Anticipatory Networking in Future Generation Mobile Networks: a Survey

Nicola Bui, Student Member, IEEE, Matteo Cesana, Member, IEEE, S. Amir Hosseinii, Student Member, IEEE, Qi Liao, Member, IEEE, Ilaria Malanchini, Member, IEEE, and Joerg Widmer, Senior Member, IEEE

Abstract—A growing trend for information technology is to not just react to changes, but as much as possible anticipate them. This paradigm made modern solutions such as recommendation systems a ubiquitous presence in today’s digital transactions. Anticipatory networking extends the idea to communication technologies by studying patterns and periodicity in human behavior and network dynamics to optimize network performance. This survey collects and analyzes recent papers leveraging context information to forecast the evolution of network conditions and, in turn, to improve network performance. In particular, we identify the main prediction and optimization tools adopted in this body of work and link them with objectives and constraints of the typical applications and scenarios. Finally, we consider open challenges and research directions to make anticipatory networking part of next generation networks.

Index Terms—Anticipatory, Prediction, Optimization, 5G, Mobile Networks.

I. INTRODUCTION

Evolving from one generation to the next, wireless networks have been constantly increasing their performance in many different ways and for diverse purposes. Among them, communication efficiency has always been paramount to increase the network capabilities without updating the entire infrastructure. This survey investigates anticipatory networking, a recent research direction that supports network optimization through system state prediction.

The core concept of anticipatory networking is that, nowadays, tools exist to make reliable prediction about network information and performance. Moreover, information availability is increasing every day as human behavior is becoming more socially and digitally interconnected. In addition, data centers are becoming more and more important in providing services and tools to access and analyze huge amounts of data.

As a consequence, not only is it possible for researchers to tailor their solutions to specific places and users, but also to anticipate the sequence of locations a user is going to visit or to forecast whether connectivity might be worsening, and to exploit the forecast information to take action before the event happens. This makes it possible to take full advantage of good future conditions (such as getting closer to a base station or entering a less loaded cell) and to mitigate the impact of negative events (e.g., entering a tunnel).

This survey covers a body of recent works on anticipatory networking, which share two common aspects:

- **Anticipation**: they either explore prediction techniques directly or consider some future knowledge as given.
- **Networking**: they aim to optimize communications in mobile networks.

In addition, this survey delves into the following questions: How can prediction support wireless networks? Which type of information is possible to be predicted and which applications can take advantage of it? Which tools are the best for a given scenario or application? Which scenarios envisioned for 5G networks can benefit from anticipatory networking? What is yet to be studied for anticipatory networking to be part of 5G networks?

II. BACKGROUND AND GUIDELINES

Anticipatory networking is the engineering branch that focuses on communication solutions that leverage the knowledge of the future evolution of a system to improve its operation. For instance, while a standard networking solution would answer the question “which is the best user to be served?”, an anticipatory equivalent would answer “which are the best users to be served in the next time frames given the predicted evolution of their channel condition and service requirements?”.

A typical anticipatory networking solution is usually characterized by the following three attributes, which also determine the structure of the main part of this survey:

- **Context** defines the set of parameters considered to forecast the system evolution.
- **Prediction** specifies how the system evolution is obtained from the current and past context.
- **Optimization** describes how prediction is exploited to meet the application objective.

To continue with the access selection example, the anticipatory networking solution might exploit the history of Global Positioning System (GPS) information (the context) to train an AutoRegressive (AR) model (the prediction) to predict the future positions of the users and their channel conditions to solve an Integer Linear Programming (ILP) problem (the optimization) that maximizes their Quality-of-Experience (QoE).
The main body of the anticipatory networking literature can be split into four categories based on the context used to characterize the system state and to determine its evolution: geographic, such as human mobility patterns derived from location-based information; link, such as channel gain, noise and interference levels obtained from reference signal feedback; traffic, such as network load, throughput, and occupied physical resource blocks based on higher-layer performance indicators; social, such as user’s behavior, profile, and information derived from user-generated contents and social networks.

In order to determine which techniques are the most suitable to solve a given problem, it is important to analyze the following:

- **Properties** of the context:
  1) **Dimension** describes the number of variables predicted by the model, which can be uni- or multivariate.
  2) **Granularity and precision** define the smallest variation of the parameter considered by the context and the accuracy of the data: the lower the granularity, the higher the precision and vice versa. Temporal and spatial granularities are crucial to strike a balance between efficiency and accuracy.
  3) **Range** characterizes the distance between known data samples and the farthest predicted sample. It is also known as prediction (or optimization) horizon.

- **Constraints** of the prediction or optimization model:
  1) **Availability of physical model** states whether a closed-form expression exists to describe the phenomenon.
  2) **Linearity** expresses the quality of the functions linking inputs and outputs of a problem.
  3) **Side information** determines whether the main context can be related to auxiliary information.
  4) **Reliability and validity of information** specifies the noisiness of the data set, depending on which the prediction robustness should be calibrated.

The rest of the survey consists of a context-based classification of the anticipatory networking literature in Section III, two handbooks on prediction and optimization techniques in Section IV and Section V, respectively, and Section VI, which concludes the survey reporting the impact of anticipatory networking on future networks, the envisioned obstacles to its implementation and the next open challenges.

The classification section will help the reader to understand the link between the different contexts and the solutions adopted to satisfy the given application requirements. Also, it is meant to provide a complete panorama of anticipatory networking. The two handbooks have the twofold objective of providing the reader with a short overview of the tools adopted in the literature and to analyze them in terms of variables of interest and constraints of the models.

Table I provides a mapping between the techniques described in Section IV and V (rows) and the context discussed in Section III (columns). Each main category is further split into subcategories according to its internal structure. Namely, the prediction category is subdivided into ideal (perfect prediction is assumed to be available), time series predictive modeling, similarity-based classification and regression analysis, and probabilistic methods. The optimization category is split into heuristic, convex optimization (ConvOpt), Markov Decision Process (MDP) and Model Predictive Control (MPC), Heuristic and Direct (application of prediction) approaches.

### III. CONTEXT-BASED CLASSIFICATION

In this section, we show the different types of context that can be predicted and exploited. For each one, we highlight the most popular prediction techniques as well as the applications for which an anticipatory optimization is performed.

#### A. Geographic Context

Geographic context refers to the geographic area associated with a specific event or information. In wireless communications, it refers to the location of the mobile users, often enriched with speed information as well as past and future trajectories. Understanding human mobility is an emergent research field that especially in the last few years has significantly benefited from the rapid proliferation of wireless devices that frequently report status and location updates. Fig. 1 illustrates an example of estimated trajectories of 6 mobile users.

The potential predictability in user mobility can be as high as 93% [11]. Along the same line, [2] investigates both the maximal predictability and how close to this value practical algorithms can come when applied to a large mobile phone dataset. Those results indicate that human mobility is very far from being random. Therefore, collecting, predicting and exploiting geographic context is of crucial importance.

In the rest of this section we organize the papers dealing with geographic context according to their main focus: the

---

1. Value obtained for a high-income country with stable social conditions. The percentage can decrease for different countries, e.g., low-income country or natural disaster situation.
Collective behavioral patterns of similar users. In particular, [6] aims at predicting a given user location from coarse, but positioning can be obtained with little extra energy. Over time, the higher the accuracy of forecasting the rest.

Real-time prediction schemes are proposed and the feedback is used to improve the accuracy of prediction. A more general approach that exploits Artificial Neural Networks (ANNs) is discussed in [4]. Extreme Learning Machines (ELMs), which do not require any parameter tuning, are used to speed up the learning process. The method is evaluated using synthetic data and nodes are assumed to move at constant speed. Results show that the proposed method approaches an accuracy above 90% for a prediction horizon of some tens of seconds.

A more general approach that exploits Artificial Neural Networks (ANNs) is discussed in [4]. Extreme Learning Machines (ELMs), which do not require any parameter tuning, are used to speed up the learning process. The method is evaluated using synthetic data and different mobility models.

To extend the prediction horizon, [3] exploits users’ locations and short-term trajectories to predict the next handover. The authors use Channel State Information (CSI) and handover history to solve a classification problem via supervised learning, i.e., employing a multi-class Support Vector Machine (SVM). In particular, each classifier corresponds to a possible previous cell and predicts the next cell. A real-time prediction scheme is proposed and the feedback is used to improve the accuracy over time. Simulation results have been derived using both synthetic and real datasets. The longer a user went through a given path, the higher the accuracy of forecasting the rest.

Location information can be extracted from cellular network records. In this way the granularity of the prediction is more coarse, but positioning can be obtained with little extra energy. In particular, [3] aims at predicting a given user location from those of similar users. Collective behavioral patterns and a Markovian predictor are used to compute the next six locations of a user with a one-hour granularity, i.e., a six-hour prediction horizon. Evaluation is done using a real dataset and shows that an accuracy of about 70% can be achieved in the first hour, decreasing to 40 – 50% for the sixth hour of prediction.

2) Space and time prediction: Prediction of mobility in a combined space-time domain is often modeled using statistical methods. In [7], the idea is to predict not only the future location a user will reach, but also when and for how long the user will stay there. To incorporate the sojourn time during which a user remains in a certain location, mobility is modeled as a semi-Markov process. In particular, the transition probability matrix and the sojourn time distribution are derived from the previous association history. Evaluation is done on a real dataset and shows approximately 80% accuracy. A similar approach is presented in [8], where the prediction is extended from single to multi-transitions (estimating the likelihood of the future event after an arbitrary number of transitions). Both papers provide also some preliminary results on the benefits of the prediction on resource allocation and balancing.

In [9], the authors represent the network coverage and movements using graph theory. The user mobility is modeled using a Continuous Time Markov (CTM) process where the prediction of the next node to be visited depends not only on the current node but also on the previous one (i.e., second-order Markovian predictor). Considering both local as well as global users’ profiles, [10] extends the previous Markovian predictor and improves accuracy by about 30%. As pointed out in [11], sojourn times and transition probabilities are inhomogeneous. Thus, an inhomogeneous CTM process is exploited to predict user mobility. Evaluation on a real dataset shows an accuracy of 67% for long time scale prediction.

