A Survey on Client Throughput Prediction Algorithms in Wired and Wireless Networks

JOSEF SCHMID and ALFRED HÖSS, OTH Amberg-Weiden, Germany
BJÖRN W. SCHULLER, University of Augsburg, Germany

Network communication has become a part of everyday life, and the interconnection among devices and people will increase even more in the future. Nevertheless, prediction of Quality of Service parameters, particularly throughput, is quite a challenging task. In this survey, we provide an extensive insight into the literature on Transmission Control Protocol throughput prediction. The goal is to provide an overview of the used techniques and to elaborate on open aspects and white spots in this area. We assessed more than 35 approaches spanning from equation-based over various time smoothing to modern learning and location smoothing methods. In addition, different error functions for the evaluation of the approaches as well as publicly available recording tools and datasets are discussed. To conclude, we point out open challenges especially looking in the area of moving mobile network clients. The use of throughput prediction not only enables a more efficient use of the available bandwidth, the techniques shown in this work also result in more robust and stable communication.

CCS Concepts: • **Networks** → *Network control algorithms*; **Traffic engineering algorithms**; **Packet scheduling**; Transport protocols;

Additional Key Words and Phrases: TCP, throughput prediction, connectivity map, time series

ACM Reference format:
Josef Schmid, Alfred Höss, and Björn W. Schuller. 2021. A Survey on Client Throughput Prediction Algorithms in Wired and Wireless Networks. *ACM Comput. Surv.* 54, 9, Article 194 (October 2021), 33 pages.
https://doi.org/10.1145/3477204

1 INTRODUCTION

The dramatic increase in data transmitted over the Internet represents a challenge for the infrastructure. To meet this challenge, transmission rates are being increased and new technologies are being developed. More efficient use of bandwidth can also help to increase the amount of data transferred. This is demonstrated, for example, by developments in the field of mobile video streaming, in which **Throughput Prediction (TPP)** becomes progressively more important. The topic itself, however, is not a new one. Knowing the time required to transfer a certain amount of data was even relevant at times without mobile devices at all. With the change to services provided via the Internet, the use cases for TPP changed. In the early years of TPP, the transfer time

Authors' addresses: J. Schmid and A. Höss, OTH Amberg-Weiden, Kaiser-Willhelm-Ring 23, Amberg, Germany; emails: {j.schmid, a.hoess}@oth-aw.de; B. W. Schuller, University of Augsburg, Eichleitnerstr. 30, Augsburg, Germany; email: schuller@informatik.uni-augsburg.de.

0360-0300
https://doi.org/10.1145/3477204

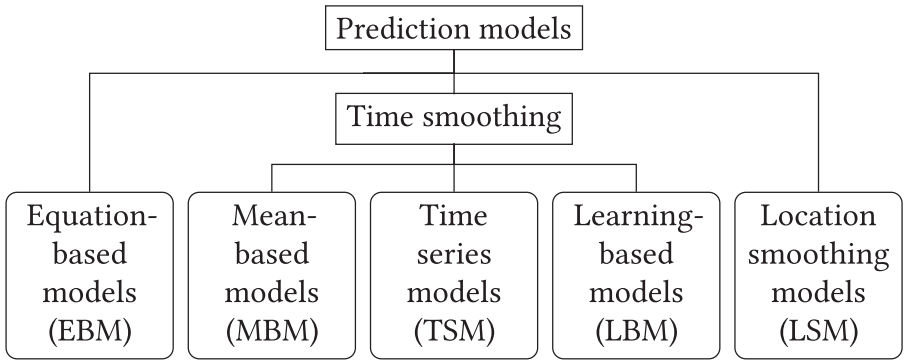


Fig. 1. A holistic taxonomy of commonly used prediction models. Starting from the first equation-based approaches to the state-of-the-art geographic and machine learning-based ones.

of bulky **Transmission Control Protocol (TCP)** connections was one of the most interesting topics [38]. Nowadays, research shifted more and more to applications in the area of mobile adaptive streaming (see, e.g., References [5, 7, 9, 10, 29, 61, 80, 124, 134]), other mobile applications like vehicle-to-network communication for autonomous driving [48, 99]. Especially for streaming real-time map information [46] and scheduling multi-provider connections in public transport systems [126], there is already evidence that TPP can improve the process [137]. There are also approaches, which use the prediction to decide if the download of software updates should be started or not [28], as well as for more general tasks in data transmission in vehicle to infrastructure [78] or cellular vehicle-to-vehicle environment, which is one form of intelligent transportation system communication [59]. Therefore, this article focuses on presenting an overview of different prediction models. It concentrates on the prediction of the two quality parameters **Throughput (TP)** and **Round-Trip Time (RTT)**, as well as on providing a taxonomy of the different methods for doing so. To cope with this challenge, this work is targeting the following main contributions:

- First, Section 2 describes the recent works and discusses surveys related to the topic of this work. This includes also a holistic taxonomy of commonly used prediction model shown in Figure 1.
- Deployment areas and advantages of TPP are subject of Section 3 and range from classic server to client communication to modern mobile networks.
- A detailed introduction of more than 35 different approaches is given in Section 4, which to our knowledge, is the most holistic work ever done on this topic.
- A summarizing overview of all prediction models is presented in online only material. It comprises their classification, input parameters and error function.
- A list of available datasets and of tools, which can be used to create additional datasets, is shown in Section 5.
- The work is supported by the discussion of open issues, and future work that should be done on TPP in Section 6.
- Finally, Section 7 presents the conclusion of the survey.

First, however, the scope and limitations of this work have to be clarified.

1.1 Scope and Limitations of This Work

Since the main purpose of this work is to study the TP of a single connection on the client side, the scenarios and methods described in this article are limited to this scope. This means that traffic,

e.g., handled in data centers or other types of large-scale broadband networks, is not further investigated in this article, although there are, of course, interesting approaches for these applications, as demonstrated in the works of Kelly et al. and Cardwell et al. [18, 52].

In addition, approaches containing multiple paths for one connection, such as Multipath TCP [115] or the control of mutable connection [114] are not discussed in this work.

2 RELATED WORK

In this section, the most relevant surveys regarding TPP are discussed. To cluster the related work, this section is structured in methods using just the throughput itself, so called univariable approaches and algorithms using additional parameters like low-level value of the connection or the geographic location, called multivariable approaches.

2.1 Univariable Approaches

The use of algorithms to predict the TP of TCP connections is shown by Qiao et al. [83], where the focus is on mean-based approaches as well as on linear and nonlinear time series methods. A wide range of prediction models is evaluated in three traces recorded on different wired scenarios. The **Time Series Models (TSMs)** can also be used in mobile network data as presented by Bui et al. [17], where the authors not only investigated the classic linear models, e.g., **Moving Average (MA)**, **Autoregressive Model (AR)**, **Autoregressive Moving Average (ARMA)**, **Autoregressive Integrated Moving Average (ARIMA)** (presented in Section 4.3) but also nonlinear ones.

2.2 Multivariable Approaches

Another type of approaches is shown by Raca et al. [85]. Here, **Learning-Based (LB)** algorithms are explored on simulated data provided for static and mobile cellular network connections. The simulation was done using the Network Simulator 3, where Raca et al. applied single algorithms, namely, **Support Vector Machine (SVM)** and Gaussian Process as well as ensemble methods like **Random Forest (RF)** and Gradient Boosting. Each approach was optimized via grid-search techniques. An evaluation of SVM used for regression and **Neural Networks (NN)** applied on real-world static cellular network data is presented by Liu and Lee [56]. The authors performed an empirical study to compare different **Mean-Based (MB)** methods against time series and machine learning models by using the data recorded at three different locations in Hong Kong. In total, they analysed seven prediction algorithms regarding their performance and characteristics. The fact that LB models are highly interesting in areas related to TPP is also shown by Nguyen and Armitage [72], who presented a survey regarding techniques used for **Internet Protocol (IP)** address traffic classification. The work of Bui et al. [16] takes also location smoothing prediction methods into account. The authors focus on wireless networks and present a large number of models utilizing learning-based algorithms and time smoothing as well. They concentrate not only on TPP but also show approaches regarding the quality of experience for specific applications and a discussion on challenges and issues in the context of the new mobile network standard **5th Generation of Cellular Mobile Communication (5G)**. But the survey of Bui et al. lacks in explaining methods for TPP that are not based on machine learning. For example, the authors are mentioning the nearest neighbour measurements as a **Location Smoothing Model (LSM)**, but other methods, such as the segment building ones presented in Section 4.5, are not mentioned and further categories are missing at all.

There is also a study of Zhang et al. [130] concentrating on deep learning methods in mobile networks. This study is investigating different areas in mobile and wireless network, apart from TPP. But, however, it only takes deep learning into account and does not consider approaches like **Location Smoothing (LS)** as mentioned in this work.

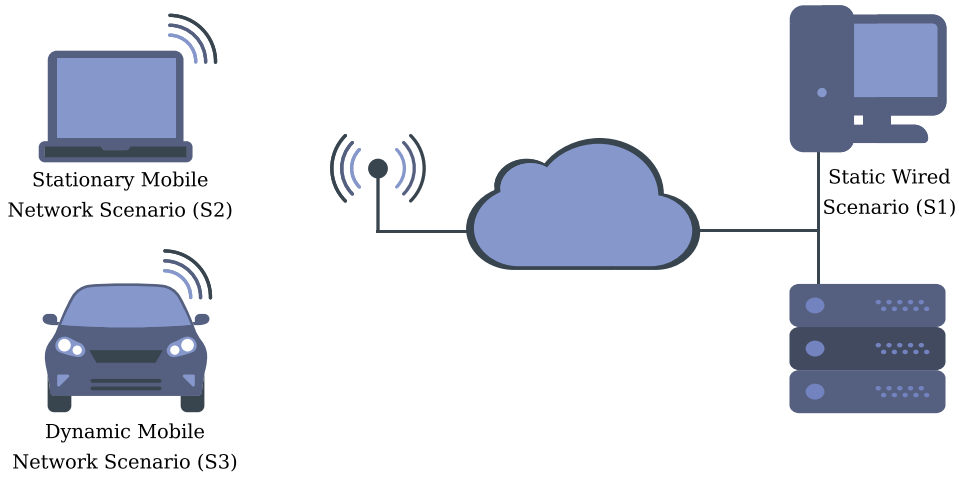


Fig. 2. The three different scenarios in which throughput prediction is used. S1 describes the classical wired client-server connection. The other two (S2 and S3) are establishing a connection via mobile network for static and moving clients.

2.3 Taxonomy of Models

Although the surveys listed in the previous section are presenting a selected number of prediction models for either wired or wireless scenarios, to the best of our knowledge, this is the first survey describing such a large number of different TCP TPP models including **Equation-Based (EB)**, time smoothing, and LS once. To structure the large number of models, a taxonomy is introduced in Figure 1, clustering the approaches according to the methods used. However, since there are very different techniques based on time smoothing, these are subdivided again. This results in five groups, which are explained in more detail in Sections 4.1 to 4.5. Of course, there are also methods that sometimes cannot be clearly assigned to a category due to the combination of techniques, but this has been noted in the following text as well as in a table presented in the online only material.

The employed transfer technologies have huge impact on the predictability and of course the accuracy of the TP prediction. Section 3 describes three basic scenarios, in which the individual models shown later on in Section 4 are categorized.

3 SCENARIO

To structure the approaches in this survey, the terms *use case* and *scenario* are used. A use case describes a concrete application like video on demand or bulky file download, while scenario means the compilation of the techniques and the environment, in which a method is evaluated. The different TPP approaches are categorized into three scenarios: First, a static wired client to Internet scenario (S1), second a scenario with a static client, which is connected via a mobile network and uses the whole time the same cell of the cellular network (S2) and third, a scenario with a moving mobile network client that changes its location (S3). We need to distinguish between S2 and S3, since other works, e.g., Mirza et al. [67] or Yue et al. [129], have shown huge differences in accuracy between stationary and moving mobile network clients. Of course, not all approaches are focused on only one of the scenarios, shown in Table 1. Therefore, this article shows a summary of models for all three of them.

3.1 Static Wired Scenario (S1)

The most frequent scenario in which TPP was investigated, is the static wired scenario based on a classical cable link between client and server: This can be done either via a directed cable connection or by using the Internet, e.g., with a broadband interface. Since this is the oldest scenario, many models exist for it, and there are even testbeds, which can be used to collect data. In simulation environments, the Internet is described as a group of meshed local links with different bandwidths and latencies. Most of the prediction algorithms for this scenario are only based on TCP related parameters, since the lower level network parameters seem not to have a big impact and cannot easily be investigated because of their diversity. As already mentioned in Section 1.1, this scenario does not include data center environments, since these are out of the scope of this article.

3.2 Stationary Mobile Network Scenario (S2)

The stationary mobile network scenario describes a static client with a mobile data connection (e.g., **Long-Term Evolution (LTE)** or 5G) that is connected to a server. Although the client would be able to move in this scenario, S2 focuses on measurements done at a specific location. So, there are no influences from moving and cell hand-over, and so on. This scenario looks quite similar to S3, where the client is dynamic. However, e.g., Mirza et al. [67], who performed measurements for Wi-Fi networks, found significant differences regarding prediction accuracy for static wireless networks compared to moving ones. Mirza et al. figured out a factor of 1.5 in accuracy. Also other papers are distinguishing between static (S2) and (S3) [110, 129].

3.3 Dynamic Mobile Network Scenario (S3)

The S3 scenario summarizes all cases in which the client moves. This can mean walking, driving or even travelling in a high-speed train [117]. Like in the other two scenarios, this scenario could be separated into several sub-scenarios, which would add a lot of complexity, especially to the comparison shown in online only material. The main challenge of the S3 scenario is the handling of cell hand-overs and the higher fluctuation of lower level parameters during movement. The scenario does not include device-to-device communication in a cellular network, as analysed by Asadi et al. [3], but to the best of our knowledge, no TCP TP prediction is performed in this area so far, so no other scenario is needed.

