is essential for mobile networks, especially for high-rate and low-latency traffic flows, which may originate in edge clouds, be routed to the core, and then continue on to servers responsible for cloud RAN (C-RAN) processing. Poorly chosen paths lead to network congestion and degrade the quality of service. While typically routing protocols assign (static or dynamic) weights to links, the increasing popularity of SDN offers new degrees of freedom for routing (e.g., flow-based routing). With an ML algorithm that can learn flow statistics with high accuracy and mine traffic patterns, more sophisticated and accurate routing policies can be devised. In the next section, we present a realization of our framework that addresses this particular routing use case.

Orchestration and Use Case Implementation

We demonstrate how the proposed ML-based optimization framework can be instantiated to orchestrate simultaneous ML pipelines for learning-driven routing and VNF scaling. Further, we assess the benefits of ML in the former use case.

Let us consider part of a mobile core network, a cluster of eNBs deployed in the city of Milan (Fig. 4). We envision a set of routing nodes that connect the eNBs to the core network via diverse backhaul links. The traffic measurements were collected by Telecom Italia between November and December 2013 [14]. The experiments preserve the trends in the original dataset, but we scale up the demands by a factor of 10 according to current market studies on traffic consumption growth. Figure 5 illustrates different 30-min snapshots of the ground truth observations and the demand forecast with our D-STN at 4 selected eNBs (as numbered in Fig. 4).

Orchestration

In a 5G core network with control-user plane separation, the user plane function (UPF) takes care of the data path, while AMF and SMF cover the control path. AMF performs functions related to user authentication and authorization, and handles mobility management. SMF is in charge of all session management functions, allocates IP addresses, and selects the UPF (Fig. 4).

Our ML framework (Fig. 1) will simultaneously orchestrate the scaling of AMF and routing. The orchestrator instructs one ML-pipeline to gather

Our approach does not require fine-grained information about individual flows because it makes predictions based on aggregate traffic volumes. Further, by mixing predictions with historical information, this solution estimates the future traffic demand with small errors for long periods of time.

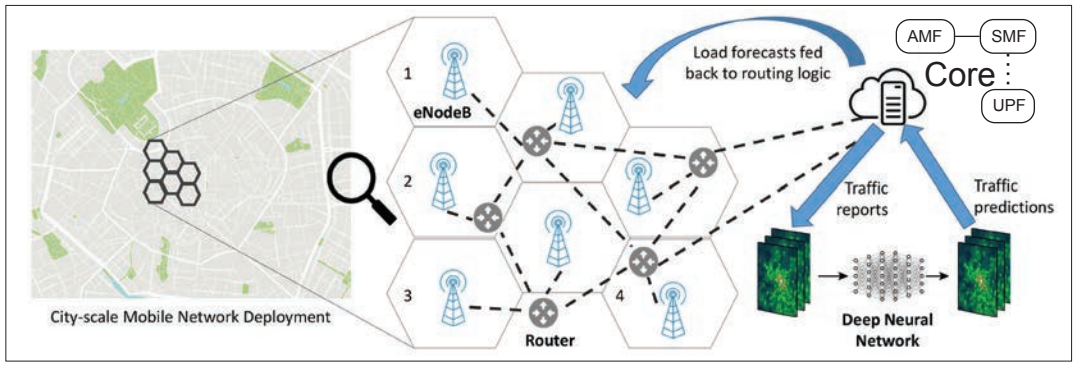

Figure 4. Example of an eNB cluster in the city of Milan, with our ML framework orchestrating, in the core network, VNF scaling and proactive routing, both based on traffic forecast measured per eNB.

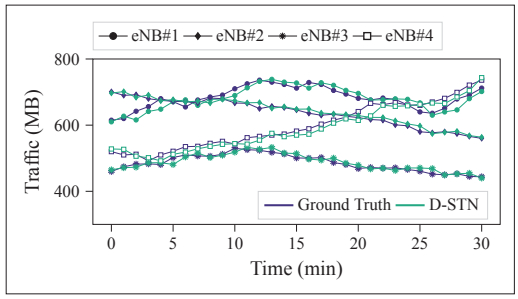

Figure 5. Traffic forecast with a deep spatio-temporal neural network (D-STN) at the four eNBs indicated in Fig. 4. Note that the predictions closely follow ground truth measurements.

control traffic to determine the number of accepted UEs and the time they need to attach to the network [7]. A second ML-pipeline is instructed to aggregate data traffic into minute-granularity summaries along with timing information and the coordinates of the corresponding source nodes ("Extraction of Input Parameters"). Both ML pipelines extract these parameters from mobile traffic observed at different eNodeBs ("Measurement Points" and "Data" in Fig. 1). Input parameters are fed to specific ML algorithms, for example, LSTM, and ConvLSTM and 3D-CNN structures for VNF scaling and routing, respectively ("Traffic Forecasting" module). Finally, the outputs, (i.e., the optimal number of VNFs and paths) are fed to the "Decision Module," which instructs the VNF orchestrator and routers.