The interdependence between time and space is investigated also in [12] by examining real data collected from smartphones during a two month deployment. Furthermore, [13] shows the benefit of using a location-dependent Markov predictor with respect to a location-independent model based on nonlinear time series analysis. Additionally, it is shown that information on arrival times and periodicity of location visits is needed to provide accurate prediction. A system design, named SmartDC, is presented in [14], [15]. SmartDC comprises a mobility learner, a mobility predictor and an adaptive duty cycling. The system has been implemented and tested in a real environment. Notably, this is also one of the few papers that takes into account the cost of prediction, which in this case is evaluated in terms of energy. Namely, the authors detect approximately 90% of location changes, while reducing energy consumption at the expense of higher detection delay.

3) Location sequences and trajectories: A natural extension of the spatio-temporal perspective is the prediction of the location patterns and trajectories of the users. In [16], an approach for location prediction based on nonlinear time series analysis is presented. The framework focuses on the temporal predictability of users’ location, considering their arrival and residence times in relevant places. The evaluation is done considering four different real datasets. The authors evaluate first the predictability of the considered data and then show that the proposed nonlinear predictor outperforms both linear and Markov-based predictors. Precision approaches 70 – 90% for medium scale prediction (5 minutes) and decreases to
In order to improve the accuracy of time series techniques, in [17] the authors exploit the movement of friends, people, and, in general, entities, with correlated mobility patterns. By means of multivariate nonlinear time series prediction techniques, they show that forecasting accuracy approaches 95% for medium time scale prediction (5 to 10 minutes) and is approximately 50% for 3 hour prediction. Confidence bands show a significant improvement when prediction exploits patterns with high correlation. Evaluation is done considering two different real datasets.

Moving from discrete to continuous trajectories, Kalman filtering is used to predict the future velocity and moving trends of vehicles and to improve the performance of broadcasting [18]. The main idea is that each node should send the message to be broadcast to the fastest candidate based on its neighbors’ future mobility. Simulation results show modest gains, in terms of percentage of packet delivery and end-to-end delay, with respect to non-predictive methods.

An alternative to Kalman filters is the use of regression techniques [19], which analyze GPS observations of past trips. A systematic methodology, based on geometrical structures and data-mining techniques, is proposed to extract meaningful information for location patterns. This work characterizes the location patterns, i.e., the set of locations visited, for several millions of users using nationwide call data records. The analysis highlights statistical properties of the typical covered area and route, such as its size, average length and spatial correlation.

Along the same line, [20] shows how the regularity of driver’s behavior can be exploited to predict the current end-to-end route. The prediction is done by exploiting clustering techniques and is evaluated on a real dataset. A similar approach, named WhereNext, is proposed in [21]. This method predicts the next location of a moving object using past movement patterns that are based on both spatial and temporal information. The prediction is done by building a decision tree, whose nodes are the regions frequently visited. It is then used to predict the future location of a moving object. Results are shown using a real dataset provided by the GeoPKDD project [22]. The authors show the trade-off between the fraction of predicted trajectories and the accuracy. Both [20] and [21] show similar performance with an accuracy of approximately 40% and medium time scale prediction (order of minutes).

4) Dealing with errors: The impact of estimation and prediction errors is modeled in [23]. The authors propose a comprehensive overview of several mobility predictors and associated errors and investigate the main error sources and their impact on prediction. Based on this, they propose a stochastic model to predict user throughput that accounts for uncertainty. The method is evaluated using synthetic data while assuming that prediction’s errors have a truncated Gaussian distribution. Location errors are also considered in [24] where both temporal and spatial correlation are exploited to predict the average channel gain. The proposed method combines an AR model with functional linear regression and relies on location information. Results are derived using real data taken from the MOMENTUM project [25] and show that the proposed method outperforms SVM and AR processes.

5) Geographically-assisted video optimization: One of the main applications that has been used to show the benefits of geographic context is video streaming. A pioneer work showing the benefit of a long-term location-based scheduling for streaming is [26]. The authors propose a system for bandwidth prediction based on geographic location and past network conditions. Specifically, the streaming device can use a GPS-based bandwidth-lookup service in order to predict the expected bandwidth availability and to optimally schedule the video playout. The authors show simulation as well as experimental results, where the prediction is performed for the upcoming 100 meters. The predictive algorithm reduces the number of buffer underruns and provides stable video quality.

Application-layer video optimization based on prediction of user’s mobility and expected capacity, is proposed also in [27]–[29]. In [27], the authors minimize a utility function based on system utilization and rebuffering time. For the single user case they propose an online scheme based on partial knowledge, whereas the multiuser case is studied assuming complete future knowledge. In [28], different types of traffic are considered: full buffer, file download and buffered video. Prediction is assumed to be available and accurate over a limited time window. Three different utility functions are compared: maximization of the network throughput, maximization of the minimum user throughput, and minimization of the degradations of buffered video streams. Both works show results using synthetic data and assuming perfect prediction of the future wireless capacity variations over a time window with size ranging from tens to hundreds of seconds. In contrast, [29] introduces a data rate prediction mechanism that exploits mobility information and is used by an enhanced PF scheduler. The performance gain is evaluated using a real dataset and shows a throughput increase of 15%–55%.

Delay tolerant traffic can also benefit from offloading and prefetching as shown in [30]. The authors propose methods to minimize the data transfer over a mobile network by increasing the traffic offloaded to WiFi hotspots. Three different algorithms are proposed for both delay tolerant and delay sensitive traffic. They are evaluated using empirical measurements and assuming errors in the prediction. Results show that offloaded traffic is maximized when using prediction, even when this is affected by errors.

A geo-predictive streaming system called GTube, is presented in [31]. The application obtains the user’s GPS locations and informs a server which provides the expected connection quality for future locations. The streaming parameters are adjusted accordingly. In particular, two quality adaptation algorithms are presented, where the video quality level is adapted for the upcoming 1 and $n$ steps, respectively, based on the estimated bandwidth. The system is tested using a real dataset and shows that accuracy reaches almost 90% for very short time scale prediction (few seconds), but it decreases very fast approaching zero for medium time scale prediction (few minutes). However, the proposed $n$-step algorithm improves the stability of the video quality and increases bandwidth utilization.
B. Link Context

Link context refers to the prediction of the evolution of the physical wireless channel, i.e., the channel quality and its specific parameters, so that it is possible either to take advantage of future link improvements or to counter bad conditions before they impact the system. As an example, Fig. 2 shows a pathloss map of the center of Berlin realized with the data of the MOMENTUM project.

1) Channel parameter prediction: One possible approach to anticipate the evolution of the physical channel state is to predict the specific parameters that characterize it. In general, the variations of the physical channel can be caused by large-scale and small-scale fading. While predicting the latter is quite challenging, if not impossible, predicting the former, which includes pathloss and shadowing effects, can be feasible and has been the focus of several papers. In [32], the time-varying nonlinear wireless channel model is adopted to predict the channel quality variation anticipating distance and pathloss exponent. The performance evaluation is done using both an indoor and an outdoor testbed. The goodput obtained with the proposed bitrate control scheme can be almost doubled compared to other approaches.

Pathloss prediction in urban environments is investigated in [33]. The authors propose a two-step approach that combines machine learning and dimensional reduction techniques. Specifically, they propose a new model for generating the input vector, the dimension of which is reduced by applying linear and nonlinear principal component analysis. The reduced vector is then given to a trained learning machine. The authors compare ANNs and SVMs using real measurements and conclude that slightly better results can be achieved using the ANN regressors.

Supporting the temporal prediction with spatial information is proposed in, e.g., [34] to study the evolution of shadow fading. The authors suggest to implement a Kriged Kalman Filter (KKF) to track the time varying shadowing using a network of cognitive radios. The prediction is used to anticipate the position of the primary users and the expected interference and, consequently, to maximize the transmission rate of cognitive radio networks. Errors with the proposed model approach 2 dB (compared to 10 dB obtained with the pathloss based model). Targeting the same objective, but using a different methodology, [35] formulates the cognitive radio throughput optimization problem as an MDP. In particular, the predicted channel availability is used to maximize the throughput and to reduce the time overhead of channel sensing. Predictors robust to channel variations are investigated also in [36]. A clustering method with supervised SVM classification is proposed. The performance is shown for bulk data transport via Transmission Control Protocol (TCP) and it is also shown that the predictive approach outperforms non-predictive ones.

Finally, maps can be used to summarize predicted information; for instance, algorithms to build for pathloss map are proposed in [37]. In this paper, the authors propose two kernel-based adaptive algorithms, namely the adaptive projected subgradient method and the multikernel approach with adaptive model selection. Numerical evaluation is done for both a urban scenario and a campus network scenario, using real measurements. The performance of the algorithms is evaluated assuming perfect knowledge of the users’ trajectories.

2) Combined channel and mobility context: Channel quality and mobility information are jointly predicted in [38]. The authors combine information on visited locations and corresponding achieved link quality to provide connectivity forecast. A Markov model is implemented in order to forecast future channel conditions. Location prediction accuracy is approximately 70% for a prediction window of 20 seconds. However, the location information has quite a coarse granularity (of about 100 m). In terms of bandwidth, the proposed model, evaluated on a real dataset, shows an accuracy within 10 KB/s for over 50% of the evaluation period, and within 50 KB/s for over 80% of the time. In [39], prediction is employed to adjust the routing metrics in ad hoc wireless networks. In particular, the metrics considered in the paper are the average number of retransmissions needed and the time expected to transmit a data packet. The solution anticipates the future signal strength using linear regression on the history of the link quality measurements. Simulations show that the packet delivery ratio is close to 100%, even though it drops to 20% using classical methods.

When the information used to drive the prediction is affected by errors, it is important to account for the magnitude of the error as it is done in [40] for the impact of location uncertainties. The authors show that classical Gaussian Process (GP) wrongly predict the channel gain in presence of errors, while uncertain GP, which explicitly account for location uncertainty, outperforms the former in both learning and predicting the received power. Gains are shown also when using anticipation for predictive resource allocation. Uncertainties are also dealt with in [41], where a resource allocation algorithm for mobile networks that leverages link quality prediction is proposed. Time series filtering techniques (AutoRegressive and Moving Average (ARMA)) are used to predict near term link quality, whereas medium to long term prediction is based on statistical models. The authors propose a resource allocation optimization framework under imperfect prediction of future available capacity. Simulations are done using a real dataset and show that the proposed solution outperforms the limited horizon optimizer (i.e., when
the prediction is done only for the upcoming few seconds) by 10–15%. Resource allocation is also addressed in [29], which extends the standard Proportionally Fair (PF) scheduler of 4G networks to account for data rate prediction obtained through adaptive radio maps.