4 PREDICTION MODELS

There are already proposals to categorize TPP models, described by He et al. [38]. They divided the models in formula-based and history-based ones. Since this survey covers a large range of models, we introduced our own taxonomy by defining five groups of different approaches shown in Figure 1. In Section 4.1 approaches are presented, which try to calculate the throughput by modelling the protocols used in TCP. These models can be classified as EB and related to the formula-based ones shown by Zhou et al. [136]. Another type of models are the **Mean-Based Models (MBMs)** described in Section 4.2. Those utilize smoothing of the last samples to predict the next value based on an average of the previous ones. This technique suggests that the TCP TP can be interpreted as a time series of measurement points. Therefore, Section 4.3 shows the approaches using TSMs like AR or ARMA and Section 4.4 contains models utilizing LB techniques, mainly regression. A different strategy is shown in the fifth group, where models based on the location are described. To further structure these groups, the models using MB and LB techniques as well as the TSM are grouped together under the category time smoothing. In addition to Figure 1, Table 1 provides an overview of all approaches, categorizing them regarding the used prediction

Table 1. Overview of the Prediction Approaches, Classified by Their Scenarios and Models

	Static Wired Scenario (S1)	Stationary Mobile Network Scenario (S2)	Dynamic Mobile Network Scenario (S3)
EBM	He et al. [37], Hwang and Yoo [43], Padhye et al. [76], Cardwell et al. [19], Goyal et al. [33], Huang and Subhlok [42], Liu and Rao [55]		
MBM	Borzemski and Starczewski [12], Miller et al. [65], Liu and Lee [56]		
TSM	Zhani et al. [133, 135], Yoshida et al. [127], Karrer [50], Sadek and Khotanzad [89],	Torres et al. [106]	Wei et al. [110]
LBM	Hu et al. [41], Zhani et al. [133], El Khayat et al. [27], Mirza et al. [66], Borzemski and Starczewski [12], Lee et al. [54], Rao et al. [87]	Samba et al. [90] [91], Ghasemi [32], Wei et at. [112]	Wei et at. [110] [112], Samba et al. [90] [91]
LSM			Yao et al. [122], Yue et al. [129], Pögel and Wolf [82], Murtaza et al. [69], Curcio et al. [22], Hao et al. [36], Riiser et al. [88], Kamakaris and Nickerson [49], Estevez and Carlsson [28], Opitz et al. [74], Taani and Zimmermann [105], Sliwa et al. [99]

method as well as the scenario. A more detailed summary is shown in the online only material, which includes input parameters and error functions, used to calculate the difference between predicted and measured values. The last paragraph of this section provides a comparison of the five groups identifying their strengths and weaknesses.

4.1 Equation-Based Models

In this section, we present methods, which use mathematical equations to describe the characteristics of a TCP transmission. On the one hand, these approaches do not require a lot of computing power, but on the other hand, they use a very fine granularity, which means that this approaches are performed more often and some of the models presented in this section are belonging to this category. Consequently, the main scenario for this class of predictors is S1, as explained in Section 3, where the connection is more stable and lower level parameters such as the Signal-to-Noise Ratio are quite constant.

Considering the range of **Equation-Based Models (EBMs)**, the TCP **Congestion Control (CC)** with its two schemes slow start and congestion window is very important. To achieve a high performance, the CC algorithm is defining a **congestion window size (*CWND*)**, which is a multiple of the maximum segment size. During the slow start, this *CWND* is doubled after every correct transmission, until the slow start threshold is reached. At this point, the *CWND* is calculated using the congestion window scheme. For TCP algorithms like TCP Tahoe, this means that the *CWND* is increased by one. But the *CWND* can also be decreased when a loss event happens. There are two types of loss events:

The first one is the loss of single segment and can be detected with the duplicate acknowledgment method. The period between two of these events is defined as **Triple-Duplicate ACK Time**

Table 2. Values for Congestion Avoidance Constant (C) According to Loss Type and Acknowledgment Strategy

Derivation	Acknowledgment strategy	C
Periodic Loss [64]	Every Packet	$1.22 = \sqrt{3/2}$
Periodic Loss [64]	Delayed	$0.87 = \sqrt{3/4}$
Random Loss [75]	Every Packet	1.31
Random Loss [75]	Delayed	0.93

(T_D) in the following. The second one is a timeout, to model this loss, the **TCP Retransmission Timeout Period (T_O)** is used.

One model for calculating the **Predicted Bandwidth ($\hat{B}$)** that is based on the CC, is the Mathis-Equation (1) presented by Mathis et al. [64]. It also considers the RTT. This is the amount of time needed for a segment to be sent and answered, and therefore, has a huge effect on the bandwidth. The $CWND$ and the likelihood that a single random message gets lost are also taken into account. The latter is defined as the **Package Loss Probability (l)** and describes the probability of a segment loss after the correct transmission of a certain number of segments. Due to the definition of the $CWND$ done by the CC, there is an additional constant Congestion Avoidance Constant (C) ($C = \sqrt{3/2}$) in the Equation (1). In this article, the authors Mathis et al. also define values of C according to the type of single loss and the acknowledgment strategy used by the TCP implementation. This assumption is presented in Table 2. This model is also called the SQRT model [27, 133]. Mathis et al. introduced the C as a combination of several terms that are typically constant for a given combination of TCP implementations. This term can be carried through any of the derivations and always reduces C by $\sqrt{2}$.

$$\hat{B} = \frac{data\ per\ cycle}{time\ per\ cycle} = \frac{CWND}{RTT} \times \frac{C}{\sqrt{l}}. \tag{1}$$

Looking at the Mathis equation, it is very important to note that it is strongly connected to the TCP CC, which leads to the situation that it is not well suited for TCP flows that are not in line with this algorithm. Therefore, the model is inaccurate on:

- A small data transfer, which is even too small that the performance is controlled by the CC, because it only happens within the slow start phase.
- A link, which is not continuously sending.
- A connection using another $CWND$ strategy.

An evaluation on simulated and real-world data is shown by Padhye et al. [76], which demonstrates that the model can be used if the losses are infrequent and isolated. This also means that the model is inaccurate for connections with non-randomized losses, such as drop-tail queues, for which all packages were lost after a single error.

Therefore, another Equation-Based Model (EBM) was created by Padhye et al. to prevent the overestimation of the Mathis-Equation in some cases. It models the behaviour of the TCP CC in rounds. A round begins with the transmission of $CWND$ TCP segments. Once all segments are sent, nothing more is sent until the first **Acknowledgement (ACK)** corresponding to one of these segments is revised. This Acknowledgement (ACK) reception signals the end of the current round and the start of the next one. The model defines the RTT as the duration of a round and independent of the $CWND$. In addition, the number of packages acknowledged in an answer is defined as b and typically set to 2. So, for a round of $CWND$ segments, $CWND/b$ acknowledges must be received. To

handle a single package loss for the *triple duplicate acknowledgments*, the property l is introduced. The losses detected by a T_O are also taken into account. Finally, the model depends on the maximal $CWND$, $CWND_{max}$, which is the upper limit for the throughput. The whole equation is given by

$$\hat{B}(l) \approx \min\left(\frac{CWND_{max}}{RTT}, \frac{1}{RTT\sqrt{\frac{2bl}{3}} + T_O \times min(1, 3\sqrt{\frac{3bl}{8}})l(1+32l^2)}\right). \tag{2}$$

The model is validated in a wired server-to-server scenario, where Padhye et al. [76] measured the number of packets, the losses indicated by duplicate acknowledgments or time outs, the RTT and the time out time. They built two datasets: One with 28 traces each 1 h long and another one with 13 traces and a duration of 100 s each. The results are showing a Normalized Error (NE) between 0.1 and 2.2 for the 1 h datasets and an error from 0.08 to 0.6 for the 100 s,

$$NE = \frac{\sum_{i=1}^{n} \frac{\hat{B}_i - B_i}{B_i}}{n}. \tag{3}$$

He et al. [37] have highlighted the limitation of the approach regarding lossless paths (l is zero). He et al. deal with this by predicting the TCP TP based on the Mean Bandwidth ($\bar{B}$), which can be measured non-intrusively with end-to-end probing techniques as shown by Jain and Dovrolis [45]. They also define an upper boundary $CWND_{max}/RTT$ for lossless paths. If applied together with Equation (2), then it leads to the following:

$$\hat{B}(l) = \begin{cases} \hat{B}(l) & l > 0 \\ \min\left(\frac{CWND_{max}}{RTT}, \bar{B}\right) & l = 0 \end{cases}. \tag{4}$$

Hwang and Yoo [43] have also shown an improvement of Equation (2) by defining the model for different value ranges of $CWND$. This changes the model in a way that it is no longer a function of loss rate, which is difficult to observe accurately, but depends on the available bandwidth. So, this model improves the field of EB prediction in accuracy as well as in usability. Therefore, TCP Retransmission Timeout Period (T_O), l and the window size W_m of the receiver side need to be taken into account (see Equation (2)). In addition, the Router Buffer Size as Packet Unit (R_p) is also considered as well as the Average Bandwidth per Seconds (S). The model defines $CWND$ as a function depending on these two variables: $CWND(S, R_p) = S \times RTT + R_p$. The TPP function is also based on S and R_p and defined for different ranges of these parameters. It is given as

$$T(S, R_p) = \begin{cases} \frac{\frac{1}{4}CWND(\frac{3}{2}CWND+5)+Q(CWND)}{RTT(\frac{CWND}{2}+2)+\frac{R_p}{S}(\frac{R_p+3}{2})+QZ} & 0 < R_p \leq \frac{CWND}{2}, CWND < CWND_{max} \\ \frac{\frac{1}{4}CWND(\frac{3}{2}CWND+5)+Q(CWND)}{RTT(\frac{CWND}{2}+2)-\frac{(\frac{W+2}{2})(R_p-\frac{W}{4})+R_p}{S}+QZ} & \frac{CWND}{2} < R_p, CWND < CWND_{max} \\ \frac{W_m}{RTT+\frac{max(0, CWND_{max}-S\times RTT)}{S}} & CWND \leq CWND_{max} \end{cases}. \tag{5}$$

The variables $Q(CWND)$ and QZ are used for the probability of a triple-duplicate ACK in relation to the window size and the likelihood of a timeout. Hwang and Yoo [37] evaluated this model by predicting the real TCP TP of a 100 Mbps **Local Area Network (LAN)** environment with a simple Dumbbell topology. To simulate Wide Area Network situations, a number of background TCP transfers was used. A plot of the simulation shows that their model is much more accurate than the one shown by Padhye et al. [76]. But apart from Equation (5), there are also other improvements on the Padhye model. Cardwell et al. [19] introduced a technique for better slow start prediction,

resulting in a more accurate model overall. Goyal et al. [33] modified the model to be more accurate in terms of bulk transfer TPP.

A totally different approach based on the TCP algorithms is called fast pattern prediction shown by Huang and Subhlok [42]. They define four distinct TCP data transfer patterns:

- **Rate control:** Where the TCP TP is constant and limited by a bottleneck.
- **Congestion Control:** Where the TCP TP is fully limited by the TCP Congestion Avoidance algorithm, which means the window size is rising until a loss of a TCP message happens, which first cuts the throughput and then raises it again.
- **Rate control with delay:** Similar to rate control, but with short break done due to a delay.
- **Mixed Congestion Control:** A mixture of the other patterns, where it is not clear, which pattern is measured.

The predictor tries to detect the actual pattern by features like the climb rate of the TP. As a simplification, the authors used a moving average with a small window for data smoothing. The model was evaluated in a typical wired network download scenario (S1), where files with an average size of 30 MB are downloaded using the Linux program *wget*. The evaluation was done using the absolute error, defined as $\frac{\hat{B}-B}{B} \times 100\%$. The results show an error between 15 % to 25 % depending on the number of previous measurements used and the prediction horizon.

All approaches described above are based on a simple TCP implementation using only slow start, Triple-Duplicate ACK Time (T_D) and T_O, but there are also further improvements of the TCP implementations and mechanisms. One of them is, e.g., the additive increase multiplicative decrease algorithm [20] with improves the rate control. To predict the TP of connections using such implementations, models such as the ones described by Yang and Lam [119] are needed. The approach of congestion control using feedback of the other end of the TCP connection is presented by Chiu and Jain [20], but since this requires the control of both sides of the connection, it is not further investigated in this work. There are also TCP implementations, which are using a basic TP estimation to control the *CWND*, as done by TCP Vegas [57]. This implementation uses the *RTT* to calculate an expected data rate. The difference between this expected rate and the actual one is then used to increase or decrease the *CWND*. TCP Westwood [63], which is also a modification of the TCP congestion window algorithm aims to improve the performance of wireless links. A detailed analysis of different algorithms and implementations used for congestion control can be found in the work of Srikant [103]. To implement complex congestion control functions, the paper of Narayan et al. [70] is worth as further reading.

All approaches described above are based on a single TCP stream in a wired environment. To use them for multiple TCP connections, some modifications proposed by Lu et al. [58] have to be made. Equation (1) is modified in a way to process multiple TCP flows. Another EBM used for multiple stream is to calculate the number of connections that can be used without congesting the network as proposed by Yildirim et al. [125].

However, apart from the TCP implementations, there are also concave-convex methods used for TCP modeling under S1, like those presented by Rao et al. [86] and Liu and Rao [55]. These models are able to propose a coefficient that characterizes the overall TP profile and offer possibilities for **Quality of Service (QoS)** optimization by the adjustment of parameters like the buffer sizes and parallelism.

In addition to EB prediction, there are also models based on previously recorded data. He et al. [37] and Arlitt et al. [2] compared them and came to the conclusion that in general these data-based models can be superior to EB ones. He et al. compared the EB predictor shown in Equation (4) for lossy paths against MB ones, such as Moving Average (MA) and **Exponentially Weighted Moving Average (EWMA)**, which are going to be introduced in Section 4.2. They came to the

conclusion that even simple data smoothing prediction on average is much more accurate. One major cause for the inaccuracy of EB prediction is that RTT and l before the transfer can be significantly different to the one measured while the transfer is taking place [37]. Therefore, the rest of this survey focuses on models using historical recorded data.

4.2 Mean-Based Models

One type of basic approaches for prediction is data smoothing. As a technique to improve the signal quality by smoothing the noise; hence, it is very useful to detect the trend of a signal. In this section, a closer look on smoothing methods, utilizing a set of the last values is provided. Although the principle of using the mean value of past measurements to predict the future is commonly used in many applications in the area of TCP TPP, mean-based prediction is mainly applied for comparison with other models and not introduced as an approach on its own. Liu and Lee [56] investigated four different types of MB algorithms. The equations use K time intervals, with the measured throughput T_{i-k} for $k = 1, 2, \ldots, K$. $\hat{T}_i$ denotes the forecast value for the next time interval. Consequently, these models are able to predict one-step-ahead. To increase the prediction time, the duration of T_i needs to be increased. The K parameter controls how far in the past the values for prediction are taken into account for smoothing. Accordingly, K depends on the correlation of the historical data with the future value. Since the data points older K do not have any impact on the models, we call this type of methods instant time smoothing models. The four mean computations used by Liu and Lee [56] are shown in the following:

- Arithmetic Mean (AM).
- Harmonic Mean.
- Geometric Mean.
- Exponentially Weighted Moving Average (EWMA) for $0 < \alpha < 1$.