Learning-Driven Proactive Routing

We now quantify the benefits of a specific ML pipeline. Our approach stems from the observation that traditional shortest path routing is increasingly being phased out because of the highly dynamic nature of traffic demands within mobile networks and progressively denser network deployments. More often, alternatives like backpressure routing [15] that forward packets based on information about queue sizes along possible paths are preferred. DL was proven to outperform conventional Open Shortest Path First (OSPF) by making routing decisions based on observed traffic patterns [6]. However, both load-based and early DL-powered solutions select routing paths *a posteriori* (i.e., only after network conditions changed). This increases delays, as traffic demand information may be stale by the time forwarding decisions are enforced.

Forecasting future traffic demands and making routing decisions *proactively* can circumvent these limitations. Unfortunately, widely used forecasting techniques (e.g., ARIMA) need to be fed continuously with measurement time series, which is expensive. Furthermore, such tools must be reconfigured each time they are deployed in a new network topology. In contrast, our framework adopts DL techniques that are trained offline once, and afterward can provide city-scale traffic predictions to routing logic without retraining.

Therefore, we incorporate our recently developed deep spatio-temporal neural network (D-STN) [13], which through ConvLSTM and 3D-CNN structures extracts abstract spatial and temporal features of mobile traffic, achieving high forecasting accuracy with only limited measurements. Our approach does not require fine-grained information about individual flows because it makes predictions based on aggregate traffic volumes. Further, by mixing predictions with historical information, this solution estimates the future traffic demand with small errors for long periods of time.

We assume the backhaul routers are interconnected with wireless backhaul links that employ 2 × 2 MIMO transceivers (up to 300 Mb/s nominal data rates). We emulate the traffic flowing from each of the eNBs (Fig. 4), assuming UDP packets with 1000-byte payload. UDP is commonly employed to tunnel user traffic traveling from eNBs to the core network. To assess the gains attainable with traffic forecasting driven routing, we measure the delay that packets experience from the moment they are injected into the backhaul at the eNB level until they reach the core network. We examine this delay when routes are established with a vanilla backpressure algorithm that makes *a posteriori* decisions (without any ML logic) to balance queue sizes as traffic arrives. We compare the performance of this approach with an enhanced version that makes path computation decisions based on traffic forecasting obtained with D-STN. Decisions are made with the granularity of one traffic prediction step (1 s), whereas the computation time is dominated by the inference time of the neural network (on the order of milliseconds). Figure 6 shows the cumulative distribution function (CDF) of the delays. Observe that even in the small topology considered, the median of the packet delay is 3× smaller with traffic forecasting driven routing. To appreciate the potential negative effects of incorrect traffic forecasts, we also show the latency performance in the ideal case where

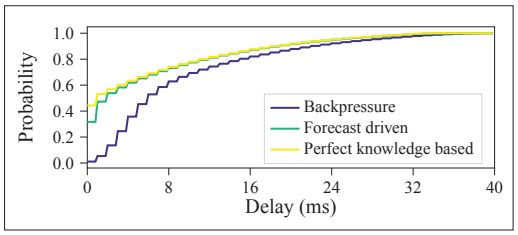

Figure 6. Delay CDF for the topology shown in Fig. 4. The path selection is made using a vanilla backpressure algorithm, an enhanced version that uses mobile traffic forecasts, and the ideal case where perfect knowledge of future traffic is available.

perfect knowledge of future traffic is available (i.e., no prediction errors). Our approach closely follows the ideal scenario, as the median of the delay is only marginally higher, which confirms that the proposed ML-based framework can bring substantial performance benefits in (beyond) 5G mobile networks.

Conclusions

In this article, we present an ML-based framework to optimize the operation of (beyond) 5G networks. Unlike current approaches that embed ML directly within network control systems, our framework does not require designing use-case-specific ML algorithms and changing existing network algorithms. Our framework instantiates ML pipelines to characterize traffic features and predict future traffic demands. The predictions are subsequently fed into existing network control mechanisms. Our approach brings together and harmonizes concepts from ITU-T, 3GPP, and other specifications in a single comprehensive framework. We show how our framework can instantiate multiple ML pipelines for different objectives. We implement and test our framework for one (i.e., proactive routing). Results indicate that even in small topologies, our solution reduces packet delay significantly.