3) Channel-assisted video optimization: In [42], the authors propose an adaptive mobile video streaming framework, which stores video in the cloud and offers to each user a continuous video streaming adapted to the fluctuations of the link quality. The paper proposes a mechanism to predict the potential available bandwidth in the next time window (of a duration of a few seconds) based on the measurements of the link quality done in the previous time window. A prototype implementation of the proposed framework is used to evaluate the performance. This shows that the prediction has a relative error of about 10% for very short time windows (a couple of seconds) but becomes relatively poor for larger time windows. The video performance is evaluated in terms of “click-to-play” delay, which is halved with the proposed approach. A Markov model is used also in [43], where information on both channel and buffer states is combined to optimize mobile video streaming. Both an optimal policy as well as a fast heuristic are proposed. A drive test was conducted to evaluate the performance of the proposed solution. In particular, the authors show the proportional dependency between utility and buffer size, as well as the complexity of the two algorithms. Finally, a Markov model is adopted to represent different users’ achievable rates [44] and channel states [45]. The transition matrix is derived empirically to minimize the number of video stalls and their duration over a 10-second horizon.

4) Efficiency bounds and approximations for multimedia streaming applications: A few papers ([46]–[53]) investigate resource allocation optimization assuming that the future channel state is perfectly known. While addressing different objectives, these papers share similar methods: they first devise a problem formulation from which an optimal solution can be obtained (using standard optimization techniques), then they propose sub-optimal approaches and on-line algorithms to obtain an approximation of the optimal solution. Furthermore, all these papers leverage a buffer to counteract the randomness of the channel. For instance, in case a given amount of information has to be gathered within a deadline, the buffer allows the system to optimize (for a given objective function) the resource allocation while meeting the deadline.

In this regard, energy-efficiency is the primary objective in [46], [47], which is optimized by allowing the network base stations to be switched off once the users’ streaming requirements have been satisfied. Simulations show that an energy saving up to 80% with respect to the baseline approach can be achieved and that the performance of the heuristic solution is quite close to the optimal (but impractical) Mixed-Integer Linear Programming (MILP) approach. Buffer size is investigated in [52], where the author introduces a linear formulation that minimizes the amount for resources assigned to non-real time video streaming with constraints on the user’s playout buffer. Results are shown for a scenario with both video and best effort users and highlight the gain in terms of required resources to serve the video users as well as data rate for the best effort users.

The trade-off between streaming interruption time and average quality is investigated in [50], [51] by devising a mixed-integer quadratically constrained problem which computes the optimal download time and quality for video segments. Then, the authors propose a set of heuristics tailored to greedily optimize segment scheduling according to a specific objective function, e.g., maximum quality, minimum streaming interruption, or fairness. Similar objectives are tackled in [48], [49] in a lexicographic approach, so that streaming continuity is always prioritized over quality. They first propose a heuristic for the lateness-quality problem that performs almost as good as the MILP formulation. Then, they extend the MILP formulation to include Quality-of-Service (QoS) guarantees and they introduce an iterative approximation based on a simpler Linear Programming (LP) formulation. A further heuristic approach is devised in [53] and accounts for the buffer and channel state prediction. The proposed approach maximizes the streaming quality while guaranteeing that there are no interruptions.

C. Traffic Context

This section overviews some of the approaches that focus on traffic and throughput prediction. Although related to the previous context, the papers discussed in this section leverage information collected from higher layer of the protocol stack. For instance, solutions falling in this category try to predict, among other parameters, the number of active users in the network and the amount of traffic they are going to produce. Similarly, but from the perspective of a single user, the prediction can target the data rate that a streaming application is going to achieve in the near term.

We grouped these papers in three main classes: pure analysis of mobile traffic in Section III-C1, traffic prediction for networking optimization in Section III-C2 and direct throughput prediction in Section III-C3.

1) Traffic analysis and characterization: The analysis of mobile traffic is fundamental for long-term network optimization and re-configuration. To this end, several pieces of work have addressed such research topics in the recent past.

The work in [54] targets the creation of regressors for different performance indicators at different spatio-temporal granularity for mobile cellular networks. Namely, the authors focus on the characterization of per-device throughput, base station throughput and device mobility. A one-week nationwide cellular network dataset is collected through proprietary traffic inspection tools placed in the operator network and are used to characterize the per-user traffic, cell-aggregate traffic and to perform further spatio-temporal correlation analysis.

A similar scope is addressed by [55] which, on the other hand, focuses more on core network measurements. Flow level mobile device traffic data are collected from a cellular operator’s core network and are used to characterize the IP traffic patterns of mobile cellular devices.

More recently, the authors of [50] studied traffic prediction in cloud analytics and prove that optimizing the choice of metrics and parameters can lead to accurate prediction even under high latency. This prediction is exploited at the
application/TCP layer to improve the performance of the application avoiding buffer overflows and/or congestion.

2) Traffic prediction: Several applications can benefit from the prediction of traffic performance features. For instance, a predictive framework that anticipates the arrival of upcoming requests is used in [57] to prefetch the needed content at the mobile terminal. The authors propose a theoretical framework to assess how the outage probability scales with the prediction horizon. The theoretical framework accounts for prediction errors and multicast delivery.

Along the same line, queue modeling [58] and analysis [59] is used to predict the upcoming workloads in a lookahead time window. Leveraging the workload prediction, a multi-slot joint power control and scheduling problem is formulated to find the optimal assignment that minimizes the total cost [58] or maximizes the QoS [59].

Multimedia optimization is the focus in [60]. By predicting throughput, packet loss and transmission delay half a second in advance, the authors propose to dynamically adjust application-level parameters of the reference video streaming or video conferencing services including the compression ratio of the video codec, the forward error correction code rate and the size of the de-jittering buffer.

Traffic prediction is also addressed in [61], where the authors propose to use a database of events (concerts, gatherings, etc.) to improve the quality of the traffic prediction in case of unexpected traffic patterns and in [62], where a general predictive control framework along with Kalman filter is proposed to counteract the impact of network delay and packet loss.

The objective of [63] is to build a model for user engagement as a function of performance metrics in the context of video streaming services. The authors use a supervised learning approach based on average bitrate, join time, buffering ratio and buffering to estimate the user engagement. Finally, inter-download time can be modeled [64] and subsequently predicted for quality optimization.

3) Throughput prediction: Rather than predicting the expected traffic or optimizing the network based on traffic prediction, the work in this section targets the prediction/optimization based on the expected throughput. A common characteristic of the work described here is that the spatio-temporal correlation is exploited in the prediction phase of the expected throughput.

Quite a few early works studied how to effectively predict the obtainable data rate. In particular, long term prediction [65] with 12-hour granularity allows to estimate aggregate demands up to 6 months in advance. Shorter and variable time scales are studied in [66], [67] adopting AutoRegressive Integrated and Moving Average (ARIMA) and Generalized AutoRegressive Conditionally Heteroskedastic (GARCH) techniques.

In [68], the authors propose a dynamic framework to allocate downlink radio resources across multiple cells of 4G systems. The proposed framework leverages context information of three types: radio maps, user’s location and mobility, as well as application-related information. The authors assume that a forecast of this information is available and can be used to optimize the resource allocation in the network. The performance of the proposed solution is evaluated through simulation for the specific use case of video streaming.

Geo-localized radio maps are also exploited in [69]. Here the optimization is performed at the application layer by letting adaptive video streaming clients and servers dynamically change the streaming rate on the basis of the current bandwidth prediction from the bandwidth maps. The empirical collection of geo-localized data rate measures is also addressed in [70] which introduces a dataset of adaptive HTTP sessions performed by mobile users.

D. Social Context

The work on anticipatory networking leveraging social context exploits ex ante or ex post information on social-type relationships between agents in the networking environment. Such information may include: (1) the network of social ties and connections, (2) the user’s preference on contents, (3) measures on user’s centrality in a social network, and (4) measures on users’ mobility habits. The aforementioned context information is leveraged in three main application scenarios: (1) caching at the edge of mobile networks, (2) mobility prediction, (3) downlink resource allocation in mobile networks.

1) Social-assisted caching: Motivated by the need of limiting the load in the backhaul of 5G networks, references [71]–[73] propose two schemes to proactively move contents closer to the end users; in the first [71], caching happens at the small cells, whereas in the latter schemes [72], [73] contents can be proactively downloaded by a subset of end users which then re-distribute them via device-to-device communication. The authors first define two optimization problems which target the load reduction in the backhaul (caching at small cells) and in the small cell (caching at end users), respectively, then heuristic algorithms based on machine learning tools are proposed to obtain sub-optimal solutions in reasonable processing time. The heuristic first collects users’ content rating/preferences to predict the popularity matrix $P_{m}$. Then, content is placed at each small cell in a greedy way starting from the most popular ones until a storage budget is hit. The first algorithmic step of caching at the end users is to identify the $K$ most connected users and to cluster the remaining users in communities. Then it is possible to characterize the content preference distributions within each community and greedily place contents at the cluster heads. In [73], the prediction leverages additional information on the underlying structure of content popularity within the communities of users.

A similar problem is addressed in [74]: the authors consider a distributed network of femto base stations, which can be leveraged to cache videos. The authors study where to cache videos such that the average sum delay across all the end users is minimized for a given video content popularity distribution, a given storage capacity and an arbitrary model for the wireless link. A greedy heuristic is then proposed to reduce the computational complexity.

2) Social-assisted mobility prediction: Motivated by the need to reduce the active scanning overhead in IEEE 802.11 networks, the authors of [75] propose a mobility prediction
tool to anticipate the next access point a WiFi user is moving to. The proposed solution is based on context information on the handoffs which were performed in the past; specifically, the system stores centrally a time varying handoff table which is then fed into an ARIMA predictor which returns the likelihood of a given user to hand-off to a specific access point. The quality of the predictor is measured in terms of signaling reduction due to active scanning.