Regarding the optimizations presented by Mirza et al. [66, 67], the parameter α should be selected to be $\alpha = 0.3$. This parameter is used in EWMA to specify the influence of past values and is adopted by Liu and Lee.

The evaluation of the models is done on a mobile network setup. A notebook with 3G/**High Speed Packet Access (HSPA)** link captures a trace for a performance comprehension. It communicates with a server connected via a 100 Mbps link. This corresponds with a scenario S2 as described in Section 3. The authors tested three different locations, with varying values for K, between 1 and 60. They showed that despite these methods are very similar, there is a significant difference in terms of accuracy between them. Referring to Liu and Lee [56], for static mobile network connection, Arithmetic Mean (AM) with $K = 60$ performed best. As a metric a Normalized Root Mean Square Error ($NRMSE$), according to Equation (6), was used:

$$NRMSE = \sqrt{\frac{1}{n}\left(\sum_{i=1}^{n}(\hat{T}_i - T_i)^2\right)}. \tag{6}$$

For all three locations, it was between 0.1 and 0.15. Borzemski and Starczewski [12] also investigated MB functions, namely the EWMA and the Arithmetic Weighted Moving Average defined by Equation (7), using the weight w_k for the TP value T_{i-k}. They compared both models with the **Transfer Regression (TR)** model shown in Section 4.4. For data recording, they traced the download of a Linux distribution multiple times and analyzed the measurements:

$$\hat{T}_i = \frac{\sum_{k=1}^{K} w_k T_{i-k}}{\sum_{k=1}^{K} w_k}. \tag{7}$$

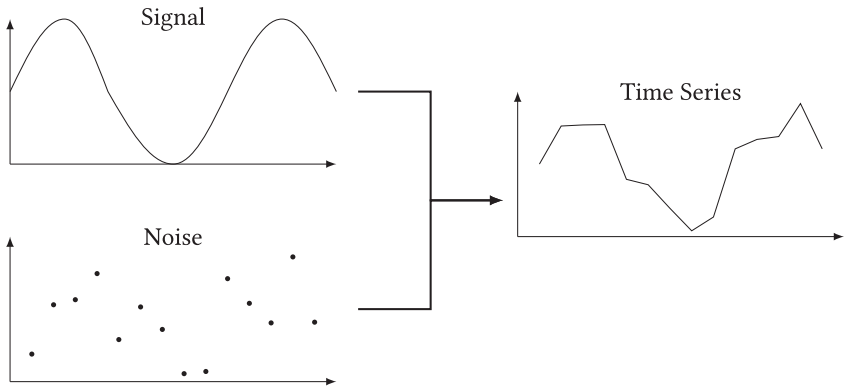


Fig. 3. A time series is a signal with white noise, i.e., a model used for time series prediction tries to estimate the signal behind the noise.

An evaluation using the Mean Absolute Percentage Error (*MAPE*) described in Equation (8), lead to a level of 50–60%, which is, according to Borzemski and Starczewski, too high for prediction:

$$MAPE = \frac{\sum_{i=0}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|}{n}. \tag{8}$$

Further investigation of smoothing methods for short-term TCP TPP was performed by Miller et al. [65]. The authors used different algorithms with multiple parameter combinations for predicting a video on demand transfer. In the process, Miller et al. focused on transmitting the video in the highest possible quality while minimizing transmission interruptions. According to their results, AM performs best and even better than more complex algorithms like the double exponential smoothing that Miller et al. described in their work [65]. In summary, it can be shown that the consideration of past values of the TCP TP increases the accuracy in predicting the future, but Mean-Based Models (MBMs) can only do this up to a certain level. Especially, when the knowledge of historical records should be considered, methods taking this history into account are needed. Therefore, we concentrate on this type of approach in the following.

4.3 Time Series Models

TSM are methods to analyze discrete temporal data. Forecasting future values of a time series is one of the most important tasks these models are facing in many areas. Therefore, it is hardly surprising that many of the known TSM were investigated for TPP. TSM algorithms interpret a **Time Series (TS)** as constant signal plus white noise with zero mean and finite variance, also called shock or innovation. This interpretation of a Time Series (TS) is illustrated in Figure 3. The goal of TSMs is to build a model for the signal behind the noise, and to predict the next value of the signal [68].

To achieve this, a generic model with different sets of parameters is trained in a so-called training phase. Since the validation of the training must be performed with similar data, there is typically a split of the recorded series into a training dataset and a test dataset and ideally an additional development set. Subsequent to the training phase, the model is applied on the test dataset. The predicted values are then used to validate the model with the measured ones from the test dataset. The whole process is illustrated in Figure 4, where y_i denotes the ith TCP TP of training set, $\vec{x}_i$ represents the vector of the corresponding past values and j is the index of the data points in the

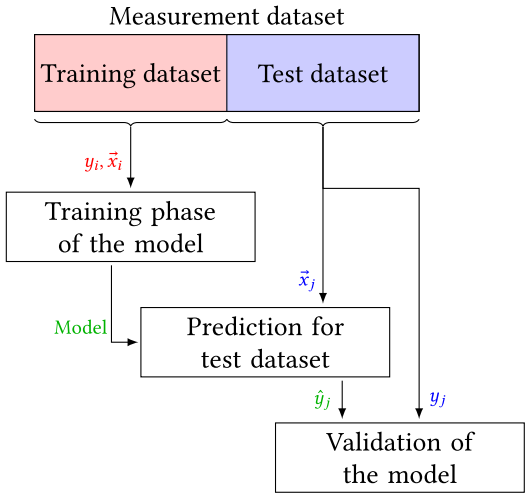


Fig. 4. Work flow used to develop a TSM model (based on References [133, 135]). First, the measured dataset is split into a training and a test dataset. This step is followed by the training of the model. Next, the model is used to predict the output value of $\hat{y_j}$ for the test dataset ($\vec{x_j}$). This prediction can be validated with the measured value y_j to assess the accuracy of the model.

test set. $\hat{y}_j$ is defined as the predicted TP for the jth position in the test dataset. For validation, the predicted values $\hat{y}_j$ can be compared with y_j.

To be able to use such models, the data must fulfill mainly two characteristics. First, the data points have to be in regular time intervals of the same length. This property means that either data must be recorded at equidistant intervals or pre-processing must be performed. Second, the data for most models must be stationary, which describes their statistical property in time. Montgomery et al. define stationarity as the following [68]:

(1) The probable values of the TS do not depend on time.
(2) The auto-covariance function defined as $Cov(y_t, y_{t-k})$ only depends on k and not on time.

One method for testing the stationarity of a TS is the Dickey–Fuller test [25]. The first TSM used for TPP is the AR model, which uses a weighted sum of the p previous values and its innovation to predict the future value. The number of p is called order, which means that a First-Order AR algorithm has $p = 1$. The model is depending on p and can be described using the following equation:

$$y_t = \mu - \phi\mu + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t, \tag{9}$$

where ϕ_1,ϕ_2,... are the weight parameters for the past values, ϵ_t is the value of the white noise and μ is the mean of the series. Karrer et al. [50] used this model to predict a set of more than 52,000 Internet TCP traces. They also had a look at four other models. One of them was the MA model, which predicts the current value by using the q previous values of the white noise as ϵ_{t-1} to ϵ_{t-q} weighted with Θ_1 to Θ_q plus the mean of the series μ, as shown in:

$$y_t = \mu + \epsilon_t - \Theta_1\epsilon_{t-1} - \cdots - \Theta_q\epsilon_{t-q}. \tag{10}$$

There is also the possibility to combine both models to a so called ARMA model (see Equation (11)), which is more flexible and depends on both parameters p and q. The order of the model can be defined as ARMA(p,q). Due to their flexibility, many time series can be modeled with $p, q \in (0, 1, 2)$ [34]:

$$y_t = \delta + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t - \Theta_1\epsilon_{t-1} - \Theta_2\epsilon_{t-2} - \cdots - \Theta_q\epsilon_{t-q}. \tag{11}$$

A proof that this model can be used to predict network traffic is given by Sang and Li [93], in which the authors also indicated a particular interest in the estimation of the lower and upper values of the prediction interval. The data processed by the ARMA model needs to be stationary. There is also an integrated version that can be used for non-stationary data. This is done by differentiating the TS d times before processing it with an ARMA model. The model is described as a function of Order of the Autoregressive Model (p), Order of the Moving Average (q) and d with the order $ARIMA(p, d, q)$.

A deeper analysis of Autoregressive Integrated Moving Average (ARIMA) models, regarding their usability for TPP was done by Zhani et al. [133]. They compared the ARMA(1,1) and the ARIMA(1,1,1) model on different datasets, but in most cases the models were similarly accurate. To improve this, Shu et al. [97] and Corradi et al. [21] used a fractal-ARIMA, which led to better results. This model implements a fractional integration given by the parameter d, which is also used in the ARIMA models. Zhani et al. also investigated the question of how many previous data points (lags) should be used and what is the optimal time interval for a data point. Their results showed that the traffic is only correlated to the last data point and the graduation of such a point is best by using 6 times the RTT. They have also proven that predicting longer data packets is easier.

Another approach is to combine the linear ARIMA model with a non-linear **Generalized Auto-Recessive Conditional Heteroskadasticity (GARCH)** model. The idea behind the Generalized Auto-Recessive Conditional Heteroskedasticity (GARCH) models is the assumption that if the last ϵ_t has a high value, which means a high change in TS, the next ϵ_t will also be high [13]. Since a GARCH model only works on stationary data, first, the integration parameter d needs to be estimated. Afterwards, algorithms like the maximum likelihood method can be used to estimate other parameters of the model. This ARIMA/GARCH approach was used by Zhou et al. [135] to predict even more than one step ahead. This was done by using a publicly available dataset of Internet TCP packages (Static Wired Scenario (S1)). The authors evaluated an error for a prediction timescale of 100 ms, as well as for 10 s. As evaluation metric the Signal to Error Ratio (SER) function was taken:

$$SER = 10log_{10}\left(\frac{\overline{y_t^2}}{\overline{(y_t - \hat{y}_t)^2}}\right)dB. \tag{12}$$

An extension to the ARIMA usage in TPP was shown by Torres et al. [106], the so-called Autoregressive Integrated Moving Average with Explanatory Variable model, which also considers seasonal effects for the estimation. Sadek and Khotanzad [89] used a k-factor Gegenbauer ARMA model to outperform the AR model. They applied their model to different types of data and predicted video streaming, as well as Ethernet, and internet traffic. For the evaluation, they used the Mean Absolute Error (MAE) (see Equation (13)) and the SER and came to the conclusion that for Internet data, the AR model can be improved by more than 40% in relation to the MAE performance:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_t - y_t|. \tag{13}$$

A look at all the approaches in this section shows that although most of them consider TP traces as stationary TS, there are also some models that are built for non-stationary TS, which may even be superior. Therefore, the question arises, whether TCP traces are stationary or not. To answer this question, Yoshida et al. [127] performed an analysis of HSPA, Long-Term Evolution (LTE), and Wi-Fi traces. They proved that the tracks contain both stationary and non-stationary parts. The authors collected throughput data produced by a 1-GB file download on different locations in

Tokyo. An evaluation using the Dickey–Fuller test [25] shows that the ratio of stationary to non-stationary parts depends on the used technology, but all measurements contain both. Taking this new knowledge into account, they created a stochastic model, which can handle both categories. The accuracy of the last 100 data points was tested by using the Root Mean Square Error (*RMSE*):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}. \tag{14}$$

4.4 Learning-Based Models

In general, LB algorithms can be seen as models, which allow the computer to learn from data. The learning process tries to extract a model from provided data samples. Depending whether a corresponding output or label is given, the LB methods can be categorized into supervised and unsupervised learning models, e.g., the clustering of unlabeled data is an unsupervised learning task, while a classification task using labelled training data as a supervised task. In the area of TPP, mainly regression tasks are used, which belong to supervised approach. These models return an absolute output value, based on a given input set. The building of such models is done in different steps. After a dataset has been recorded and pre-processed, it is split into three different subsets. The first and usually the largest one is called training set. It is used to train the algorithm and to build a model. The second one is the validation set for optimizing the model. Finally, there should also be a test set for evaluation. A detailed instruction how to create an LB project is, e.g., provided by Géron [35].

A further advantage of **Learning-Based Models (LBMs)** is their capability to use multiple input parameters. Compared to other models like MB or EB ones, Learning-Based Models (LBMs) can be easily applied to different sets of input parameters, which can significantly improve their performance as shown by Samba et al. [90, 91]. This process of selecting the correct input parameters for an LBM is called feature selection [101]. To highlight the impact of the feature selection, a study performed by El Khayat et al. [27] built models on the parameter set shown by Padhye et al. [76] as well as for an input set, which in addition used the timeout loss rate. They investigated, that the accuracy was improved by a factor of 2 to 3 depending on the model. Another investigation of datasets with different attributes was done by Borzemski and Starczewski [12]. They built nine sets containing the same data, but with different features to predict the TP during a web file download. There results are showing that the datasets with most features perform best for the Transfer Regression (TR) model. The same effect was investigated by Wei et al. [111]. Here, the authors compared **Support Vector Regression (SVR)** models using either the **Reference Signal Strength Indicator (RSSI)** of the LTE connection or the past throughput or both. Wei et al. evaluated the models in different scenarios like staying at a specific location, walking around or sitting in a bus or train. In every scenario, the Support Vector Regression (SVR) using both parameters performed better than the other ones.

Next, different LBMs used for TPP are presented in detail. One of the simplest methods are **Decision Trees (DTs)**. These models make a bunch of decisions based on the input parameters and can be used for classification or regression. A Decision Tree (DT) uses the input to build a tree structure. Each leaf is labeled with an output value. Depending on the depth of the tree, multiple decisions are leading to a prediction, which results in a certain accuracy at the output given by the training dataset. The accuracy for regression also depends on the number of output nodes, since each leaf can only represent one concrete value. Essentially, the regression task is a classification with a distinct number of classes. The flexibility of DT regarding TP is shown by El Khayat et al. [27], where the authors used the models not only for prediction, but also as an approach to recognize the overestimation of an EBM as given by Mathis et al. [64].