We show how our framework can instantiate multiple ML pipelines for different objectives. We implement and test our framework for one (i.e., proactive routing). Results indicate that even in small topologies, our solution reduces packet delay significantly.

Acknowledgments

This work is partially supported by the Madrid Regional Government through the TAPIR-CM program (S2018/TCS-4496) and the Juan de la Cierva grant (FJCI-2017-32309). Paul Patras acknowledges the support received from the Cisco University Research Program Fund (2019-197006).

References

[1] Z. Zhang *et al.*, "6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies," *IEEE Vehic. Tech. Mag.*, vol. 14, no. 3, Sept. 2019, pp. 28–41.
[2] N. Bui and J. Widmer, "Data-Driven Evaluation of Anticipatory Networking in LTE Networks," *IEEE Trans. Mobile Computing*, vol. 17, no. 10, Oct. 2018, pp. 2252–65.
[3] ITU-T Rec. Y.3172, "Architectural Framework for Machine Learning in Future Networks Including IMT-2020," June 2019.
[4] R. Li *et al.*, "Intelligent 5G: When Cellular Networks Meet Artificial Intelligence," *IEEE Wireless Commun.*, vol. 24, no. 5, Oct. 2017, pp. 175–83.
[5] C. Jiang *et al.*, "Machine Learning Paradigms for Next-Generation Wireless Networks," *IEEE Wireless Commun.*, vol. 24, no. 2, Apr. 2017, pp. 98–105.
[6] N. Kato *et al.*, "The Deep Learning Vision for Heterogeneous Network Traffic Control: Proposal, Challenges, and Future Perspective," *IEEE Wireless Commun.*, vol. 24, no. 3, June 2017, pp. 146–53.
[7] I. Alawe *et al.*, "Improving Traffic Forecasting for 5G Core Network Scalability: A Machine Learning Approach," *IEEE Network*, vol. 32, no. 6, Nov./Dec. 2018, pp. 42–49.
[8] J. Wang *et al.*, "Spatiotemporal Modeling and Prediction in Cellular Networks: A Big Data Enabled Deep Learning Approach," *Proc. IEEE INFOCOM*, May 2017, pp. 1–9.
[9] M. Wang *et al.*, "Machine Learning for Networking: Workflow, Advances and Opportunities," *IEEE Network*, vol. 32, no. 2, Mar./Apr. 2018, pp. 92–99.
[10] J. Xie *et al.*, "A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges," *IEEE Commun. Surveys & Tutorials*, vol. 21, no. 1, 2019, pp. 393–430.
[11] J. A. Ayala-Romero *et al.*, "VrAIn: A Deep Learning Approach Tailoring Computing and Radio Resources in Virtualized RANs," *Proc. ACM Mobicom*, Oct 2019, pp. 1–16.
[12] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Commun. Surveys & Tutorials*, vol. 21, no. 3, 2019, pp. 2224–87.
[13] C. Zhang and P. Patras, "Long-Term Mobile Traffic Forecasting Using Deep Spatio-Temporal Neural Networks," *Proc. ACM MobiHoc*, June 2018, pp. 231–40.
[14] G. Barlacchi *et al.*, "A Multi-Source Dataset of Urban Life in the City of Milan and the Province of Trentino," *Scientific Data*, vol. 2, 2015.
[15] J. Núñez-Martínez *et al.*, "A Self-Organized Backpressure Routing Scheme for Dynamic Small Cell Deployments," *Ad Hoc Networks*, vol. 25, 2015, pp. 130–40.

Biographies

Claudio Fiandrino [S'14, M'17] is a postdoctoral researcher in the Wireless Networking Group at IMDEA Networks Institute. His primary research interests include ML-driven network optimization, multi-access edge computing, and crowdsensing.

Chaoyun Zhang is a final year Ph.D.student in the School of Informatics at the University of Edinburgh. His is interested in applications of deep learning to problems in the computer networking domain, including traffic analysis and network control.

Paul Patras [M'11, SM'18] is a reader (associate professor) and Chancellor's Fellow in the School of Informatics at the University of Edinburgh. His research interests include mobile intelligence, performance optimization, and IoT security and privacy.

Albert Banchs [M'04, SM'12] is currently a full professor with the University Carlos III of Madrid and deputy director of the IMDEA Networks institute. His research interests include performance evaluation and algorithm design in wireless and wired networks.

Joerg Widmer [M'06, SM'10, F'20] is a research professor as well as research director of IMDEA Networks, Madrid, Spain. His research focuses on wireless networks, ranging from extremely high-frequency millimeter-wave communication and MAC layer design to mobile network architectures.