The prediction of user mobility is also addressed in [76]. The authors leverage information coming from the social platform Foursquare to predict user mobility on coarse granularity. The next check-in problem is formulated to determine the next place in an urban environment which will be most likely visited by a user. The authors build a time-stamped dataset of “check-ins” performed by Foursquare users over a period of one month across several venues worldwide. A set of features is then defined to represent user mobility including user mobility features (e.g., number of historical visits to specific venues or categories of venues, number of historical visits that friends have done to specific venues), global mobility features (e.g., popularity of venues, distance between venues, transition frequency between couples of venues), and temporal features which measures the historical check-ins over specific time periods. Such a feature set is then used to train a supervised classification problem to predict the next check-in venue. Linear regression and M5 decision trees are used in this regard. The work is mostly speculative and does not address directly any specific application/use of the proposed mobility prediction tool.

Along the same lines, the mobility of users in urban environments is characterized in [77]. Different from the previous work which only exploits social information, the authors also leverage physical information about the current position of moving users. A probabilistic model of the mobile users' behavior is built and trained on a real life dataset of user mobility traces. A social-assisted mobility prediction model is proposed in [78], where a variable-order Markov model is developed and trained on both temporal features (i.e., when users were at specific locations) and social ones (i.e., when friends of specific users were at a given location). The accuracy of the proposed model is cross-validated on two user-mobility datasets.

3) **Social-assisted radio resource allocation:** The optimization of elastic traffic in the downlink of mobile radio networks is addressed in [79], [80]. The key tenet is to provide to the downlink scheduler “richer” context to make better decisions in the allocation of the radio resources. Besides classical network-side context including the cell load and the current channel quality indicator which are widely used in the literature to steer the scheduling, the authors propose to include user-side features which generically capture the satisfaction degree of the user for the reference application: namely, the authors introduce the concept of a transaction, which represents the atomic data download requested by the end user (e.g., a web page download via HTTP, an object download via HTTP or a file download via FTP); for each transaction and for each application, a utility function is defined capturing the user’s sensitivity with respect to the transmission delay and the expected completion time. The functional form of this utility function depends on the type of application which “generated” the transaction; as an example, the authors make the distinction between transactions from applications which are running in the foreground and the background on the user’s terminal. For the sake of presentation, a parametric logistic function is used to represent the aforementioned utility. The authors then formulate an optimization problem to maximize the sum utility across all the users and transactions in a given mobile radio cell and design a greedy heuristic to obtain a sub-optimal solution in reasonable computing time. The proposed algorithm is validated against state-of-the-art scheduling solutions (PF / weighted PF scheduling) through simulation on synthetic data mimicking realistic user distributions, mobility patterns and traffic patterns.

Social-oriented techniques related to the popularity of the end users are leveraged also in [81] where the authors target the performance optimization of downlink resource allocation in future generation networks. The utility maximization problem is formulated with the utility being a combination (product) of a network-oriented term (available bandwidth) and a social-oriented term (social distance). The social-oriented term is defined to be the degree centrality measure [82] for a specific user. The proposed problem is sub-optimally solved through a heuristic which is finally validated using synthetic data.

### E. Takeaways

Hereafter, we summarize the main takeaways of the section in terms of application and objective for which different context types can be used. Table 1 provides a synthesis of the main considerations: each context is associated with its typical applications, prediction methodologies (ordered by decreasing popularity), optimization approaches and general remarks.

1) **Mobility prediction:** It has been shown that predictability of user mobility can be potentially very high. As a matter of fact, many papers study how to forecast users’ mobility by means of a variety of techniques. The most popular ones leverage geographic information: GPS data, cell records and received signal strength are used to obtain precise and frequent data sampling to locate users on a map. However, the movements of an individual are largely influenced by those of other individuals via social relations. Several papers analyze social information and location check-ins to find recurrent patterns. For this second case usually a sparser dataset is available and may limit the accuracy of the prediction.

2) **Network efficiency:** Predicting and optimizing network efficiency (i.e., increasing the performance of the network while using the same amount of resources) is the most frequent objective in anticipatory networking. We found papers exploiting all four types of context to achieve this. As such, objectives and constraints cover the whole attribute space. Improving network efficiency is likely to become the main driver for including anticipatory networking solutions in next generation networks.

3) **Multimedia streaming:** The main source of data traffic in 4G networks has been multimedia streaming and, in particular, video on demand. 5G networks are expected to continue and
even increase this trend. As a consequence, several anticipatory networking solutions focus on the optimization of this service. All the context types have been used to this extent and each has a different merit: social information is needed to predict when a given user is going to request a given content, combined geographic and social information allows the network to cache that content closer to where it will be required and physical channel information can be used to optimize the resource assignment.

4) Network offloading: Mobility prediction can be used to handover communications between different technologies to decrease network congestion, improve user experience, reduce users’ costs and increase efficiency.

5) Cognitive networking: Physical channel prediction can be exploited for cognitive networking and for network mapping. The former application allows secondary users to access a shared medium when primary subscribers left resource unused, thus, predicting when this is going to happen will highly improve the effectiveness of the solution. The latter, instead, exploits link information to build networking maps that can provide other applications with an estimate of communication quality at a given time and place.

6) Throughput- and traffic-based applications: Traffic information is usually studied to be, first, modeled and, subsequently, predicted. Traffic models and predictors are then used to improve networking efficiency by means of resource allocation, traffic shaping and network planning.

### IV. Prediction Methodologies for Anticipatory Networking

In this section, we present some selected prediction methods for the types of context introduced in Section II. The selected methods are classified into four main categories: time series methods, similarity-based classification, regression analysis, and statistical methods for probabilistic modeling. Their mathematical principles and the application to inferring and predicting the aforementioned contextual information are introduced in Sections IV-A, IV-B, IV-C, and IV-D respectively.

The goal of the prediction handbook is to show which methods work in which situation. In fact, selecting the appropriate prediction method requires to analyze the prediction variables and the model constraints with respect to the application scenario (see Section II). This section concludes with a series of takeaways that summarize some general principles for selection of prediction methods based on the scenario analysis.

#### A. Time Series Predictive Modeling

A time series is a set of time-stamped data entries which allows a natural association of data collected on a regular or irregular time basis. In wireless networks, large volumes of data are stored as time series and frequently show temporal correlation. For example, the trajectory of the mobile device can be characterized by successive time-stamped locations obtained from geographical measurements; individual social behavior can be expressed through time-evolving events; instead traffic loads modeled in time series can be leveraged for network planning and controlling. Fig. 3(a) and 3(b) illustrate two time series of per-cell and per-city aggregated uplink and downlink data traffic, where temporal correlation is clearly recognizable.

In the following, we introduce the two most widely used time series models based on linear dynamic systems: 1) AutoRegressive and Moving Average (ARMA), and 2) Kalman

### TABLE II: CONTEXT CLASSIFICATION SUMMARY

| Context | Applications | Prediction | Optimization | Remarks |
|---------|--------------|------------|--------------|---------|
| Geographic | Mobility prediction | First Probabilistic | 1) Prediction to define convex optimization problems | 1) Prediction accuracy is inversely proportional to the time scale and granularity |
|         | Multimedia streaming | 2nd Regression | 2) Prediction as the optimization objective | 2) High prediction accuracy can be obtained on long time scales if periodicity and/or trends are present |
|         | Broadcast | 3rd Time series | 3) Prediction is more effectively used in delay tolerant applications | 3) Prediction is more effectively used in delay tolerant applications |
|         | Resource allocation | 4th Classification | | |
|         | Duty cycling | | | |
| Physical | Channel forecast | First Regression | 1) Markov decision process is used when statistical knowledge of the system is available | 1) Channel quality maps can be effectively used to improve networking |
|         | Resource allocation | 2nd Time series | 2) Convex optimization is preferred when it is possible to perform accurate forecast | 2) Movement dynamics affect the prediction effectiveness |
|         | Network mapping | 3rd Probabilistic | | 3) Channel is most often predicted by means of functional regression or Markovian models |
|         | Routing | 4th Classification | | |
|         | Multimedia streaming | | | |
| Traffic | Traffic analysis | First Regression | 1) Maps are used to deterministically drive the optimization | 1) Improved long-term network optimization and reconfiguration |
|         | Resource allocation | 2nd Classification | 2) Convex optimization problems can be formulated to obtain bounds | 2) Traffic distribution is skewed both with regards to users and locations |
|         | Multimedia streaming | 3rd Probabilistic | | 3) Traffic has a strong time periodicity |
| Social | Network caching | First Classification | 1) Formal optimization problems can be defined, but they are usually impractical to be solved | 1) A fraction of social information can be accurately predicted |
|         | Mobility prediction | 2nd Regression | 2) Heuristics are the preferable solutions to obtain online algorithms | 2) Prediction obtained from social information is usually coarse |
|         | Resource allocation | 3rd Time series | | 3) Social information prediction can effectively improve application performance |
|         | Multimedia streaming | 4th Probabilistic | | |

*Ranking based on the number of papers reviewed in this survey using the predictor.*
Numerous studies have been done on prediction of traffic load in wireless or IP backbone networks using autoregressive models. The stationarity analysis often provides important clues for selecting the appropriate model. For instance, in [65], a low-order ARIMA model is applied to capture the non-stationary short memory process of traffic load, while in [66], a Gegenbauer ARMA model is used to specify long memory processes under the assumption of stationarity. Similar models are applied to mobility- or channel-related contexts. In [75], an exponential weighted moving average, equivalent to ARIMA(0, 1, 1), is used to forecast handoffs. In [3], [32], AR models are applied to predict future signal-to-noise ratio values and user positions, respectively. If the variance off the data varies with time, as in [67] for data traffic, and can be expressed using an ARMA, then the whole model is referred to as GARCH.

2) Kalman filter: Kalman filters are widely applied in time series analysis for linear dynamic systems, which track the estimated system state and its uncertainty variance. In the anticipatory networking literature, Kalman filters have been mainly adopted to model the linear dependence of the system states based on the historical data.