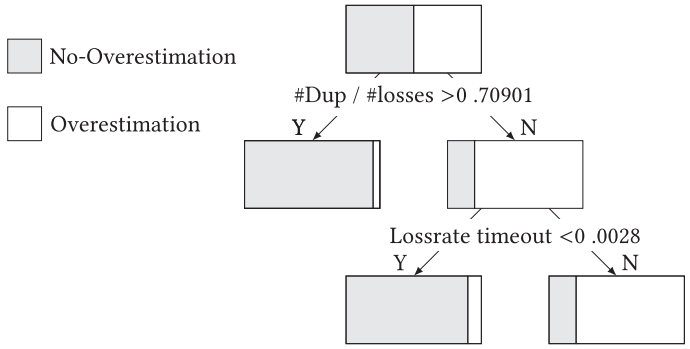


Fig. 5. Visualization of a decision tree used to predict overestimation of the equation-based model described in Equation (4), see Reference [27].

An example of such a DT is shown in Figure 5. DTs are a quite simple method in terms of prediction. They are also very popular, especially when they are used in ensembles. These are algorithms, which use multiple learning algorithms to reach an even better prediction. Some ensemble methods used for TPP were presented by El Khayat et al. [27], in particular Bootstrap aggregating, Extra-Trees, MART and Random Forest (RF). Another important task where regression trees are applied is identifying the number of past samples that are related to the predicted output. One possibility is the consideration of the correlation of the input time series [118]. Another application area is to analyze the relation of the input parameters [81].

A different ensemble technique used for TPP is Bootstrap aggregating, also known as bagging. It combines different models with an equally weighted vote. To obtain, e.g., different decision trees, bagging uses a randomized subset of the dataset provided for training [14]. An improvement of this technique is called RF [15], which changes the algorithm in a way that the sub-trees are less correlated. Furthermore, there is a method called Extra-Trees. In contrast to other tree-based ensembles, Extra-Trees use all the learning samples and choose the cut-points for their nodes totally randomly [31].

El Khayat et al. [27] compared these ensemble methods against the EB approaches shown in Equations (1) and (4). They used a simulation environment that generates random topologies with 10–600 nodes, a bandwidth between 56 kb/s and 100 Mb/s and a delay variation starting from 0.1 ms up to 500 ms. The evaluation criterion used was the Mean Square Error (*MSE*) given in Equation (15):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2. \tag{15}$$

In addition, the regression Coefficient of Determination (R) was evaluated. It indicates the confidence of the model. If it is close to zero, then the model is less accurate, but if it is close to 1, the regression is near the measured values. The coefficient of determination is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(\bar{y} - y_i)^2}. \tag{16}$$

With the average $\bar{y}$ of all y_i ($i \in [1, \ldots, n]$), R^2 can also be determined as 1, minus the ratio of the mean square error to the variance. The results of El Khayat et al. show that bagging, Extra-Trees and MART perform significantly better than EBMs but are less powerful than the Neural Networks (NN) explained later in this section.

Another investigation of RF was carried out by Samba et al. [90, 91]. They used RF to predict the instantaneous TP on a connection during establishing. Therefore, they investigated different sets of LTE low-level parameters as well as other features collected by the mobile device and came to the conclusion that by taking into account also data from the operator, much better results can be achieved. For the evaluation, the R^2 function, given in Equation (16) was used.

This is very similar to the work done by Ghasemi [32]. Here the authors showed how crowd-sourced mobile spectrum data can be used for TPP. Therefore, a memory-efficient gradient boosted tree algorithm called LightGBM was studied, as well as a dataset collected with the mobile phone application `OpenSignal`, which contains LTE parameters and the TP for up- and download. As an evaluation metric the *RMSE* was performed. Samba et al. [90] also showed that a comparison between RF and NN provided a similar accuracy for both. All their tests were performed in a static mobile network scenario (S2). RF models can also be taken in S3 scenarios. Yue et al. [129] highlighted an approach using this method in different S2 and S3 situations, with a different number of inputs, starting from the univariable TP up to a model taking additional LTE low-level values into account. Their study shows that more input parameters as well as a stationary scenario are leading to better results regarding the Relative Error (*RE*).

Borzemski and Starczewski [12] have shown TPP using TR. The TR is based on the Kolmogorov superposition theorem [77], which says that for every integer dimension $d \geq 2$, there exists a constant real function $h_{ij}(x)$ defined on the interval $U = [0, 1]$. In addition, the TR shows that for every d-dimensional function $f(x_1, \ldots, x_d)$ defined on the hypercube U^d, there exists a function $g_i(x)$. It allows reducing a very complex function with multiple variables to a term of univariate functions. The TR model is given by Equations (17) and (18):

$$\hat{y}_1 = \sum_{j=1}^{d} h_{1j}(x_j), \hat{y} = \sum_{i} \hat{y}_i, \tag{17}$$

$$\hat{y}_i = \sum_{j=1}^{d} h_{ij}(x_j; \hat{y}_1, \ldots, \hat{y}_{i-1}) + \sum_{k=d+1}^{d+i-1} h_{ik}(\hat{y}_{k-d}; \hat{y}_1, \ldots, \hat{y}_{i-1}), i > 1. \tag{18}$$

For evaluation, Borzemski and Starczewski compared their results against the WMA and EWMA models described in Section 4.2. By using the *MAPE* function, they showed that their MA model reaches an accuracy of 50–60%, which is insufficient for using it as a predictor. The TR model is able to use other inputs, apart from the past TCP TP values. So, it achieves an error of 20–22% by further considering file size, loss rate, time of the day, and other parameters.

A different learning technique are Markov models. They are mainly chosen for topics related to TPP, e.g., for analyzing the stationary of TP series [109], the bit rate selection in video streaming [6, 104] or for the prediction of the amount of traffic to be transmitted [102]. For TPP, Markov models use the current TP of the client as a state for their state machine. Each state has a calculated TP and a probability to transition to another state. Hence, taking Figure 6 into account, at one-step prediction starting a state s_0 can be calculated according to the following equation:

$$\hat{y}(s_0, 1) = ps_{00} \times y_{s_0} + ps_{01} \times y_{s_1} + ps_{02} \times y_{s_2}. \tag{19}$$

The parameter ps is introduced for the property that a state switch happens and y_{s_i} is the calculated TP for state s_i. Additionally, an approach considering a **Hidden Markov Model (HMM)** is given by Wei et al. [110]. There, a two-step method was developed, to stream videos via two connections. In the first step, the TS was classified using SVM. To make a decision, the next TP should be predicted using an HMM or an AR model. This two-step approach is very flexible and achieves an

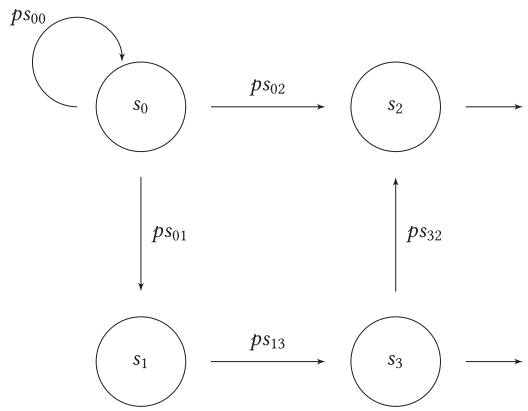


Fig. 6. Visualization of a Markov model to describe the predicted TP as a state and the changing of TP as a probability for switching this state.

accuracy between 76% and 93.33% depending on the scenario S2 or S3. To measure this accuracy, the Root Mean Square Relative Error (*RMSRE*) defined in Equation (20) is performed,

$$RMSRE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y_i}}{y_i}\right)^2}. \tag{20}$$

A further popular method used in the context of TPP is SVR. This is a powerful LB technique, which has shown good results. SVRs take multiple input variables to generate a prediction, e.g., Mirza et al. [66] employed the loss rate, the length of the TCP stream and the available bandwidth on the path to predict the TP. SVRs are also not using any particular parametric form like EBMs. Instead, so-called kernels can be used, which give high flexibility and allow the building of predictors with better accuracy. The kernel function is defined as $K(\vec{x}_i, \vec{x}_j)$ with $\vec{x}_i$ and $\vec{x}_j$ as the ith and jth input vector of the dataset. Finally, compared with other LBMs like NN, SVR can be very computing efficient. The generic model of SVR is described in Equation (21), where $\hat{y}$ is the prediction for the input vector $\vec{x}$. The model parameters a and a^* are the result of the optimization problem solved in the training phase. A more detail description of SVRs is provided by Smola and Schölkopf [100],

$$\hat{y} = \sum_{i=1}^{N}(a_i - a_i^*) \times K(\vec{x}_i, \vec{x}) + b, \tag{21}$$

where $K(\vec{x}_i, \vec{x})$ is the kernel function, which makes the basic SVR model more flexible. The different kernels studied for TPP in the work of Hu et al. [41] are as follows:

- Linear Function:
  $K(\vec{x}_i, \vec{x}_j) = \vec{x}_i^T \vec{x}_j$.
- Polynomial Function:
  $K(\vec{x}_i, \vec{x}_j) = (\gamma(\vec{x}_i^T \vec{x}_j) + a)^b, \gamma > 0, a \in \mathbb{R}, b \in \mathbb{N}$.
- Radial Basis Function:
  $K(\vec{x}_i, \vec{x}_j) = exp(-(\gamma \|\vec{x}_i - \vec{x}_j\|^2), \gamma > 0$.
- Sigmoid Function:
  $K(\vec{x}_i, \vec{x}_j) = tanh(\gamma(\vec{x}_i^T \vec{x}_j) - c), \gamma > 0, c \in \mathbb{R}$.

An extension of the SVR is the Nu-SVR or ν-SVR model, which was introduced by Schölkopf et al. [96]. In this algorithm, the parameter ν is introduced to control the number of support vectors. This gives the possibility to eliminate the accuracy parameter ϵ taken in SVR models. Of course, also ν-SVR models are able to use the different kernels. Lee et al. [54] use a ν-SVR with a polynomial kernel to predict the TP of different wide area network connection pairs. They investigated the effect of the polynomial kernel degree between 2 to 5 and consider the 3-degree polynomial kernel as a reasonable choice for their predictor. The ν-SVR model was analyzed by Hu et al. [41]. Furthermore, there also was an approach from Nicholson and Noble [73], where they utilized it for one-step-ahead prediction as well as for multi-step-ahead prediction using a Radial Basis Function kernel. As input, a vector of the past 10 throughput values was taken and an optimization as well as a cross-validation was executed. They achieved a smaller *MAE* compared to their NN model, which was built with the same inputs.

NNs are very popular in research at the moment. They are also frequently studied for regression tasks [53] as well as for related tasks like dynamic bandwidth allocation [99], so it is not surprising, that they have also been investigated in the area of TPP and first used for comparison as shown by Borzemski et al. [11]. According to their name, NNs are networks of so-called neurons. A neuron is a binary unit that computes a weighted sum. This sum is the input of the activation function that calculates the output of the unit. Such a neuron can be defined as $y = (\sum_{i=1}^{n} w_i x_i) - u$, where n is the number of inputs x_i, $i = 1, 2, \ldots, n$ and w_i the weight for every input. The threshold is defined by u. Activation functions can be linear, sinusoidal or gaussian. Since the function has a huge influence on the output, it is important to choose it carefully. The network itself is structured in layers. Each layer has a given amount of neurons and the more layers a network has, the deeper it is. There are three types of layers. The input layer, which is the first layer. It directly receives all inputs. The output layer, that returns the results and at least one hidden layer for the calculation. A more detailed description of NN is given by Jain et al. [44]. One approach using these models for generalization of an EBM is given by Rao et al. [87].

The α-SNF model is a combination of fuzzy logic and NNs [133]. It uses the flexibility of fuzzy logic and the learning ability of NN. The general structure of an α-SNF is the same than for a NN, but the neuron output is calculated differently. The fuzzy system used in such a neuron is a non-linear relation between the inputs $\vec{x}$ and the output y of the neuron. This relation is a collection of C fuzzy rules, where R_k denotes the kth rule in $1 < k < C$. A rule R_k is defined in Equation (22). A more detailed description of α-SNF is given by Abed Rouai and Ben Ahmed [1] and Zhani et al. [132],

$$R_k : \; if \; (\vec{x}) \; is \; A_k \; then \; Y_k \; is \; b_k. \tag{22}$$

Another method already investigated for similar tasks [116], are special modern NN model. In the so-called Recurrent Neural Network models, there exist different types of recurrent neurons. The Long Short-Term Memory is the most widespread. A model using this type of NN was studied by Wei et al. [112]. They evaluated four different scenarios ranging from static mobile network traffic, walking up to bus and train trips. As an evaluation function the *NRMSE* was performed. Their results indicated the importance of data preprocessing when using NNs. With their model, Wei et al. were able to outperform different TSM and MB approaches.

4.5 Location Smoothing Models

Apart from time smoothing approaches, in the last years, there was also a significant increase of location smoothing LS prediction, which can be attributed to the studies done by Yao et al. [123] showing the impact of the location for the prediction. Yao et al. investigated that previously recorded data of at the same location, to tells more about the future TP than current measures of

a different location, which is also proposed by Martínez et al. [62]. Yao et al. call this result *Past Tells More Than Present*. One way to set up such a collection is to send the data to a server via the mobile network immediately after their measurement [122]. The main difference between the approaches described in Section 4.1 to 4.4 and Location Smoothing Models (LSMs) is that these algorithms are not time-based. Accordingly, they predict the TP for a concrete time in the future; this automatically includes the prediction of the future position. However, LSMs are used to predict the TP for a particular location and not for a subtend data points in a TS.

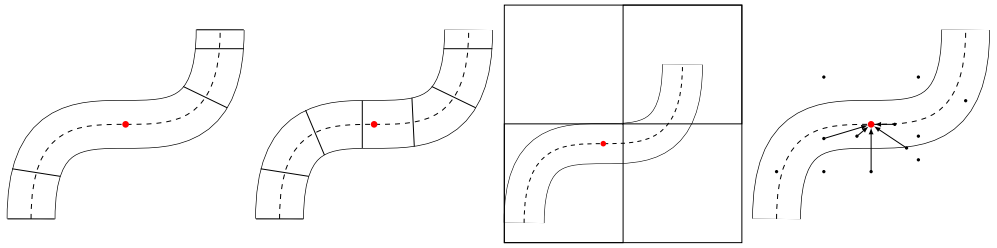
LSMs can be separated into two classes. There are aggregated maps, which use the historical data to pre-calculate a model for a defined area, like a segment describing a part of a road. In these kind of models, new measurements are used for updates. The other approach is a method that uses past measurements directly without pre-processing. These models typically take a certain amount of data points around the location that shall be predicted and compute the prediction online, e.g., by interpolating. These approaches have in common that for predicting the throughput also the future location needs to be predicted, which can be a challenging task on its own [131].