Consider a multivariate time series \( \{x_t \in \mathbb{R}^n : t \in T \} \), the Kalman filter addresses the problem of estimating state \( x_t \) that is governed by the linear stochastic difference equation

\[
x_t = A_t x_{t-1} + B_t u_t + w_t, \quad t = 0, 1, \ldots,
\]

where \( A_t \in \mathbb{R}^{n \times n} \) expresses the state transition, and \( B_t \in \mathbb{R}^{n \times l} \) relates the optional control input \( u_t \in \mathbb{R}^l \) to the state \( x_t \in \mathbb{R}^n \). The random variable \( w_t \sim \mathcal{N}(0, Q_t) \) represents a multivariate normal noise process with covariance matrix \( Q_t \in \mathbb{R}^{n \times n} \). The observation \( z_t \in \mathbb{R}^m \) of the true state \( x_t \) is given by

\[
z_t = H_t x_t + v_t,
\]

where \( H_t \in \mathbb{R}^{m \times n} \) maps the true state space into the observed space. The random variable \( v_t \) is the observation noise process following \( v_t \sim \mathcal{N}(0, R_t) \) with covariance \( R_t \in \mathbb{R}^{m \times m} \). Kalman filters iterate between 1) predicting the system state with Eq. (4) and 2) updating the model according to Eq. (5) to refine the previous prediction. The interested reader is referred to [34] for more details.

In [18], [85], Kalman filters are used to study users’ mobility. Wireless channel gains are studied in [34] with KKF, while the authors of [86] adopt the technique to predict short-term traffic volume. The extended Kalman filter adapts the standard model to nonlinear systems via online Taylor expansion. According to [87], this improves shadow/fading estimation.

### B. Similarity-based Classification

Similarity-based classification aims to find inherent structures within a dataset. The core rationale is that similarity patterns in a dataset can be used to predict unknown data or missing features. Recommendation systems are a typical application where users give a score to items and the system
tries to infer similarities among users and scores to predict the missing entries.

These techniques are unsupervised learning methods, since categories are not predetermined, but are inferred from the data. They are applied to datasets exhibiting one or more of the following properties: 1) entries of the dataset have many attributes, 2) no law is known to link the different features, and 3) no classification is available to manually label the dataset.

In what follows, we briefly review the similarity-based classification tools that have been used in the anticipatory networking literature accounted for in this survey.

1) Collaborative filtering: Recommendation systems usually adopt Collaborative Filtering (CF) to predict unknown opinions according to user’s and/or content’s similarities. While a thorough survey is available in [88], here, we just introduce the main concepts related to anticipatory networking.

CF predicts the missing entries of a \( n_c \times n_u \) matrix \( Y \in \mathcal{A}^{n_c \times n_u} \), mapping \( n_c \) users to \( n_u \) contents through their opinions which are taken from an alphabet \( \mathcal{A} \) of possible ratings; thus the entry \( y_{ik} \), \( i \in \{1, \ldots, n_c\}, k \in \{1, \ldots, n_u\} \) expresses how much user \( k \) likes content \( i \). An auxiliary matrix \( \mathbf{R} \in [0,1]^{n_c \times n_u} \) expresses whether user \( k \) evaluated content \( i \) \((r_{ik} = 1)\) or not \((r_{ik} = 0)\).

To predict the missing entries of \( Y \) the feature learning approach exploits a set of \( n_f \) features to represent contents’ and users’ similarities and defines two matrices \( \mathbf{X} \in [0,1]^{n_u \times n_f} \) and \( \Theta \in \mathcal{A}^{n_c \times n_f} \), whose entries \( x_{ij} \) and \( \theta_{kj} \) represent how much content \( i \) is represented by feature \( j \) and how high user \( k \) would rate a completely content defined by feature \( j \), respectively. The new matrices aim to map \( Y \) in the feature space and they can be computed by:

\[
\arg\min_{\mathbf{X},\Theta} \sum_{i,k:r_{ik}=1} (x_{is}\theta_{ks}^T-y_{ik})^2, \tag{6}
\]

where \( x_{is} := (\text{col}_i\mathbf{X}^T)^T \) denote the \( i\)-th row of matrix \( \mathbf{X} \). Note that in (6) the regularization terms are omitted. Solving (6) amounts to obtain a matrix \( \hat{Y} = \mathbf{X}\Theta^T \) which best approximates \( Y \) according to the available information \((i,k:r_{ik} = 1)\). Finally, \( \hat{y}_{ik} = x_{is}\theta_{ ks } \) predicts how user \( k \) with parameters \( \theta_{ks} \) rates content \( i \) having feature vector \( x_{is} \).

Other applications of CF are, for instance, network caching optimization [89], [90], where communication efficiency is optimized by storing contents where and when they are predicted to be consumed. Similarly, location-based services [76] predict where and what to serve to a given user.

2) Clustering: Clustering techniques are meant to group elements that share similar characteristics. The following provides an introduction to \( K \)-means, which is among the most commonly-used clustering techniques in anticipatory networking. The interested reader is referred to [91] for a complete review.

\( K \)-means splits a given dataset into \( K \) groups without any prior information about the group structure. The basic idea is to associate each observation point from a dataset \( \mathcal{X} := \{ \mathbf{x}_i \in \mathbb{R}^n : i = 1, \ldots, M \} \), to one of the centroids in set \( \mathcal{M} := \{ \mathbf{\mu}_j \in \mathbb{R}^n : j = 1, \ldots, K \} \). The centroids are optimized by minimizing the intra-cluster sum of squares (sum of distance of each point in the cluster to the \( K \) centroids), given by

\[
\minimize_{C,M} \sum_{j=1}^{K} \sum_{i=1}^{M} c_{ij} \| x_i - \mathbf{\mu}_j \|^2, \tag{7}
\]

where \( C := \{ c_{ij} \in \{0,1\} : i = 1, \ldots, M, j = 1, \ldots, K \} \) associates entry \( x_i \) to centroid \( \mathbf{\mu}_j \). No entry can be associated to multiple centroids \((\sum_{j=1}^{K} c_{ij} = 1, \forall i)\).

Clustering is applied in anticipatory networking to build a data-driven link model [56], to find similarities within vehicular paths [20], to identify social events [61] that might impact network performance, and to identify device types [55].

3) Decision Trees: A supervised version of clustering is decision tree learning (the interested reader is referred to [92] for a survey on the topic). Assuming that each input observation is mapped to a consequence on its target value (such as reward, utility, cost, etc.), the goal of decision tree learning is to build a set of rules to map the observations to their target values. Each decision branchs the tree into different paths that lead to leaves representing the class labels of the conclusions. With prior knowledge, decision trees can be exploited for location-based services [76], for identifying trajectory similarities [21], and for predicting the QoE for multimedia streams [63]. For continuous target variables, regression trees can be used to learn trends in network performance [60].

C. Regression Analysis

When the interest lies in understanding the relationship between different variables, regression analysis is used to predict dependent variables from a number of independent variables by means of so-called regression functions. In the following we introduce three regression techniques which are able to capture complex nonlinear relationships, namely functional regression, support vector machine and artificial neural networks.

1) Functional regression: Functional data often arises from measurements, where each point is expressed as a function over a physical continuum (e.g., Fig. 4 illustrates the example of aggregated WiFi traffic as a function of the hour of the day). Functional regression has two interesting properties: smoothness allows to study derivatives, which may reveal important aspects of the processes generating the data, and the mapping between original data and the functional space
may reduce the dimensionality of the problem and, as a consequence, the computational complexity [93]. The commonly encountered form of function prediction regression model (scalar-on-function) is given by [94]:

\[ Y_i = B_0 + \int X_i(z)B(z)dz + E_i \]  

where \( Y_i, i = 1, \ldots, M \) is a continuous response, \( X_i(z) \) is a functional predictor over the variable \( z \), \( B(z) \) is the functional coefficient, \( B_0 \) is the intercept, and \( E_i \) is the residual error.

Functional regression methods are applied in [50] to predict traffic-related Long Term Evolution (LTE) metrics (e.g., throughput, modulation and coding scheme, and used resources) showing that cloud analytics of short-term LTE metrics is feasible. In [55], functional regression is used to study churn rate of mobile subscribers to maximize the carrier profitability.

2) Support vector machines: SVM is a supervised learning technique that constructs a hyperplane or set of hyperplanes (linear or nonlinear) in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. In this survey we introduce the SVM for classification, and the same principle is used by SVM for regression. Consider a training dataset \( \{(x_i, y_i) : x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i = 1, \ldots, M\} \), where \( x_i \) is the \( i \)-th training vector and \( y_i \) the label of its class. Let us assume that the data is linearly separable first and define the linear separating hyperplane as \( w \cdot x - b = 0 \), where \( w \cdot x \) is the Euclidean inner product.

The optimal hyperplanes is the one that maximizes the margin (distance from the hyperplane to the instances closest to it on either side), found by solving the following optimization problem:

\[ \text{minimize} \quad \frac{1}{2} \|w\|^2 \]
\[ \text{subject to} \quad y_i(w \cdot x_i + b) - 1 \geq 0 \quad \forall i. \]  

Fig. 5(a) shows an example of linear SVM classifier separating two classes in \( \mathbb{R}^2 \).

If the data is not linearly separable, the training points are projected to a high-dimensional space \( \mathcal{H} \) through a nonlinear transformation \( \phi : \mathbb{R}^n \rightarrow \mathcal{H} \). Then, a linear model in the new space is built, which corresponds to a nonlinear model in the original space. Since the solution of (9) consists of inner products of training data \( x_i \cdot x_j \) for all \( i, j \), in the new space the solution is in the form of \( \phi(x_i) \cdot \phi(x_j) \). The kernel trick is applied to replace the inner product of basis functions by a kernel function \( K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \), between instances in the original input space, without explicitly building the transformation \( \phi \).

The Gaussian kernel \( K(x, y) := \exp(\gamma \|x - y\|^2) \) is one of the most widely used kernels in the literature. For example, it is used in [5] to predict user mobility. Nevertheless, choosing an appropriate kernel for a given prediction task remains one of the main challenges. In [37], the authors propose an algorithm for reconstructing coverage maps from path-loss measurements using a kernel method.

3) Artificial neural networks: ANN is a supervised machine learning solution for both regression and classification. An ANN is a network of nodes, or neurons, grouped into three layers (input, hidden and output), which allows for nonlinear classification. It can ideally achieve zero training error.

Consider a training dataset \( \{(x_i, y_i) : x_i \in \mathbb{R}^n, i = 1, \ldots, M\} \). Each hidden node \( h_l \) approximates a so called logistic function in the form \( h_l = 1/(1 + \exp(-\omega_l \cdot x)) \), where \( \omega_l \) is a weight vector. The outputs of the hidden nodes are processed by the output nodes to approximate \( y \). These nodes use linear and logistic functions for regression and classification, respectively. In the linear case, the approximated output is represented as:

\[ \hat{y} = \sum_{l=1}^{L} h_l v_l = \sum_{l=1}^{L} \frac{1}{1 + \exp(-\omega_l \cdot x)} v_l, \]  

where \( L \) is the number of hidden nodes and \( v_l \) is the weight vector of the output layer. The training of an ANN can be performed by means of the backpropagation method that finds weights for both layers to minimize the mean squared error between the training labels \( y \) and their approximations \( \hat{y} \). In the anticipatory networking literature, ANNs have been used for example to predict mobility in mobile ad-hoc networks [4], [96].

For both SVMs and ANNs, as for other supervised learning approaches, no prior knowledge about the system is required but a large training set has to be acquired for parameter setting in the predictive model. A careful analysis needs to be performed while processing the training data in order to avoid both overfitting and underlearning.

D. Statistical Methods for Probabilistic Forecasting

Probabilistic forecasting involves the use of information at hand to make statements about the likely course of future events. In the following subsections, we introduce two probabilistic forecasting techniques: Markovian models and Bayesian inference.

1) Markovian models: These models can be applied to any system for which state transitions only depend on the current state. In the following we briefly discuss the basic concepts of discrete, and continuous time Markov Chains (MCs) and their respective applications to anticipatory networking.

A Discrete Time Markov Chain (DTMC) is a discrete time stochastic process \( X_n (n \in \mathbb{N}) \), where a state \( X_n \) takes a finite number of values from a set \( \mathcal{X} \) in each time slot. The
Markovian property for a DTMC transitioning from any time slot $k$ to $k+1$ is expressed as follows:

$$P(X_{k+1} = j | X_k = i) = p_{ij}(k).$$ (11)

For a stationary DTMC, the subscript $k$ is omitted and the transition matrix $P$, where $p_{ij}$ represents the transition probability from state $i$ to state $j$, completely describes the model.

A great deal of empirical measurements on mobility and traffic evolution can be accurately predicted using a DTMC with low computational complexity [9], [13], [15], [55], [78].

However, obtaining the transition probabilities of the system requires a variable training period, which depends on the prediction goal. In practice, the data collection period can be in the order of one [55] or even multiple weeks [10], [38].

A DTMC assumes the time the system spends in each state is equal for all states. This time depends on the prediction application and can range from a few hundred milliseconds to predict wireless channel quality [55], to tens of seconds for user mobility prediction [9], [38], to hours for Internet traffic [55]. For tractability reason, the state space is often compressed by means of simple heuristics [10], [38], [64], $K$-means clustering [45], [78], equal probability classification [64], and density-based clustering [78].

Eq. (11) defines a first order DTMC and can be extended to the $l$-th order (i.e., transition probabilities depend on the $l$ previous states). Using higher order DTMCs can increase the accuracy of the prediction at the expense of a longer training time and an increased computational complexity [9], [13], [78].

If the sojourn time of each state is important, the system can be modeled as a Continuous Time Markov Chain (CTMC). The Markovian property is preserved in CTMC when the sojourn time is exponentially distributed, as in [11]. When the sojourn time has an arbitrary distribution, the process becomes a Markov renewal process as in [7], [8].

If the transition probabilities cannot be directly measured, but only the output of the system is quantifiable (dependent on the state), hidden Markov models allow to map the output state space to the unobservable model that governs the system. As an example, the inter-download times of video segments are predicted in [64], where the output sequences are the inter-download times of the already downloaded segments and the states are the instants of the next download request.

2) Bayesian inference and prediction: This approach allows to make statements about what is unknown, by conditioning on what is known. Bayesian prediction can be summarized in the following steps: 1) define a model that expresses qualitative aspects of our knowledge but has unknown parameters, 2) specify a prior probability distribution for the unknown parameters, 3) compute the posterior probability distribution for the parameters, given the observed data, and 4) make predictions by averaging over the posterior distribution.

Given a set of observed data $D := \{ (x_i, y_i) : i = 1, \ldots, M \}$ consisting of a set of input samples $X := \{ x_i \in \mathbb{R}^p : i = 1, \ldots, M \}$ and a set of output samples $Y := \{ y_i \in \mathbb{R}^q : i = 1, \ldots, M \}$, inference in Bayesian models is based on the posterior distribution over the parameters, given by the Bayes’ rule:

$$p(\theta|D) = \frac{p(Y|X, \theta)p(\theta)}{p(Y|X)} \propto p(Y|X, \theta)p(\theta),$$ (12)

where $\theta$ is the unknown parameter vector.

Two recent works adopting the Bayesian framework are [40] and [24]. The former focuses on spatial prediction of the wireless channel, building a 2D non-stationary random field accounting for pathloss, shadowing and multipath, while the latter exploits spatial and temporal correlation to develop a general prediction model for the channel gain of mobile users.

E. Takeaways

Hereafter, we provide some guidelines for selecting the appropriate prediction methods depending on the application scenario or context of interest.

1) Applications and data: The predicted context is the most important information that drives decision making in anticipatory optimization problems (see Section V). Thus, the selection of the prediction method shall take into consideration the objectives of the application and the constraints imposed by the available data.

a) Choosing the outputs: Applications define the properties of the predicted variables, such as dimension, granularity, accuracy, and range. For example, large granularity or high data aggregation (such as frequently visited location, social behavior pattern) is best dealt with similarity-based classification methods which provide sufficiently accurate prediction without the complexity of other model-based regression techniques.

b) System model and data: The application environment is equally important as its outputs, which determines the constraints of modeling. Often, an accurate analysis of the scenario might highlight linearity, deterministic and/or causal laws among the variables that can further improve the prediction accuracy. Moreover, the quality of dataset heavily affects the prediction accuracy. Different methods exhibit different level of robustness to noisy data.

2) Guidelines for selecting methods: To choose the correct tool among the aforementioned set, we study the rationale for adopting each of them in the literature and derive the following practical guidelines.

a) Model-based methods: When physical models exists, model-based regression techniques based on closed-form expressions can be used to obtain an accurate prediction. They are usually preferable for long-term forecast and exhibit good resilience to poor data quality.

b) Time series: These are the most convenient tools when the information is abundant and shows strong temporal correlation. Under these conditions, time series methods provide simple means to obtain multiple scale prediction of moderate to high precision.

c) Causal methods: If the data exhibits large and fast variations, causality laws can be key to obtain robust predictions. In particular, if a causal relationship can be observed between the variables of interest and the other observable data, causal models usually outperform pure data-driven models.
V. Optimization Methodologies for Anticipatory Networking

This section identifies the main optimization techniques adopted by anticipatory networking solutions to achieve their objectives. Disregarding the particular domain of each work, the common denominator is to leverage some future knowledge obtained by means of prediction to drive the system optimization. How this optimization is performed depends both on the ultimate objectives and how data are predicted and stored.

In general, we found two main strategies for optimization: (1) adopting a well-known optimization framework to model the problem and (2) designing a novel solution (most often) based on heuristic considerations about the problem. The two strategies are not mutually exclusive and often, when known approaches lead to too complex or impractical solutions, they are mixed in order to provide feasible approximation of the original problem.

Heuristic approaches usually consist of (1) algorithms that allow for fast computation of an approximation of the solution of a more complex problem (e.g., convex optimization) and (2) greedy approaches that can be proven optimal under some set of assumptions. Both approaches trade optimality for complexity and most often are able to obtain performance quite close to the optimal one. However, heuristic approaches are tailored to the specific application and are usually difficult to be generalized or to be adapted for different scenarios, thus they cannot be directly applied to new applications if the new requirements do not match those of the original scenario.

In what follows, we focus on optimization methods only and will provide some introductory descriptions of the most relevant ones, that are used for anticipatory networking. The objective is to provide the reader with a minimum set of tools to understand the methodologies and to highlight the main properties and applications.

A. Convex Optimization

Convex optimization is a field that studies the problem of minimizing a convex function over convex sets. The interested reader can refer to [97] for convex optimization theory and algorithms. Hereafter, we will adopt Boyd’s notation [97] to introduce definitions and formulations that frequently appear in anticipatory networking papers.

The inputs are often referred to as the optimization variables of the problem and defined as the vector \( x = (x_1, \ldots, x_n) \). In order to compute the best configuration or, more precisely, to optimize the variables, an objective is defined: this usually corresponds to minimizing a function of the optimization variables, \( f_0 : \mathbb{R}^n \rightarrow \mathbb{R} \). The feasible set of input configurations is usually defined through a set of \( m \) constraints \( f_i(x) \leq b_i, \ i = 1, \ldots, m, \) with \( f_i : \mathbb{R}^n \rightarrow \mathbb{R} \). The general formulation of the problem is

\[
\begin{align*}
\text{minimize} \ & f_0(x) \\
\text{subject to} \ & f_i(x) \leq b_i, \ i = 1, \ldots, m.
\end{align*}
\] (13)

The solution to the optimization problem is an optimal vector \( x^* \) that provides the smallest value of the objective function, while satisfying all the constraints.
The convexity property (i.e., objective and constraint functions satisfy \( f_i(ax + (1-a)y) \leq af_i(x) + (1-a)f_i(y) \) for all \( x, y \in \mathbb{R}^n \) and \( a \in [0,1] \)) can be exploited in order to derive efficient algorithms that allow for fast computation of the optimal solution. Furthermore, if the optimization function and the constraints are linear, i.e., \( f_i(ax + by) = af_i(x) + bf_i(y) \) for all \( x, y \in \mathbb{R}^n \) and \( a, b \in \mathbb{R} \), the problem belongs to the class of \textit{linear optimization}. For this class, highly efficient solvers exist, thanks to their inherently simple structure. Within the linear optimization class, three subclasses are of particular interest for anticipatory networking: least-squares problems, linear programs and mixed-integer linear programs.