The aggregated map differs in the way the segments are built and the methods, which are used to calculate the prediction. One possibility is to use segments, which are already defined by the map provider. Such a model was proposed by Pögel and Wolf [82] as well as by Kelch et al. [51]. One drawback of this approach is that road segments differ a lot in size. The authors used an upper boundary of 50 m to compensate this fluctuation, but since there are also segments, e.g. at intersections that are only few meters in length, there are still large differences. This problem can be solved by using self-defined segments with a fixed length as shown in the works [69, 122, 126]. Here, fixed size segments of 500 or 1000 m were used. These lengths result from the fact that for every segment a measurement probe should be taken, so the segments need to be long enough, to have time to record a measurement [126]. In addition, e.g., Yoa et al. [126] expand their initial size of 200 m, since there are some segments containing no probe, which led them to choose a length of 500 m.

Of course this is less flexible, since a test track needs to be a multiple of this length. Before calculating a prediction for such a segment, it must first be checked whether a sufficient number of data points has been collected. A suitable method for this is an entropy analysis, as proposed by Yao et al. [121]. Afterwards, this calculation can be done by building the mean and the standard deviation [69] or by using another mean-based prediction like the EWMA, which takes the newer samples with higher weights into account [126]. Since the construction of segments causes additional costs, an approach that uses a grid-based map can be advantageous [28]. Such a model does not require road matching for raw **Global Positioning System (GPS)** points as shown by Quddus et al. [84], and the calculation of a EWMA for a grid is much easier, because of its simple geometry. A more detailed description of the different approaches is given in the following.

4.5.1 Map with Flexible Segment Length. To build a TP map, it can be very beneficial to use already defined structures like road parts from a map provider. This is helpful, especially if the data shall be used in automotive use case, e.g., connected car scenarios. Pögel and Wolf [82] presented an approach using the segments in the **Open Street Map (OSM)** data. These contain open source map data that are maintained by the Open Street Map (OSM) community. Since the road parts are not created with any constraint in length, they can span from a few meters up to some kilometers. Aggregation of all measurements of such a segment as shown in Figure 7(a), can lead to inaccuracy, since the number of measurement points is normally related to the segment length and the aggregated value is always a forecast for the whole segment.

4.5.2 Map with Fixed Segment Length. To avoid the problem of road segments with different lengths, fixed length segments can be used as shown in Figure 7(b). For creating a map with such



(a) Road segments of flexible length, e. g. segments from a map provider.

(b) Road segments of fixed length, built by recalculating the roads from a map provider.

(c) Grid based approach. The area can be split by building a simple grid.

(d) Nearest neighbour measurements. This approach calculates the prediction based on relevant measurements in the past.

Fig. 7. Different types of geographic-based models used for throughput prediction. The red dot shows the point, which should be predicted. The black areas and points indicate the geometries used to calculate a prediction.

segments, additional effort is required to define them, and it leads to the problem that only tracks can be chosen for prediction, where these segments are defined. To efficiently build a predictor using this map, equidistant TP measurements should be used.

4.5.3 Grid-based Map. A compromise between equidistant map structures and the effort for their construction is the use of a grid as illustrated in Figure 7(c). On the one hand, grids can be built easily even for the whole world, and, on the other hand, they define a kind of boundary regarding the structure length. They also have the advantage that matching inaccurate raw Global Positioning System (GPS) point to a road is not absolutely necessary. Another aspect is the handling of parallel roads. A road segment–based map would create independent predictors for the parallel roads. In a grid-based approach, there is the same prediction for these two segments if they are in the same cell of the grid. Another approach that also ends up in a grid-based map is given by Kamakaris and Nickerson [49]. There, the authors started with a contour throughput map as an ideal approach. Then, they built a grid-based overlay and labelled the fields with the map values. A first combination of LS and LB methods is presented by Sliwa et al. [99], where they use the grid cell as an input parameter for their NN.

4.5.4 Nearest Neighbour Measurements. There are also methods for online calculation of TPP, which are mainly based on past measurements (see Figure 7(d)). The simplest model is to calculate the mean value from all data points at a certain distance to the point that shall be predicted [22, 74]. To take also the time into account, there are models, which only count the last days, as proposed, e.g., in the work of Evensen et al. [29], or divide the data into time slots based on the hour of the day [36]. An algorithm to use such nearby measurements to predict a whole route between cities was investigated by Riiser et al. [88]. They were sampling a track into equidistant points with a distance of 100 m and sent these locations to a server, which calculated the mean and standard deviation for each of these points. The calculated values can be used by the client, e.g., to schedule the quality of a video stream. Another method to select the samples used for the prediction is the k-nearest neighbour algorithm, which returns the nearest locations (nl_1, nl_2, …, nl_k) to the location pl that should be predicted. These points can be used to calculate a bandwidth value. Hao et al. [36] applied this method to develop an Inverse Distance Weighted interpolation, which took

the distance of the location nl_i to the point to be predicted Prediction Point (pl) into account. This technique is shown in the following equation:

$$\hat{B}(pl) = \sqrt[n]{\sum_{i=1}^{k} \left(\frac{B(nl_i)}{dist(nl_i, pl)} \right)^n \Big/ \sum_{i=1}^{k} \left(\frac{1}{dist(nl_i, pl)} \right)^n}. \tag{23}$$

Taani and Zimmermann [105] advanced this method by using Kriging, which is a geo-statistical method applied for interpolating a value based on measurements around. Kriging is also used in simulation topics like the generation of radio coverage maps [23]. A detailed explanation of the Kriging method is available in the work of Heuvelink et al. [39]. The general formula for a bandwidth prediction technique based on Kriging is shown in Equation (24):

$$\hat{B}(pl) = \sum_{i=1}^{k} \lambda_i B(nl_i), \tag{24}$$

where λ_i is the unknown Kriging weight for location i. Taani and Zimmermann also compared this model against the methods shown by Riiser et al. [88] and Hao et al. [36] using the mean error. Their results show that the Kriging model outperforms both.

4.6 Comparison

After a detailed analysis of the different approaches, this paragraph is giving a comparison of the model categories shown in Sections 4.1 to 4.5. Although it is difficult to compare them directly, because the procedures differ in many characteristics, such as granularity or the error methods used for evaluation, nevertheless, a general conclusion can be drawn with regard to a few categories of methods. The first category, EB, differentiates from the others because of the fact that they make their changes in a very high frequency, after each round of TCP segments. Since they are implemented in TCP CC and TCP is widely used, they are constantly being enhanced. Next, the MBMs can be considered. They have the advantage that they only need the last data points and can be used without previous training, but their prediction is also very inaccurate, so they are outperformed by TSM as shown by Wei et al. [112]. These classic models for predicting TS are divided into two groups, stationary and non-stationary ones. But since the data of wireless links like High Speed Packet Access (HSPA), LTE, and Wireless LAN Media Access Control and Physical Layer (Wi-Fi) contain both stationary and non-stationary parts as investigated by Yoshida et al. [127], combined models like the ARIMA/GARCH approach, which was used by Zhou et al. [135] should be used. The third category, LBMs, offer the possibility to include other parameters for the prediction besides the TP as well. If the input values are pre-processed correctly, then the performance of the methods also exceeds the performance of MB algorithms or TSM, as shown by Wei et al.. Another feature of the TS models is that they are location-independent during the prediction of mobile wireless connections. This distinguishes them from the LS methods that use the location reference from previous measurements to predict mobile network connectivity. LS methods require either the storage of all previous measurements, or a map must be created. A comparison of LS and LB approaches is one of the open points as shown in Section 6.

Another aspect is the changed usage of data networks. Since traffic load as well as capacity have significantly increased in the last years, the results of early prediction methods are not directly comparable with new ones.

Table 3. Collection of Publicly Available Datasets That Can Be Used for TPP in the Scenarios (Sce.): Static Wired Scenario (S1) and Dynamic Mobile Network Scenario (S3)

Reference	Sce.	Data Point	Usages	Features
Zhou et al. [135][1]	S1	18 000 k	congestion control, network management	TCP package timestamps, source host, destination host, source port, destination port, package size, TCP flags, sequence number, acknowledgement number
Yao et al. [121][2]	S3	26 k	improving mobile internet performance	timestamps, location, bandwidth
Bokani et al. [8][3]	S3	15 k	adaptive video streaming	timestamps, download time, download rate, download size, start location, end location, network type, operator information
Jomrich et al. [47][4]	S3	53 k	enable highly automated driving, associate comfort services for the driver	timestamps, cell ID, tracking area, network type, provider information, signal strength, RSIP, RSSI, TP, device identifier, start location, end location, speed, number of packages, packages loss, weekday, download or upload, …

The table is sorted by reference. The number of data points, purpose of use, as well as the recorded features are given.

5 AVAILABLE DATASETS AND MEASUREMENT TOOLS

Section 4 presented the different use cases and scenarios, in which TPP is relevant. It also pointed out that some models need more input parameters than others. To obtain such an input dataset, e.g., for the development of a suitable approach, there are essentially two possibilities. The first one is to use a publicly available dataset, the other one is utilization of a measurement tool.

The first option, the collection of open datasets is described in the following and summarized in Table 3. Zhou et al. [135] use two hours of recorded traffic between the Lawrence Berkeley Laboratory and the World Wide Web. The measurement was done with the well-known tool *tcpdump*, which is able to store the sent and revised TCP packages to a binary file. For the conversion between raw data and the provided dataset, the authors used a bunch of scripts that are also available. Thus, it is not only possible to use their dataset for benchmarking, but to create a similar one. This is beneficial, as the data depend on the Internet connection. The features of the set are low-level TCP package parameters and header information. To use them for TPP, a pre-processing will be needed for calculating the TP, loss rate or other higher level parameters. Furthermore, no further features like physical transmission characteristics or routing buffers are recorded.

[1] http://ita.ee.lbl.gov/html/contrib/LBL-TCP-3.html.
[2] https://github.com/aubokani/Bandwidth-Dataset/blob/master/Sydney_bandwidth_2008.zip.
[3] https://github.com/aubokani/Bandwidth-Dataset/blob/master/Sydney_bandwidth_2015.zip.
[4] https://github.com/florianjomrich/cellularLTEMeasurementsHighwayA60.

Table 4. Collection of Publicly Available Tools That Can Be Used for TPP in the Scenarios (Sce.): Static Wired Scenario (S1), Stationary Mobile Network Scenario (S2), and Dynamic Mobile Network Scenario (S3)

Reference	Sce.	Usages	Features
Yu et al. [128]	S1	estimation of bulk transfer capacity	IP, URL, Download file size, Download speed
De Silva et al. [24]	S2, S3	performance prediction for mobile devices	Wifi/LTE, Location, *RTT*/Throughput, Timestamps
Torres et al. [106]	S3	forecasting the average downlink throughput in vehicles	Device ID, Provider ID, RSSI, RSRQ, RSRP, Frequency band, LAC, Cell ID, IP, Location, *RTT*, Throughput, …

The table is sorted by reference. The purposes of use as well as the recorded features are given.

The other datasets of this section are related to the mobile network, but differ in number of features and network type. While older sets like the one shown in Yao et al. [121] are recorded with a vehicle PC using 3G hardware, newer ones are created using a smart phone with LTE. Looking at the provided features, the dataset from Yao et al. contains timestamps and location information measured by a GPS sensor as well as the corresponding bandwidth. It is therefore a lightweight set. A more accurate dataset regarding the location is provided by Bokani et al. [8]. They recorded the start and end position of the data transfer, and not merely a location information during transmission. This led to the fact that a throughput is assigned to a road segment and not to a single point. For building a multivariable model, which also takes LTE low-level parameters into account, the dataset of Jomrich et al. [47] should be used. Apart from the large amount of different features, the authors have already prepared the dataset for the utiliztion of machine learning approaches. New LBMs can, therefore, be easily implemented. The dataset also contains measurements performed on different hardware and various providers.

The second method for collecting input data is to use a measurement tool, shown in Table 4. One of the first available tools was *Pathperf* [128], which is capable of estimating the bandwidth of a path. Another possibility to record the raw network traffic and extract the needed parameters afterwards is shown above. Due to the increase of mobile network and smartphone applications, meanwhile, there are further tools available, which are able to measure the TP and GPS position. One of them is shown by De Silva et al. [24]. Such tools have in common that their measurements are limited to the upload and download TP as well as the ping time. Consequently, no LTE parameters are provided. There are also tools in place, which can measure LTE low-level parameters, but these do not record the TP. If position dependent, then lower level mobile network parameters and the TP are needed.

One open source tool is publicly available. It is presented by Torres et al. [106] and described their measurement setup developed in the European H2020 Research Project MONROE. It comprises four LTE models as well as a computing unit and an additional Raspberry PI. The software is based on the micro service technique and allows to measure LTE parameters, GPS location and network parameters. In addition, there are publications describing how to implement such a tool. You et al. [120] showed a measurement setup, which is capable of creating datasets similar to the one used by You et al. [121]. Other authors, e.g., Martínez et al. [62] and Pögel et al. [79] showed approaches capable of measuring features from different input sources like the network interface and the low-level model parameters. The paper of Schmid et al. [94] presented a tool, which is able to store a dataset on disk as well as to process the recorded samples of LTE, TCP, and GPS

values immediately. Of course, there are a number of tools like Netradar [71] or SamKnows [92] that provide performance measurements at the application level, but since these do not provide TCP data, they have not been studied in this work. The RIPE Atlas [4] platform also falls into a similar category, although it provides detailed data on a number of measurement types. But since it measures application protocols such as DNS and HTTP, it is unfortunately not suitable for the prediction of TCP values.

6 OPEN ISSUES AND FURTHER WORK

Although the previous sections are showing the work done in the area of TPP, there are still open challenges to solve. A summary of the most important aspects is provided in the following subsections.

6.1 Open Dataset for Benchmarking and Evaluation

Comparability of the individual approaches is a key topic. The table presented in the online only material shows that many models are evaluated using different functions or methods, which makes it difficult to compare them against each other. Due to the fact that various datasets with different inputs were used for evaluation, the comparison of research results becomes even more difficult, since effects such as the correlation between TP and TCP flow size cannot be proven as shown by Dong et al. [26]. To solve this problem, a dataset is needed that fulfills at least the following requirements: First, it should concentrate at least on one of the three scenarios defined in Section 3. Apart from this, it needs to contain all features that are required by the different algorithms applied in that scenario. And of course, the evaluation must be done with the same method or error function.