\textit{Least-squares} problems can be thought of as distance minimization problems. They have no constraints \((m = 0)\) and their general formulation is:

\[
\text{minimize } f_0(x) = \|Ax - b\|^2_2, \quad (14)
\]

where \( A \in \mathbb{R}^{k \times n} \) with \( k \geq n \) and \( \|x\|_2 \) is the Euclidean norm. Notably, problems of this class have an analytical solution \( x = (A^T A)^{-1} A^T b \) (where superscript \( T \) denotes the transpose) derived from reducing the problem to the set of linear equations \( A^T Ax = A^T b \).

\textit{Linear programming} (LP) problems are characterized by linear objective function and constraints and are written as:

\[
\text{minimize } c^T x \quad \text{subject to } \quad Ax \leq b, \quad (15)
\]

where \( c \in \mathbb{R}^n \), \( A \in \mathbb{R}^{m \times n} \) and \( b \in \mathbb{R}^m \) are the parameters of the problem. Although, there is no closed-form analytical solution to LP problems, a variety of efficient algorithms are available to compute the optimal vector \( x^* \). When the optimization variable is a vector of integers \( x \in \mathbb{Z}^n \), the class of problems is called \textit{integer linear programming} (ILP), while the class of \textit{mixed-integers linear programming} (MILP) allows for both integer and real variables to co-exist. These last two classes of problems can be shown to be NP-hard (while LP is P complete) and their solution often implies combinatorial aspects. See [98] for more details on integer optimization.

In anticipatory networking, we find that resource allocation problems are often modeled as LP, ILP or MILP, by setting the amount of resources to be allocated as the optimization variable and accounting for prediction in the constraints of the problem. In [46], prediction of the channel gain is exploited to optimize the energy efficiency of the network. Time is modeled as a finite number of slots corresponding to the look-ahead time of the prediction. When dealing with multimedia streaming, the data buffer is usually modeled in the constraints of the problem by linking the state at a given time slot to the previous slot. The solver will then choose whether to use resources in the current slot or use what has been accumulated in the buffer, as in, e.g., [51]. Admission control is often used to enforce quality-of-service, e.g., [48], [99], with the drawback of introducing integer variables in the optimization function. In these cases, the optimal ILP/MILP formulation is followed by a fast heuristic that enables the implementation of real-time algorithms.

B. Model Predictive Control

Model Predictive Control (MPC) is a control theoretic approach that optimizes the sequence of actions in a dynamic system by using the process model of that system within a finite time horizon. Therefore, the process model, i.e., the process that turns the system from one state to the next, should be known. In each time slot \( t \), the system state, \( x(t) \), is defined as a vector of attributes that define the relevant properties of the system. At each state, the control action, \( u(t) \), turns the system to the next state \( x(t+1) \) and results in the output \( y(t+1) \). In case the system is linear, both the next state and the output can be determined as follows:

\[
x(t+1) = Ax(t) + Bu(t) + \psi(t), \quad (16)
\]

\[
y(t) = Cx(t) + e(t), \quad (17)
\]

where \( \psi(t) \) and \( e(t) \) are usually zero mean random variables used to model the effect of disturbances on the input and output, respectively, and \( A, B, \) and \( C \) are matrices determined by the system model.

At each time slot, the next \( N \) states and their respective outputs are predicted and a cost function \( J(\cdot) \) is minimized to determine the optimal control action \( u^*(t) \) at \( t = t_0 \):

\[
u^*(t_0) = \arg \min_{u(t_0)} J(\hat{x}(t_0), u(t_0)), \quad (18)
\]

where \( \hat{x}(t_0) \) is the set of all the predicted states from \( t = t_0 + 1 \) to \( t = t_0 + N \), including the observed state at \( t = t_0 \). The expression in (18) essentially states that the optimal action of the current time slot is computed based on the predicted states of a finite time horizon in the future. In other words, in each time slot the MPC sequentially performs a \( N \) step lookahead open loop optimization of which only the first step is implemented [100].

This approach has been adopted for on-line prediction and optimization of wireless networks [62], [101]. Since the process model (for the prediction of future states and outputs) is available in this kind of systems, autoregressive methods can be used along with Kalman filtering [62], or max-min MPC formulation [102]. In [101], Kalman filtering is compared to other methods such as mean and median value estimation, Markov chains, and exponential averaging filters.

Optimization based on MPC relies on a finite horizon. The length of the horizon determines the trade-off between complexity and accuracy. Longer horizons need further look ahead and more complex prediction but in turn result in a more foresighted control action [102]. Reducing the horizon reduces the complexity while resulting in a more myopic action. This trade-off is examined in [101] by proposing an algorithm that adaptively adjusts the horizon length. In general, the prediction horizon is kept to a fairly low number (1 step in [102] and 6 steps in [62]) to avoid high computation overhead.

It is worth noting that MPC methods can be extended to the nonlinear case. In this case, the prediction accuracy and control optimality increase at the cost of more complex algorithms to find the solution [100]. Another benefit of these approaches is their applicability to non-stationary problems.
C. Markov Decision Process

Markov Decision Process (MDP) is an efficient tool for optimizing sequential decision making in stochastic environments. Unlike MPCs, MDPs can only be applied to stationary systems where a priori information about the dynamics of the system as well as the state-action space is available.

A MDP consists of a four tuple \((X, U, P, r)\), where \(X\) and \(U\) represent the set of all achievable states in the system and the set of all actions that can be performed in each of the states, respectively. Time is assumed to be slotted and in any time slot \(t\), the system is in state \(x_t \in X\) from which it can take an action \(u_t\) from the set \(U_{x_t} \subseteq U\). Due to the assumption of stationarity, we can omit the time subscript for states and actions. Upon taking action \(u\) in state \(x\), the system moves to the next state \(x' \in X\) with transition probability \(P(x'|x,u)\) and receives a reward equal to \(r(x,u,x')\). The transition probabilities are predicted and modeled as a Markov Chain prior to solving the MDP and preserve the Markovian behavior of the system.

The goal is to find the optimal policy \(\pi^∗ : X → U\) (i.e., optimal sequence of actions that must be taken from any initial state) in order to maximize the long term discounted average reward \(\mathbb{E}(\sum_{t=0}^{\infty} \gamma^t r(x_t, u_t, x_{t+1}))\), where \(0 ≤ \gamma < 1\) is called discount factor and determines how myopic (if closer to zero) or foresighted (if closer to 1) the decision process should be.

In order to derive the optimal policy, each state is assigned a value function \(V^\pi(x)\) which is defined as the long term discounted sum of rewards obtained by following policy \(\pi\) from state \(x\) onwards. The goal of MDP algorithms is to find \(V^\pi^∗(x) (\forall x \in X)\). Given that the Markovian property holds, it has been proved that the optimal value functions follow the Bellman optimality criterion described below \[103\]:

\[
V^\pi^∗(x) = \max_{u \in U} \sum_{x' \in X'} \left( r(x,u,x') + \gamma P(x'|x,u)V^\pi^∗(x') \right) \tag{19}
\]

where \(X' \subseteq X\) is the set of states for which \(P(x'|x,u)>0\).

In order to solve the above equation set, linear programming or dynamic programming techniques can be used, in which the optimal policy is derived by simple iterative algorithms such as policy iteration and value iteration \[103\].

MDPs are very efficient for several problems, especially in the framework of anticipatory networking, due to their wide applicability and ease of implementation. MDP-based optimized download policies for adaptive video transmission under varying channel and network conditions are presented in \[43\], \[45\], \[104\].

In order to avoid large state spaces (which limit the applicability of MDPs), there are cases where the accuracy of the model must be compromised for simplicity. In \[104\], a large video receiver buffer is modeled for storing video on demand but only a small portion of the buffer is used in the optimization, while the rest of the buffer follows a heuristic download policy. \[43\], \[45\] solve this problem by increasing the duration of the time slot such that more video can be downloaded in each slot and, therefore, the buffer is filled entirely based on the optimal policy. This in turn comes at the cost of lower accuracy, since the assumption is that the system is static within the duration of a time slot. Heuristic approaches are also adopted for on-line applications. For instance, creating decision trees with low depth from the MDP outputs is proposed in \[45\]. Simpler heuristics were also applied to the MDP outputs in \[43\], \[90\], \[104\].

If any of the assumptions discussed above does not hold, or if the state space of the system is too large, MDPs and their respective dynamic programming solutions algorithms fail. However, there are alternative techniques to solve these kinds of problems. For instance, if the system dynamics follow a Markov Renewal Process instead of a MC, a semi MDP is solved instead of the regular one \[103\]. In non-stationary systems, for which the dynamics cannot be predicted a priori or the reward function is not known beforehand, reinforcement learning \[105\] can be applied and the optimization turns into an on-line unsupervised learning problem. Large state spaces can be dealt with using value function approximation, where the value function of the MDP is approximated as a linear function, a neural network, or a decision tree \[105\]. If different subsets of state attributes have independent effects on the overall reward, i.e., multi user resource allocation, the problem can be modeled as a weakly coupled MDP \[106\] and can be decomposed into smaller and more tractable MDPs.

D. Takeaways

This section (and Table \[V\]) summarizes the main takeaways of this optimization handbook.

1) Convex Optimization methods: These methods are often combined with time series analysis or ideal prediction. The main reason is that they are used to determine performance bounds when the solving time is not a system constraint. Thus, convex optimization is suggested as a benchmark for large scale prediction. This may have to be replaced by fast heuristics in case the optimization tool needs to work in real-time. An exception to this is LP for which very efficient algorithms exist that can compute a solution in polynomial time. In contrast, convex optimization methods should be preferred when dealing with high precision and continuous output. They require the complete dataset and show a reliability comparable to that of the used predictor.

2) Model Predictive Control: MPC combines prediction and optimization to minimize the control error by tuning both the prediction and the control parameters. Therefore, it can be coupled with any predictor. The main drawback of this approach is that, by definition, prediction and optimization cannot be decoupled and must be evaluated at each iteration. This makes the solution computationally very heavy and it is generally difficult to obtain real-time algorithms based on MPC. The close coupling between prediction and optimization makes it possible to adopt the method for any application for which a predictor can be designed with the only additional constraint being the execution time. Objectives and constraints are usually those imposed by the used predictor.