6.2 Comparison of Location and Time-based Approaches

Another topic concerns the comparability of LB and LS prediction methods. If LBMs are chosen to predict the throughput of a moving client, then the prediction normally includes the estimation of the future position of the client, which can become inaccurate if the model is only trained on known routes and evaluated on other ones. The problem becomes even more obvious, when LS prediction is applied, because of the need of a location for the prediction. To overcome this problem, one approach is to assume that the path of the client is known as proposed in the work [40]. This can work quite well in a public transport environment like a train. But it may not work well for the usage of passenger cars or walking pedestrian scenarios. Here, a suitable approach is to make a look ahead or to cover whole predictions as done by Singh et al. [98]. An alternative is to estimate the probability of a location change. We already showed a comparison between a grid-based map and different LBMs in our previous work [95], in which we considered the future location as given, so the location prediction error is ignored, which is a good starting point. In summary, finding a suitable method to compare location and time-based approaches including all aspects is still an open issue.

6.3 Clearer Description of Measurement Environment

Looking in the area of mobile networks, the continuous development has a huge impact on the TPP and the correlation between parameters [108]. In LTE, e.g., carrier aggregation probably will have a significant influence, so not only the client device and scenario need to be described in a dataset or publication, but also the supported technologies of the network providers need to be mentioned, to describe the overall setup. This becomes even more relevant in 5G, where continuous evolution is planned.

6.4 Investigation on Multiple Server Scenario

Still open is the topic of a client being connected to multiple servers at the same time. There were investigations on the similarity of connections done by Vazhkudai and Schopf [107], but some additional work is required in this area, e.g., the generation of a public dataset that was mentioned in Section 6.1. An investigation of this scenario is also very important, since the connection to multiple servers is more frequent than communication with just a single server.

6.5 Investigation on the Environment Influence

The influence of the environment such as forests or buildings is well studied for radio-related topics. Up to now, these effects are not considered in the area of TPP. Location-based methods taking these effects into account should become more accurate.

6.6 Investigation on Recent Developments in LB Techniques

Another interesting approach is given by recurrent machine learning models, which are from our point of view understudied. These methods can be used to build very accurate models in other regression tasks [30, 113], and should therefore preferably be more considered for TCP TPP. In particular, their memory and representation learning capabilities as given by gated recurrent units in combination with convolutional neural networks offer strong potential. Generative adversarial networks further allow to learn from limited observation data.

6.7 Investigation on Combining Approaches

Further, there are approaches that combine different algorithms, e.g., shown by Wei et al. [110] and Madan and Sarathi Mangipudi [60] combining an LB method classification with a TSM algorithm. There is still a lack of suitable combined approaches. In particular, merging of LS and multivariable-based LB methods could offer strong potential, since previous works [95] showed that depending on the evaluation metric either LS or LB methods are performing better. So a combination of both should outperform the single approaches. However, only a few scientifically studies started yet, e.g., Sliwa et al. [99], which are investigating a technique to combine LBMs with a grid-based map. Even if this approach shows the potential of linking LB and LS, there are currently no approaches to combine more recent methods.

7 CONCLUSION AND OUTLOOK

In this work, we distilled the most relevant approaches regarding TCP throughput prediction across the vast range of models, starting from equation-based analyses, over instant time smoothing, to learning-based or location smoothing models. Furthermore, available datasets and tools were presented, and open topics and challenges were pointed out.

This survey showed that research is currently focusing more on mobile network applications and thus on S3. This is especially true for data traffic between data centers and can be explained by the increased use of server services in mobile networks. Related to the prediction algorithms, a clear trend in the direction of LBMs at LSMs could be shown for a mobile scenario S3. In S1, however, pure LBMs or new variants of TCP are increasingly used.

Nevertheless, up to now, there is nearly no approach comparing their properties and performances yet. Since both categories have their pros and cons, we feel confident that a method combining them could outperform both. Therefore, this is definitely a development path that will evolve in the future.

Considering the research in this area, it should be possible to transfer higher priority data for the client without violating net neutrality as well as detecting areas where the mobile network Quality

of Service (QoS) has to be improved. Given such improvements, we envision the next generation of intelligent vehicles to be very well aware of throughput.

8 USED ANNOTATIONS

In this section, annotations and parameters used in the survey are defined. In general, inputs are described with x and outputs with y. A predicted value is marked by a Circumflex, e.g., $\hat{y}$. Looking into the area of learning-based models, where multiple inputs are used, the collection of these features is described by vectors. So, the input of these models is $\vec{x}$. Also, commonly used is B for bandwidth and TP for throughput. The main difference between the terms bandwidth and throughput is that bandwidth is used for the capacity of the whole network link, while throughput stands for the capacity of a single client to server connection. Additionally, especially in Section 4.1, some further symbols are introduced, e.g., RTT for round trip time and T_O for timeout.

REFERENCES

[1] F. Abed Rouai and M. Ben Ahmed. 2001. A new approach for fuzzy neural network weight initialization. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'01)*, Vol. 2. IEEE, 1322–1327.https://doi.org/10.1109/IJCNN.2001.939553

[2] Martin Arlitt, Balachander Krishnamurthy, and Jeffrey C. Mogul. 2005. Predicting Short-transfer Latency from TCP Arcana: A Trace-based Validation. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement (IMC'05)*. USENIX Association, Berkeley, CA, 19–19. http://dl.acm.org/citation.cfm?id=1251086.1251105.

[3] Arash Asadi, Qing Wang, and Vincenzo Mancuso. 2014. A survey on device-to-device communication in cellular networks. *IEEE Commun. Surv. Tutor.* 16, 4 (Apr. 2014), 1801–1819. https://doi.org/10.1109/COMST.2014.2319555

[4] RIPE Atlas. 2021. Retrieved from https://atlas.ripe.net/.

[5] Ayub Bokani. 2014. Location-Based adaptation for DASH in vehicular environment. In *Proceedings of the 2014 CoNEXT on Student Workshop (CoNEXT Student Workshop'14)*. ACM, 21–23. https://doi.org/10.1145/2680821.2680836

[6] Ayub Bokani, Mahbub Hassan, and Salil Kanhere. 2013. HTTP-based adaptive streaming for mobile clients using Markov decision process. In *Proceedings of the 20th International Packet Video Workshop*. IEEE, 1–8. https://doi.org/10.1109/PV.2013.6691443

[7] Ayub Bokani, Mahbub Hassan, Salil Kanhere, and Xiaoqing Zhu. 2015. Optimizing HTTP-based adaptive streaming in vehicular environment using markov decision process. *IEEE Trans. Multimedia* 17, 12 (Oct. 2015), 2297–2309. https://doi.org/10.1109/TMM.2015.2494458

[8] Ayub Bokani, Mahbub Hassan, Salil S. Kanhere, Jun Yao, and Garson Zhong. 2016. Comprehensive mobile bandwidth traces from vehicular networks. In *Proceedings of the 7th International Conference on Multimedia Systems (MMSys'16)*. ACM, 44. https://doi.org/10.1145/2910017.2910618

[9] Ayub Bokani, S. Amir Hoseini, Mahbub Hassan, and Salil S. Kanhere. 2015. Empirical evaluation of MDP-based DASH player. In *Proceedings of the International Telecommunication Networks and Applications Conference (ITNAC'15)*. IEEE, 332–337. https://doi.org/10.1109/ATNAC.2015.7366835

[10] Ayub Bokani, S. Amir Hoseini, Mahbub Hassan, and Salil S. Kanhere. 2016. Implementation and evaluation of adaptive video streaming based on Markov decision process. In *Proceedings of the IEEE International Conference on Communications (ICC'16)*. IEEE, 1–6. https://doi.org/10.1109/ICC.2016.7511226

[11] Leszek Borzemski, Marta Kliber, and Ziemowit Nowak. 2008. Application of data mining algorithms to TCP throughput prediction in HTTP transactions. In *Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Ngoc Thanh Nguyen, Leszek Borzemski, Adam Grzech, and Moonis Ali (Eds.). Springer, Berlin, 159–168. https://doi.org/10.1007/978-3-540-69052-8_17

[12] Leszek Borzemski and Gabriel Starczewski. 2009. Application of transfer regression to TCP throughput prediction. In *2009 Proceedings of the 1st Asian Conference on Intelligent Information and Database Systems*. IEEE, Dong Hoi, Vietnam, 28–33. https://doi.org/10.1109/ACIIDS.2009.74

[13] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. 2016. *Time Series Analysis: Forecasting and Control* (5th ed.). John Wiley & Sons. OCLC: 923448270.

[14] Leo Breiman. 1996. Bagging predictors. *Mach. Learn.* 24, 2 (Aug. 1996), 123–140. https://doi.org/10.1007/BF00058655

[15] Leo Breiman. 2001. Random Forests. *Mach. Learn.* 45, 1 (Aug. 2001), 5–32. https://doi.org/10.1023/A:1010933404324

[16] Nicola Bui, Matteo Cesana, S. Amir Hosseini, Qi Liao, Ilaria Malanchini, and Joerg Widmer. 2017. A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques. *IEEE Commun. Surv. Tutor.* 19, 3 (Apr. 2017), 1790–1821. https://doi.org/10.1109/COMST.2017.2694140

[17] Nicola Bui, Foivos Michelinakis, and Joerg Widmer. 2014. A model for throughput prediction for mobile users. In *European Wireless 2014; 20th Proceedings of the European Wireless Conference*. VDE, VDE, Barcelona, Spain, 1–6.

[18] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. 2017. BBR: congestion-based congestion control. *Commun. ACM* 60, 2 (Jan. 2017), 58–66. https://doi.org/10.1145/3009824

[19] Neal Cardwell, Stefan Savage, and Thomas Anderson. 2000. Modeling TCP latency. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'00)*, Vol. 3. IEEE, 1742–1751. https://doi.org/10.1109/INFCOM.2000.832574

[20] Dah-Ming Chiu and Raj Jain. 1989. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Comput. Netw. ISDN Syst.* 17, 1 (Jun. 1989), 1–14. https://doi.org/10.1016/0169-7552(89)90019-6

[21] M. Corradi, R. G. Garroppo, S. Giordano, and M. Pagano. 2001. Analysis of f-ARIMA processes in the modelling of broadband traffic. In *Proceedings of the IEEE International Conference on Communications. Conference Record (ICC'01)*, Vol. 3. IEEE, 964–968. https://doi.org/10.1109/ICC.2001.937380

[22] Igor D. D. Curcio, Vinod Kumar Malamal Vadakital, and Miska M. Hannuksela. 2010. Geo-predictive Real-time media delivery in mobile environment. In *Proceedings of the 3rd Workshop on Mobile Video Delivery*. ACM, 3–8. https://doi.org/10.1145/1878022.1878036

[23] Emiliano Dall'Anese, Seung-Jun Kim, and Georgios B. Giannakis. 2011. Channel gain map tracking via distributed kriging. *IEEE Trans. Vehic. Technol.* 60, 3 (Feb. 2011), 1205–1211. https://doi.org/10.1109/TVT.2011.2113195

[24] Girisha De Silva, Mun Choon Chan, and Wei Tsang Ooi. 2016. Throughput estimation for short lived TCP cubic flows. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS'16)*. ACM, New York, NY, 227–236. https://doi.org/10.1145/2994374.2994391

[25] David A. Dickey and Wayne A. Fuller. 1979. Distribution of the estimators for autoregressive time series with a unit root. *J. Am. Statist. Assoc.* 74, 366 (Jun. 1979), 427–431. https://doi.org/10.1080/01621459.1979.10482531

[26] Lu Dong, Yi Qiao, Peter Dinda Dinda, and Fabián E. Bustamante. 2005. Characterizing and predicting TCP throughput on the wide area network. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*. 414–424. https://doi.org/10.1109/ICDCS.2005.17

[27] I. El Khayat, P. Geurts, and G. Leduc. 2007. Machine-learnt versus analytical models of TCP throughput. *Comput. Netw.* 51, 10 (2007), 2631–2644. https://doi.org/10.1016/j.comnet.2006.11.017

[28] Alberto Garcia Estevez and Niklas Carlsson. 2014. Geo-location-aware emulations for performance evaluation of mobile applications. In *Proceedings of the 11th Annual Conference on Wireless On-demand Network Systems and Services (WONS'14)*. IEEE, 73–76. https://doi.org/10.1109/WONS.2014.6814724

[29] Kristian Evensen, Andreas Petlund, Haakon Riiser, Paul Vigmostad, Dominik Kaspar, Carsten Griwodz, and Pål Halvorsen. 2011. Mobile video streaming using location-based network prediction and transparent handover. In *Proceedings of the 21st International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'11)*. ACM, 21–26. https://doi.org/10.1145/1989240.1989248

[30] Florian Eyben, Felix Weninger, Erik Marchi, and Björn W. Schuller. 2013. Likability of human voices: A feature analysis and a neural network regression approach to automatic likability estimation. In *Proceedings of the 14th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'13)*. Paris, France, 1–4. https://doi.org/10.1109/WIAMIS.2013.6616159

[31] Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Mach. Learn.* 63, 1 (Apr. 2006), 3–42. https://doi.org/10.1007/s10994-006-6226-1

[32] Amir Ghasemi. 2018. Predictive Modeling of LTE User Throughput Via Crowd-Sourced Mobile Spectrum Data. In *Proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN'18)*. IEEE, 1–5. https://doi.org/10.1109/DySPAN.2018.8610464

[33] Mukul Goyal, Roch Guerin, and Raju Rajan. 2002. Predicting TCP throughput from non-invasive network sampling. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, Vol. 1. IEEE, 180–189. https://doi.org/10.1109/INFCOM.2002.1019259

[34] N. K. Groschwitz and G. C. Polyzos. 1994. A time series model of long-term NSFNET backbone traffic. In *Proceedings of the International Conference on Communications*, Vol. 3. 1400–1404. https://doi.org/10.1109/ICC.1994.368876

[35] Aurélien Géron. 2017. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media, Inc. 543 pages.