3) Markov Decision Processes: MDPs are characterized by a statistical description of the system state and they usually...
model the system evolution through probabilistic predictors. As such, they best fit to scenarios that show similar objective functions and constraints as those of probabilistic predictors. Thus, MDPs are the ideal choice when the optimization objective aims at obtaining stationary policies (i.e., policies that can be applied independently of the system time). This translates to low precision and high reliability. Moreover, even though they require a computationally heavy phase to optimize the policies, once the policies are obtained, fast algorithms can easily be applied.

VI. SUMMARY AND OPEN CHALLENGES

We conclude the paper by providing some insights on how anticipatory optimization will enable new 5G use cases and by detailing the open challenges of anticipatory networking to be successfully applied in 5G.

A. Anticipation-enabled use cases

Future networks are envisioned to cater to a large variety of new services and applications. Broadband access in dense areas, massive sensor networks, tactile Internet and ultra-reliable communications are only a few of the use cases detailed in [107]. The network capabilities of today’s systems (i.e., 4G systems) are not able to support such requirements. Therefore, 5G systems will be designed to guarantee an efficient and flexible use (and sharing) of wireless resources, supported by a native software defined network and/or network function virtualization architecture [107]. Big data analysis and context awareness are not only enablers for new value added services but, combined with the power of anticipatory optimization, they will be part of the 5G technology.

1) Mobility management: Network densification will be used in 5G systems in order to cope with the tremendous growth of traffic volume. As a drawback, mobility management will become more difficult. Additionally, it is foreseen that mobility in 5G will be on-demand [107], i.e., provided for and customized to the specific service that needs it. In this sense, being able to predict the user’s context (e.g., requested service) and his mobility behavior can be extremely useful in order to speed up handover procedures and to enable seamless connectivity. Furthermore, since individual mobility is highly social, social context and mobility information will be jointly used to perform predictions for a group of socially related individuals.

2) Network sharing: 5G systems will support resource and network sharing among different stakeholders, e.g., operators, infrastructure providers, service providers. The effectiveness of such sharing mechanisms relies on the ability of each player to predict the evolution of his own network, e.g., expected network load, anticipated user’s link quality and prediction of the requested services. Wireless sharing mechanisms can strongly benefit from the added value provided by anticipation, especially when prediction is available at fine granularity, e.g., in a multi-operator scheduler [108].

3) Extreme real-time communications: Tactile Internet is only one of the applications that will require a very low latency (i.e., in the order of some milliseconds). Allocating resources and guaranteeing such low end-to-end delay will be very challenging. 5G systems will support such requirements by means of a new physical layer (e.g., a new air interface). However, this will not be enough if not combined with context information used to prioritize control information (e.g., used to move virtual or real objects in real time) over content [109]. Knowledge about the information that is transmitted and its specific requirements will be crucial in order to assign priorities and meet the expected quality-of-experience in a combined effort of physical and higher layers.

4) Ultra-reliable communications: Reliability is mentioned in several 5G white papers, e.g. in [107], as necessary prerequisite for lifeline communications and e-health services, e.g., remote surgery. A recent work [110] proposed a quantified definition of reliability in wireless access networks. As outlined here, a posteriori evaluation of the achieved reliability is not enough in order to meet the expected target, which in some cases is as high as 99.999%. To this end, it is mandatory to design resource allocation mechanisms that account for (and are able to anticipate the impact on) reliability in advance.

B. Open challenges

While the literature surveyed so far clearly points out how anticipatory networking enhances current networks, this section discusses a few obstacles to its adoption.

1) Privacy and security: In our opinion, one of the main hindrances for anticipatory networking to become part of next generation networks is related to how users feel about sharing data and being profiled. While voluntarily sharing personal information has become a daily habit, many disapprove that companies create profiles using their data [111]. In a similar way, there might be a strong resistance against a new technology that, even though in an anonymous way, collects
and analyzes users’ behavior to anticipate users’ decisions. Standards and procedures need to be studied to enforce users’ privacy, data anonymity and an adequate security level for information storage.

2) Network functions and interfaces: Many of the applications that are likely to benefit from anticipatory networking capabilities require unprecedented interactions among information producers, analyzers and consumers. A simple example is provided by predictive media streaming optimizers, which need to obtain content information from the related database and user streaming information from the user and/or the network operator. This information is then analyzed and fed to a streaming provider that optimizes its service accordingly. While ad hoc services can be realized exploiting the current networking functionalities, next generation applications, such as the extreme real-time communications mentioned above, will greatly benefit from a tighter coupling between context information and communication interfaces.

3) Next generation architecture: 5G networks are currently being discussed and, while much attention is paid to increasing the network capacity and virtualizing the network architecture, we believe that the current infrastructure should be enhanced with repositories for context information and application profiles [12] to assist the realization of novel predictive applications. As per the previous concerns above, sharing sensible information, even in an anonymized way, will require particular care in terms of users’ privacy and database accessibility.

To conclude, while the literature reviewed in this works suggests that anticipatory networking is a quite mature approach to improve the performance of mobile networks, we believe that issues (mainly at the system level) still need to be solved to fully unleash its potentials. In particular, most of the work which has been evaluated in this survey tends to focus on the benefit of anticipation, while overlooking possible threats in the anticipatory networking framework.

All the main components of anticipatory networking, the context database and the prediction/anticipation intelligence, must be effectively integrated into the mobile network architecture which poses challenges at different levels; first, new interfaces and communication paradigms must be defined for data collection from both end users and sources external to the mobile network itself; second, the management of the context databases brings an additional burden in terms of required bandwidth and processing power for several network elements which may lead to scalability issues.

To this extent, a thorough and comprehensive cost-benefit analysis for specific anticipatory networking scenarios is, in our opinion, a required next step for the research in the field.

REFERENCES

[1] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, “Limits of predictability in human mobility,” Science, vol. 327, no. 5968, pp. 1018–1021, 2010.
[2] X. Lu, E. Wetter, N. Bharti, A. J. Tatem, and L. Bengtsson, “Approaching the limit of predictability in human mobility,” Scientific reports, vol. 3, 2013.
[3] Y. Jiang, D. C. Dhanapala, and A. P. Jayasumana, “Tracking and prediction of mobility without physical distance measurements in sensor networks,” in Communications (ICC), 2013 IEEE International Conference on. IEEE, 2013, pp. 1845–1850.
[4] L. Ghouti, T. R. Sheltami, and K. S. Alutaibi, “Mobility prediction in mobile ad hoc networks using extreme learning machines,” Procedia Computer Science, vol. 11, pp. 305–312, 2011.
[5] X. Chen, F. Mériaux, and S. Valentin, “Predicting a user’s next cell with supervised learning based on channel states,” in Signal Processing Advances in Wireless Communications (SPAWC), 2013 IEEE 14th Workshop on, June 2013, pp. 36–40.
[6] H. Xiong, D. Zhang, D. Zhang, V. Gauthier, K. Yang, and M. Becker, “MPaaS: Mobility prediction as a service in the telecom cloud,” Information Systems Frontiers, vol. 16, no. 1, pp. 59–75, 2014.
[7] J.-K. Lee and J. C. Hou, “Modeling steady-state and transient behaviors of user mobility: formulation, analysis, and application,” in Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing. ACM, 2006, pp. 85–96.
[8] H. Abu-Ghazaleh and A. S. Alfa, “Application of mobility prediction in wireless networks using Markov renewal theory,” Vehicular Technology, IEEE Transactions on, vol. 59, no. 2, pp. 788–802, 2010.
[9] D. Barb, S. Bellahsene, and K. Kloul, “Mobility prediction using mobile user profiles,” in Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on. IEEE, 2011, pp. 286–294.
[10] ——, “Combining local and global profiles for mobility prediction in LTE femtocells,” in Proceedings of the 15th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems. ACM, 2012, pp. 333–342.
[11] G. Gidófálvai and F. Dong, “When and where next: Individual mobility prediction,” in Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems. ACM, 2012, pp. 57–64.
[12] Y. Chon, N. D. Lane, Y. Kim, F. Zhao, and H. Cha, “Understanding the coverage and scalability of place-centric crowdsensing,” in Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. ACM, 2013, pp. 3–12.
[13] Y. Chon, H. Shin, E. Talipov, and H. Cha, “Evaluating mobility models for temporal prediction with high-granularity mobility data,” in Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on. IEEE, 2012, pp. 206–212.
[14] Y. Chon, E. Talipov, H. Shin, and H. Cha, “SmartDC: Mobility prediction-based adaptive duty cycling for everyday location monitoring,” Mobile Computing, IEEE Transactions on, vol. 13, no. 3, pp. 512–523, 2014.
[15] ——, “Mobility prediction-based smartphone energy optimization for everyday location monitoring,” in Proceedings of the 9th ACM conference on embedded networked sensor systems. ACM, 2011, pp. 82–95.
[16] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell, “Nextplace: a spatio-temporal prediction framework for pervasive systems,” in Pervasive Computing. Springer, 2011, pp. 152–169.
[17] M. De Domenico, A. Lima, and M. Musolesi, “Interdependence and predictability of human mobility and social interactions,” Pervasive and Mobile Computing, vol. 9, no. 6, pp. 798–807, 2013.
[18] J. Yang and Z. Fei, “Broadcasting with prediction and selective forwarding in vehicular networks,” International journal of distributed sensor networks, vol. 2013, 2013.
[19] A. Sridharan and J. Bolot, “Location patterns of mobile users: A large-scale study,” in INFOCOM, 2013 Proceedings IEEE, April 2013, pp. 1007–1015.
[20] J. Froehlich and J. Krumm, “Route prediction from trip observations,” SAE Technical Paper, Tech. Rep., 2008.
[21] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti, “WhereNext: a location predictor on trajectory pattern mining,” in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 637–646.
[22] “GeoPKDD: Geographic Privacy-aware Knowledge Discovery and Delivery,” 2005-2008. [Online]. Available: http://www.geopkdd.eu
[23] N. Bui, F. Micheliniakis, and J. Widmer, “A Model for Throughput Prediction for Mobile Users,” in European Wireless 2014; 20th European Wireless Conference; Proceedings of, 2014, pp. 1–6.
[24] Y. Li, S. Valentin, and S. Stanczak, “Channel gain prediction in wireless networks based on spatial-temporal correlation,” in Signal Processing Advances in Wireless Communications (SPAWC), 2015 IEEE 16th International Workshop on. IEEE, 2015, pp. 400–404.