[36] Jia Hao, Roger Zimmermann, and Haiyang Ma. 2014. GTube: Geo-predictive video streaming over HTTP in mobile environments. In *Proceedings of the 5th ACM Multimedia Systems Conference (MMSys'14)*. ACM, 259–270. https://doi.org/10.1145/2557642.2557647

[37] Qi He, Constantine Dovrolis, and Mostafa Ammar. 2005. On the predictability of large transfer TCP throughput. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'05)*. ACM, New York, NY, 145–156. https://doi.org/10.1145/1080091.1080110

[38] Qi He, Constantinos Dovrolis, and Mostafa Ammar. 2005. Prediction of TCP throughput: Formula-based and history-based methods. In *ACM SIGMETRICS Performance Evaluation Review* (2005-06-06). ACM, 388–389. https://doi.org/10.1145/1064212.1064268

[39] Gerard B. M. Heuvelink, Edzer Pebesma, and Benedikt Gräler. 2015. Space-time geostatistics. In *Encyclopedia of GIS*. Springer International Publishing, 1–7. https://doi.org/10.1007/978-3-319-23519-6_1647-1

[40] Kim Hojgaard-Hansen, Tatiana K Madsen, and Hans-Peter Schwefel. 2012. Reducing communication overhead by scheduling tcp transfers on mobile devices using wireless network performance maps. In *Proceedings of the 18th European Wireless Conference*. VDE, IEEE, 1–8.

[41] Liang Hu, Xilong Che, and Xiaochun Cheng. 2010. Bandwidth prediction based on nu-support vector regression and parallel hybrid PArticle swarm optimization. *Int. J. Comput. Intell. Syst.* 3, 1 (Apr. 2010), 70–83. https://doi.org/10.1080/18756891.2010.9727678

[42] Tsung-i Huang and Jaspal Subhlok. 2005. Fast pattern-based throughput prediction for TCP bulk transfers. In *Proceedings of the IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05)*, Vol. 1. IEEE, 410–417. https://doi.org/10.1109/CCGRID.2005.1558584

[43] Jae-Hyun Hwang and Chuck Yoo. 2010. Formula-based TCP throughput prediction with available bandwidth. *IEEE Commun. Lett.* 14, 4 (Apr. 2010), 363–365. https://doi.org/10.1109/LCOMM.2010.04.092309

[44] A. K. Jain, Jianchang Mao, and K. M. Mohiuddin. 1996. Artificial neural networks: A tutorial. *Computer* 29, 3 (1996), 31–44. https://doi.org/10.1109/2.485891

[45] Manish Jain and Constantinos Dovrolis. 2003. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. *IEEE/ACM Trans. Netw.* 11, 4 (Aug. 2003), 537–549. https://doi.org/10.1109/TNET.2003.815304

[46] Florian Jomrich, Florian Fischer, Steffan Knapp, Tobias Meuser, Björn Richerzhagen, and Ralf Steinmetz. 2018. Enhanced cellular bandwidth prediction for highly automated driving. In *Proceedings of the 7th International Conference on Smart Cities, Green Technologies and Intelligent Transport Systems (SMARTGREENS'18), and Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS'18)*. Communications in Computer and Information Science, Vol. 992. Springer International Publishing. https://doi.org/10.1007/978-3-030-26633-2

[47] Florian Jomrich, Alexander Herzberger, Tobias Meuser, Björn Richerzhagen, Ralf Steinmetz, and Cornelius Wille. 2018. Cellular bandwidth prediction for highly automated driving-evaluation of machine learning approaches based on real-world data. In *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS'18)*, Vol. 1. INSTICC, SciTePress, 121–132. https://doi.org/10.5220/0006692501210132

[48] Florian Jomrich, Josef Schmid, Steffen Knapp, Alfred Hos, Ralf Steinmetz, and Bjorn Schuller. 2018. Analysing communication requirements for crowd sourced backend generation of HD Maps used in automated driving. In *Proceedings of the IEEE Vehicular Networking Conference (VNC'18)*. IEEE, 1–8. https://doi.org/10.1109/VNC.2018.8628335

[49] Theodoros Kamakaris and Jeffrey V' Nickerson. 2005. Connectivity maps: Measurements and applications. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. 307–307. https://doi.org/10.1109/HICSS.2005.162

[50] Roger P. Karrer. 2007. TCP prediction for adaptive applications. In *Proceedings of the 32nd IEEE Conference on Local Computer Networks (LCN'07)*. IEEE, 989–996. https://doi.org/10.1109/LCN.2007.145

[51] Lutz Kelch, Tobias Pogel, Lars Wolf, and Andreas Sasse. 2013. CQI maps for optimized data distribution. In *Proceedings of the IEEE 78th Vehicular Technology Conference (VTC Fall'13)*. IEEE, 1–5. https://doi.org/10.1109/VTCFall.2013.6692148

[52] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. 1998. Rate control for communication networks: Shadow prices, proportional fairness and stability. *J. Operat. Res. Soc.* 49, 3 (Mar. 1998), 237–252. https://doi.org/10.1057/palgrave.jors.2600523

[53] Gil Keren, Nicholas Cummins, and Björn Schuller. 2018. Calibrated prediction intervals for neural network regressors. *IEEE Access* 6 (Sep. 2018), 1–1. https://doi.org/10.1109/ACCESS.2018.2871713

[54] Chunghan Lee, Hirotake Abe, Toshio Hirotsu, and Kyoji Umemura. 2012. Analytical modeling of network throughput prediction on the internet. *IEICE Trans. Inf. Syst.* E95-D, 12 (2012), 2870–2878. https://doi.org/10.1587/transinf.E95.D.2870

[55] Q. Liu and N. Rao. 2018. On concavity and utilization analytics of wide-area network transport protocols. In *2018 Proceedings of the IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS'18)*. 430–438. https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00088

[56] Yan Liu and Jack Y. B. Lee. 2015. An empirical study of throughput prediction in mobile data networks. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'15)*. IEEE, 1–6. https://doi.org/10.1109/GLOCOM.2015.7417858

[57] Steven H. Low, Larry L. Peterson, and Limin Wang. 2002. Understanding TCP Vegas: A duality model. https://doi.org/10.1145/506147.506152

[58] Dong Lu, Yi Qiao, P. A. Dinda, and F. E. Bustamante. 2005. Modeling and taming parallel tcp on the wide area network. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium.* IEEE, Denver, CO, USA, 10–20. https://doi.org/10.1109/IPDPS.2005.291

[59] Zachary MacHardy, Ashiq Khan, Kazuaki Obana, and Shigeru Iwashina. 2018. V2x access technologies: Regulation, research, and remaining challenges. *IEEE Commun. Surv. Tutor.* 20, 3 (Feb. 2018), 1858–1877. https://doi.org/10.1109/COMST.2018.2808444

[60] Rishabh Madan and Partha Sarathi Mangipudi. 2018. Predicting computer network traffic: A time series forecasting approach using DWT, ARIMA and RNN. In *Proceedings of the 11th International Conference on Contemporary Computing (IC3'18).* IEEE, 1–5. https://doi.org/10.1109/IC3.2018.8530608

[61] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural adaptive video streaming with pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM'17).* Association for Computing Machinery, 197–210. https://doi.org/10.1145/3098822.3098843

[62] Miquel Martínez, David de Andrés, Juan-Carlos Ruiz, Mahbub Hassan, and Salil Kanhere. 2012. Towards changing the user perception of mobile communications through geotagged information. In *Proceedings of the 1st European Workshop on AppRoaches to MObiquiTous Resilience (ARMOR'12).* ACM, 5. https://doi.org/10.1145/2222436.2222441

[63] Saverio Mascolo, Claudio Casetti, Mario Gerla, M. Y. Sanadidi, and Ren Wang. 2001. TCP westwood: Bandwidth estimation for enhanced transport over wireless links. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom'01).* Association for Computing Machinery, 287–297. https://doi.org/10.1145/381677.381704

[64] Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, and Teunis Ott. 1997. The Macroscopic behavior of the TCP congestion avoidance algorithm. *SIGCOMM Comput. Commun. Rev.* 27, 3 (Jul. 1997), 67–82. https://doi.org/10.1145/263932.264023

[65] Konstantin Miller, Abdel-Karim Al-Tamimi, and Adam Wolisz. 2016. QoE-Based low-delay live streaming using throughput predictions. *ACM Trans. Multimedia Comput. Commun. Appl.* 13, 1, Article 4 (Oct. 2016), 24 pages. https://doi.org/10.1145/2990505

[66] Mariyam Mirza, Joel Sommers, Paul Barford, and Xiaojin Zhu. 2010. A machine learning approach to TCP throughput prediction. *IEEE/ACM Trans. Netw.* 18, 4 (Aug. 2010), 1026–1039. https://doi.org/10.1109/TNET.2009.2037812

[67] Mariyam Mirza, Kevin Springborn, Suman Banerjee, Paul Barford, Michael Blodgett, and Xiaojin Zhu. 2009. On the accuracy of TCP throughput prediction for opportunistic wireless networks. In *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks.* IEEE, 1–9. https://doi.org/10.1109/SAHCN.2009.5168952

[68] Douglas C. Montgomery, Cheryl L. Jennings, and Murat Kulahci. 2015. *Introduction to Time Series Analysis and Forecasting* (2nd ed.). John Wiley & Sons, Hoboken, New Jersey.

[69] Ghulam Murtaza, Andreas Reinhardt, Mahbub Hassan, and Salil S. Kanhere. 2014. Creating personal bandwidth maps using opportunistic throughput measurements. In *2014 Proceedings of the IEEE International Conference on Communications (ICC'14).* IEEE, 2454–2459. https://doi.org/10.1109/ICC.2014.6883691

[70] Akshay Narayan, Frank Cangialosi, Deepti Raghavan, Prateesh Goyal, Srinivas Narayana, Radhika Mittal, Mohammad Alizadeh, and Hari Balakrishnan. 2018. Restructuring endpoint congestion control. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM'18).* Association for Computing Machinery, 30–43. https://doi.org/10.1145/3230543.3230553

[71] Netradar. 2021. Retrieved from https://www.netradar.com/.

[72] Thuy T. T. Nguyen and Grenville Armitage. 2008. A survey of techniques for internet traffic classification using machine learning. *IEEE Commun. Surv. Tutor.* 10, 4 (2008), 56–76. https://doi.org/10.1109/SURV.2008.080406

[73] Anthony J. Nicholson and Brian D. Noble. 2008. Breadcrumbs: Forecasting mobile connectivity. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking.* ACM, 46–57. https://doi.org/10.1145/1409944.1409952

[74] Stefanie Judith Opitz, Nicole Todtenberg, and Hartmut König. 2014. Mobile bandwidth prediction in the context of emergency medical service. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA'14).* ACM, 49. https://doi.org/10.1145/2674396.2674411

[75] Teunis J. Ott, J. H. B. Kemperman, and Matt Mathis. 1996. *The Stationary Behavior of Ideal TCP Congestion Avoidance.*

[76] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. 1998. Modeling TCP throughput: A simple model and its empirical validation. *SIGCOMM Comput. Commun. Rev.* 28, 4 (Oct. 1998), 303–314. https://doi.org/10.1145/285243.285291

[77] Edwin Pednault. 2006. Transform regression and the kolmogorov superposition theorem. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, Joydeep Ghosh, Diane Lambert, David Skillicorn, and Jaideep

Srivastava (Eds.). SIAM, Society for Industrial and Applied Mathematics, Bethesda, MD, 35–46. https://doi.org/10.1137/1.9781611972764.4

[78] Johannes Pillmann, Benjamin Sliwa, Christian Kastin, and Christian Wietfeld. 2017. Empirical evaluation of predictive channel-aware transmission for resource efficient car-to-cloud communication. In *Proceedings of the IEEE Vehicular Networking Conference (VNC'17)*. IEEE, 235–238. https://doi.org/10.1109/VNC.2017.8275635

[79] Tobias Pögel, Jan Lübbe, and Lars Wolf. 2012. Passive client-based bandwidth and latency measurements in cellular networks. In *Proceedings of the IEEE INFOCOM Workshops*. IEEE, 37–42. https://doi.org/10.1109/INFCOMW.2012.6193516

[80] Tobias Pogel and Lars Wolf. 2015. Optimization of vehicular applications and communication properties with Connectivity Maps. In *Proceedings of the IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops'15)*. IEEE, 870–877. https://doi.org/10.1109/lcnw.2015.7365940

[81] Tobias Pögel and Lars Wolf. 2012. Analysis of operational 3G network characteristics for adaptive vehicular Connectivity Maps. In *Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW'12)*. IEEE, 355–359. https://doi.org/10.1109/WCNCW.2012.6215521

[82] Tobias Pögel and Lars Wolf. 2012. Prediction of 3G network characteristics for adaptive vehicular Connectivity Maps (Poster). In *Proceedings of the IEEE Vehicular Networking Conference (VNC'12)*. IEEE, 121–128. https://doi.org/10.1109/VNC.2012.6407420

[83] Yi Qiao, J. Skicewicz, and P. Dinda. 2004. An empirical study of the multiscale predictability of network traffic. In *Proceedings of the 13th IEEE International Symposium on High performance Distributed Computing*. IEEE, 66–76. https://doi.org/10.1109/HPDC.2004.1323493

[84] Mohammed Quddus, Washington Yotto Ochieng, Lin Zhao, and Robert Noland. 2003. A General map matching algorithm for transport telematics applications. *GPS Solut.* 7, 3 (Dec. 2003), 157–167. https://doi.org/10.1007/s10291-003-0069-z

[85] Darijo Raca, Ahmed H. Zahran, Cormac J. Sreenan, Rakesh K. Sinha, Emir Halepovic, Rittwik Jana, and Vijay Gopalakrishnan. 2017. Back to the Future: Throughput prediction for cellular networks using radio KPIs. In *Proceedings of the 4th ACM Workshop on Hot Topics in Wireless (HotWireless'17)*. ACM, 37–41. https://doi.org/10.1145/3127882.3127892

[86] Nageswara S.V. Rao, Qiang Liu, Satyabrata Sen, Don Towlsey, Gayane Vardoyan, Raj Kettimuthu, and Ian Foster. 2017. TCP throughput profiles using measurements over dedicated connections. In *Proceedings of the 26th International Symposium on High-Performance Parallel and Distributed Computing (HPDC'17)*. Association for Computing Machinery, New York, NY, 193–204. https://doi.org/10.1145/3078597.3078615

[87] Nageswara S. V. Rao, Satyabrata Sen, Zhengchun Liu, Rajkumar Kettimuthu, and Ian Foster. 2019. Learning concave-convex profiles of data transport over dedicated Connections. In *Machine Learning for Networking*, Lecture Notes in Computer Science. Springer International Publishing, Cham, 1–22. https://doi.org/10.1007/978-3-030-19945-6_1

[88] Haakon Riiser, Tore Endestad, Paul Vigmostad, Carsten Griwodz, and Pål Halvorsen. 2012. Video Streaming Using a Location-based Bandwidth-lookup Service for Bitrate Planning. *ACM Trans. Multimedia Comput. Commun. Appl.* 8, 3, Article 24 (Aug. 2012), 19 pages. https://doi.org/10.1145/2240136.2240137

[89] Nayera Sadek and Alireza Khotanzad. 2004. Multi-scale high-speed network traffic prediction using k-factor Gegenbauer ARMA model. In *Proceedings of the IEEE International Conference on Communications*, Vol. 4. IEEE, 2148–2152. https://doi.org/10.1109/ICC.2004.1312898

[90] Alassane Samba, Yann Busnel, Alberto Blanc, Philippe Dooze, and Gwendal Simon. 2016. Throughput prediction in cellular networks: Experiments and preliminary results. In *1ères Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication (CoRes'16) (1ères Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication)*. Bayonne, France.

[91] Alassane Samba, Yann Busnel, Alberto Blanc, Philippe Dooze, and Gwendal Simon. 2017. Instantaneous throughput prediction in cellular networks: Which information is needed? In *Proceedings of the IFIP/IEEE Symposium on Integrated Network and Service Management (IM'17)*. IEEE, 624–627. https://doi.org/10.23919/INM.2017.7987345

[92] SamKnows. 2021. Retrieved from https://www.samknows.com.

[93] Aimin Sang and San-qi Li. 2002. A predictability analysis of network traffic. *Comput. Netw.* 39, 4 (Jul. 2002), 329–345. https://doi.org/10.1016/S1389-1286(01)00304-8

[94] Josef Schmid, Philipp Heß, Alfred Höß, and Björn W Schuller. 2018. Passive monitoring and geo-based prediction of mobile network vehicle-to- server communication. In *Proceedings of the 14th International Wireless Communications & Mobile Computing Conference (IWCMC'18)*, Vol. 14. IEEE, 1483–1488. https://doi.org/10.1109/IWCMC.2018.8450395

[95] Josef Schmid, Mathias Schneider, Alfred Höß, and Björn W Schuller. 2019. A comparison of AI-Based throughput prediction for cellular vehicle-to-server communication. In *Proceedings of the IWCMC 2019 Vehicular Symposium (IWCMC-VehicularCom'19)*.

[96] Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. 2000. New support vector algorithms. *Neural Comput.* 12, 5 (2000), 1207–1245. https://doi.org/10.1162/089976600300015565

[97] Yantai Shu, Zhigang Jin, Lianfang Zhang, Lei Wang, and O. W. W. Yang. 1999. Traffic prediction using FARIMA models. In *Proceedings of the IEEE International Conference on Communications (ICC'99)*, Vol. 2. IEEE, 891–895. https://doi.org/10.1109/ICC.1999.765402

[98] Varun Singh, Jorg Ott, and Igor D. D. Curcio. 2012. Predictive buffering for streaming video in 3G networks. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'12)*. IEEE, 1–10. https://doi.org/10.1109/WoWMoM.2012.6263710

[99] Benjamin Sliwa, Robert Falkenberg, Thomas Liebig, Johannes Pillmann, and Christian Wietfeld. 2018. Machine learning based context-predictive car-to-cloud communication using multi-layer connectivity maps for upcoming 5G networks. In *Proceedings of the IEEE 88th Vehicular Technology Conference (VTC-Fall'18)*. IEEE, 1–7. https://doi.org/10.1109/VTCFall.2018.8690856

[100] Alex J. Smola and Bernhard Schölkopf. 2004. A tutorial on support vector regression. *Stat. Comput.* 14, 3 (2004), 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

[101] Sebastian Sonntag, Lennart Schulte, and Jukka Manner. 2013. Mobile network measurements—It's not all about signal strength. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'13)*. IEEE, 4624–4629. https://doi.org/10.1109/WCNC.2013.6555324

[102] Evangelos D. Spyrou and Dimitris Mitrakos. 2018. Hidden Markov Model Traffic Characterisation in Wireless Networks. Retrieved from http://www.scitepress.org/PublicationsDetail.aspx?ID=OW8EQCImPnk=&t=1.

[103] Rayadurgam Srikant. 2012. *The Mathematics of Internet Congestion Control*. Springer Science & Business Media.

[104] Yi Sun, Xiaoqi Yin, Junchen Jiang, Vyas Sekar, Fuyuan Lin, Nanshu Wang, Tao Liu, and Bruno Sinopoli. 2016. CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction. In *Proceedings of the 2016 ACM SIGCOMM Conference (SIGCOMM'16)*. ACM, 272–285. https://doi.org/10.1145/2934872.2934898

[105] Bayan Taani and Roger Zimmermann. 2016. Spatio-temporal analysis of bandwidth maps for geo-predictive video streaming in mobile environments. In *Proceedings of the 2016 ACM on Multimedia Conference (MM'16)*. ACM, Amsterdam, The Netherlands, 888–897. https://doi.org/10.1145/2964284.2964333

[106] Pedro Torres, Paulo Marques, Hugo Marques, Rogério Dionísio, Tiago Alves, Luis Pereira, and Jorge Ribeiro. 2017. Data analytics for forecasting cell congestion on LTE networks. In *Proceedings of the Network Traffic Measurement and Analysis Conference (TMA'17)*. 1–6. https://doi.org/10.23919/TMA.2017.8002917

[107] Sudharshan Vazhkudai and Jennifer M. Schopf. 2002. Predicting sporadic grid data transfers. In *Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing*. IEEE, 188–196. https://doi.org/10.1109/HPDC.2002.1029918

[108] Ermias Andargie Walelgne, Jukka Manner, Vaibhav Bajpai, and Jorg Ott. 2018. Analyzing throughput and stability in cellular networks. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS'18)*. IEEE, 1–9. https://doi.org/10.1109/NOMS.2018.8406261

[109] Bo Wei, Kenji Kanai, and Jiro Katto. 2016. History-based throughput prediction with Hidden Markov Model in mobile networks. In *Proceedings of the IEEE International Conference on Multimedia & Expo Workshops (ICMEW'16)*. IEEE, 1–6. https://doi.org/10.1109/ICMEW.2016.7574683

[110] Bo Wei, Kenji Kanai, Wataru Kawakami, and Jiro Katto. 2018. HOAH: A hybrid TCP throughput prediction with autoregressive model and hidden markov model for mobile networks. *IEICE Trans. Commun.* E101.B, 7 (2018), 1612–1624. https://doi.org/10.1587/transcom.2017CQP0007

[111] Bo Wei, Wataru Kawakami, Kenji Kanai, and Jiro Katto. 2017. A history-based tcp throughput prediction incorporating communication quality features by support vector regression for mobile network. In *Proceedings of the IEEE International Symposium on Multimedia (ISM'17)*. IEEE, 374–375. https://doi.org/10.1109/ISM.2017.74

[112] Bo Wei, Wataru Kawakami, Kenji Kanai, Jiro Katto, and Shangguang Wang. 2018. TRUST: A TCP throughput prediction method in mobile networks. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'18)*. IEEE, 1–6. https://doi.org/10.1109/GLOCOM.2018.8647390

[113] Felix J. Weninger, Florian Eyben, and Björn W. Schuller. 2014. On-line continuous-time music mood regression with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'14)*. 5412–5416. https://doi.org/10.1109/ICASSP.2014.6854637

[114] Keith Winstein and Hari Balakrishnan. 2013. TCP ex machina: computer-generated congestion control. *ACM SIGCOMM Computer Communication Review* 43, 4 (Sept. 2013), 123–134. https://doi.org/10.1145/2534169.2486020

[115] Damon Wischik, Costin Raiciu, Adam Greenhalgh, and Mark Handley. 2011. Design, implementation and evaluation of congestion control for multipath TCP. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation (NSDI'11)*. USENIX Association, USA, 99–112.

[116] Suchao Xiao and Wen Chen. 2018. Dynamic allocation of 5G transport network slice bandwidth based on LSTM traffic prediction. In *Proceedings of the IEEE 9th International Conference on Software Engineering and Service Science (ICSESS'18)*. IEEE, 735–739. https://doi.org/10.1109/ICSESS.2018.8663796

[117] Ke Xu, Dan Wang, Chunyi Peng, Kai Zheng, Rashid Mijumbi, Qingyang Xiao, et al. 2017. A longitudinal measurement study of TCP performance and behavior in 3G/4G networks over high speed rails. *IEEE/ACM Trans. Netw.* 25, 4 (Aug. 2017), 2195–2208. https://doi.org/10.1109/TNET.2017.2689824

[118] Qiang Xu, Sanjeev Mehrotra, Zhuoqing Mao, and Jin Li. 2013. PROTEUS: Network performance forecast for real-time, interactive mobile applications. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys'13)*. ACM, 347–360. https://doi.org/10.1145/2462456.2464453

[119] Y. R. Yang and S. S. Lam. 2000. General AIMD congestion control. In *Proceedings of the 2000 International Conference on Network Protocols*. 187–198. https://doi.org/10.1109/ICNP.2000.896303

[120] Jun Yao. 2011. *A Framework for Improving QoS in Mobile Computing*. PhD thesis. School of Computer Science and Engineering, The University of New South Wales, Sydney, Australia.

[121] Jun Yao, Salil S. Kanhere, and Mahbub Hassan. 2008. An empirical study of bandwidth predictability in mobile computing. In *Proceedings of the 3rd ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH'08)*. ACM, New York, NY, 11–18. https://doi.org/10.1145/1410077.1410081

[122] Jun Yao, Salil S. Kanhere, and Mahbub Hassan. 2010. Using bandwidth-road maps for improving vehicular internet access. In *Proceedings of the 2nd International Conference on COMmunication Systems and NETworks (COMSNETS'10)*. IEEE, 460–461. https://doi.org/10.1109/COMSNETS.2010.5431969

[123] Jun Yao, Salil S. Kanhere, and Mahbub Hassan. 2012. Improving QoS in high-speed mobility using bandwidth maps. *IEEE Trans. Mobile Comput.* 11, 4 (2012), 603–617. https://doi.org/10.1109/TMC.2011.97

[124] Jun Yao, Salil S. Kanhere, Imran Hossain, and Mahbub Hassan. 2011. Empirical evaluation of HTTP adaptive streaming under vehicular mobility. In *Proceedings of the International IFIP TC 6 Networking Conference (NETWORKING'11)*, Jordi Domingo-Pascual, Pietro Manzoni, Sergio Palazzo, Ana Pont, and Caterina Scoglio (Eds.). Springer, Berlin, 92–105. https://doi.org/10.1007/978-3-642-20757-0_8

[125] Esma Yildirim, Dengpan Yin, and Tevfik Kosar. 2011. Prediction of optimal parallelism level in wide area data transfers. *IEEE Trans. Parallel Distrib. Syst.* 22, 12 (2011), 2033–2045. https://doi.org/10.1109/TPDS.2011.228

[126] Jun Yoa, Salil S. Kanhere, and Mahbub Hassan. 2009. Geo-intelligent traffic scheduling for multi-homed on-board networks. In *Proceedings of the 4th ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch'09)*. ACM.

[127] Hiroshi Yoshida, Kozo Satoda, and Tutomu Murase. 2013. Constructing stochastic model of TCP throughput on basis of stationarity analysis. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'13)*. IEEE, 1544–1550. https://doi.org/10.1109/GLOCOM.2013.6831293

[128] Kun Yu, Congxiao Bao, and Xing Li. 2013. Pathperf: Path bandwidth estimation utilizing websites. In *Proceedings of the International Conference on Passive and Active Network Measurement*. Springer, Berlin, 270–272. https://doi.org/10.1007/978-3-642-36516-4_31

[129] Chaoqun Yue, Ruofan Jin, Kyoungwon Suh, Yanyuan Qin, Bing Wang, and Wei Wei. 2018. LinkForecast: Cellular link bandwidth prediction in LTE networks. *IEEE Trans. on Mobile Comput.* 17, 7 (2018), 1582–1594. https://doi.org/10.1109/TMC.2017.2756937

[130] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. 2019. Deep learning in mobile and wireless networking: A survey. *IEEE Commun. Surv. Tutor.* 21, 3 (2019), 2224–2287. https://doi.org/10.1109/COMST.2019.2904897

[131] Wenjing Zhang, Yuan Liu, Tingting Liu, and Chenyang Yang. 2018. Trajectory prediction with recurrent neural networks for predictive resource allocation. In *Proceedings of the 14th IEEE International Conference on Signal Processing (ICSP'18)*. IEEE, 634–639. https://doi.org/10.1109/ICSP.2018.8652460

[132] Mohamed Faten Zhani, Halima Elbiaze, and Farouk Kamoun. 2007. α-SNFAQM: An active queue management mechanism using neurofuzzy prediction. In *Proceedings of the 12th IEEE Symposium on Computers and Communications*. IEEE, 381–386. https://doi.org/10.1109/ISCC.2007.4381596

[133] Mohamed Faten Zhani, Halima Elbiaze, and Farouk Kamoun. 2009. Analysis and prediction of real network traffic. *J. Netw.* 4, 9 (Nov. 2009). https://doi.org/10.4304/jnw.4.9.855-865

[134] Garson Zhong and Ayub Bokani. 2014. A geo-adaptive javascript DASH player. In *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming (VideoNext'14)*. ACM, 39–40. https://doi.org/10.1145/2676652.2683463

[135] Bo Zhou, Dan He, Zhili Sun, and Wee Hock Ng. 2006. Network traffic modeling and prediction with ARIMA/GARCH. In *Proceedings of the International Workshop on Heterogeneous Networking Environments and Technologies Conference (HET-NETs'06)*. 1–10.

[136] Jia Zhou, Fengyuan Ren, and Chuang Lin. 2009. Saturation aware TCP throughput prediction. In *Proceedings of the IEEE 28th International Performance Computing and Communications Conference*. IEEE, 71–78. https://doi.org/10.1109/PCCC.2009.5403823

[137] Xuan Kelvin Zou, Jeffrey Erman, Vijay Gopalakrishnan, Emir Halepovic, Rittwik Jana, Xin Jin, Jennifer Rexford, and Rakesh K. Sinha. 2015. *Can Accurate Predictions Improve Video Streaming in Cellular Networks?* ACM Press, 57–62. https://doi.org/10/gcpx57 00000.