

Prediphant: Short Term Heavy User Prediction

Davide Sanvito
NEC Laboratories Europe
Heidelberg, Germany

Giuseppe Siracusano
NEC Laboratories Europe
Heidelberg, Germany

Roberto Gonzalez
NEC Laboratories Europe
Madrid, Spain

Roberto Bifulco
NEC Laboratories Europe
Heidelberg, Germany

Abstract—Traffic prediction is of paramount importance for the correct management of network infrastructures. Most research efforts try to forecast the aggregated traffic over the network and over large time windows. In this work, we tackle the problem the other way around. That is, we predict the behaviour of individual users over short time windows. First, we investigate the contribution of the most data eager users to the global network traffic. We do it by analyzing network traces coming from several thousand real users. Then, we design a technique, based on a combination of Natural Language Processing (NLP) and Long Short-Term Memory (LSTM) machine learning models, that leverages past navigation patterns to predict sudden changes in the amount of resources consumed by each user. Finally, we evaluate our method using real data finding it is able to predict about 80% of the users that will rump up their network needs in most realistic scenarios.

I. INTRODUCTION

The recent deployment of 5G, increasing significantly the speed in the access network, coupled with the pervasive usage of video and other traffic intensive applications has attracted the attention of the research community [1] and the industry [2] over the prediction of traffic consumption. A correct prediction of the amount of traffic requested is key for an efficient usage of the network resources and opens the possibility of new business models such as the network multitancy [3] or the network virtualization [4].

Current efforts are focused on the prediction of the aggregated traffic consumed by all the users of the network with different time granularity or to discover the capacity required in the network in order to avoid Service Level Agreement (SLA) infringements [5].

Both the aforementioned solutions are thought for the implementation of network slicing policies [6] to split the network resources. This split can be performed either by operator (i.e., offering different guarantees to different Mobile network operators (MNOs) or Mobile Virtual Network Operators (MVNOs) or by service (i.e., offering different conditions to different OTT services such as Netflix or Facebook). Thus, they are designed to allow the application of statistical multiplexing solutions, optimizing that way the usage of the resources and reducing the current over provision of the network.

However, all solutions so far predict the average traffic of the network with a granularity that ranges from 10 minutes [7]–[10] up to 1 hour [11]–[14]. This is because the intrinsic stochasticity of the network traffic makes impossible to predict the traffic in the very short term. Then, these solutions

may provide a good solution for the average traffic of the network, but will not capture bursty behaviours.

Furthermore, they make their predictions without taking into consideration the traffic consumed by each individual user. Even when most users in the network are *Mice* that consume a small amount of traffic, a few data hungry *Elephants* users absorb most of the network resources [15]–[17].

In this paper we present *prediphant*, a Machine Learning (ML) based system that is able to detect if a user is going to become an *Elephant* in the short term, allowing in that way the network operator to proactively adapt the resources before they are required. The system is built, trained and evaluated using traffic traces from a mobile network serving more than 10k real users during the period of one month.

To this end, we first analyze one month of mobile traffic traces to understand the characteristics of the user traffic. We perform an analysis at flow level, aggregated traffic and user level. Two main insights emerge from this analysis: the traffic in the mobile network presents a huge short term variability, with peaks of traffic that can go up to 20% over the average in a 15 min. window; and, the traffic consumption of a few users represents a big percentage of the total traffic consumed, even when the number of simultaneous users grow.

Then, we present *prediphant*. We explain the intuition behind the functioning of the system, the details about the training procedure and how it can be used online in a real scenario.

Finally, we present the evaluation and discussion. We observe that *prediphant* is able to predict up to 90% of the times a user will become an *Elephant*. This prediction allows a fine-grained prediction of the traffic to be served by the network.

II. FRAMING THE PROBLEM

In this section we analyze the specific characteristics of the traffic carried by mobile networks. In particular, we focus on understanding the different metrics that define a network flow and how they aggregate all together to define the load of the network.

A. Mobile Traffic Dataset

In this paper, we use a dataset including traces from all the mobile traffic of several thousand real users over one month in an Asian country. The information is logged at a mobile network proxy used mainly to do traffic shaping and optimization. The proxy is placed at the core of the network, however, it is the first element to be traversed by all the traffic

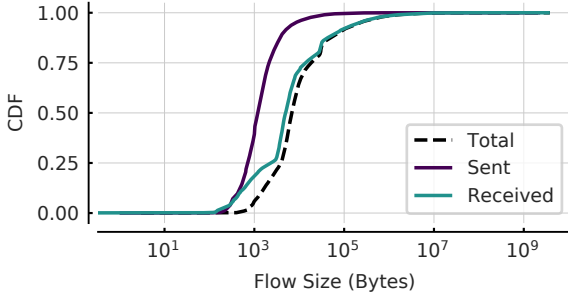


Fig. 1. CDF of the size of the flows in Bytes

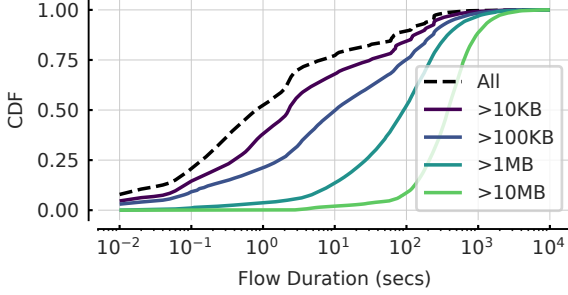


Fig. 2. CDF of the duration of the flows for flows of different sizes.

served by the mobile network. Thus, it has full visibility over all the traffic in that network, including both the traffic coming from the MNO and the different MVNOs. Moreover, the network serves a mix of 3G, 4G and 5G traffic.

For each TCP flow observed in the network we have different statistics like the (anonymized) final user establishing the connection, the time when the connection was started and ended and the total amount of bytes transmitted in both directions. Information of other protocols such as UDP is also logged.

B. Analysis per flow

This section characterises the duration and weight of the flows. Fig. 1 shows the distribution of the flow size in bytes. The median size of a flow is 6.8kB (including upload and download), only 5% of the flows have more than 220kB and 1% more than 1.5MB.

As expected, the data received (that is, the data flowing from the Internet to the mobile devices) is typically higher than the data sent by the mobile devices. This happens for 78% of the flows. The median size of the uploaded and downloaded data in each flow is 1194 bytes and about 5kB, respectively.

Then, we focus on the duration of the flows. Fig. 2 presents the Cumulative Distribution Function (CDF) for the duration of the flows, in seconds, for flows of different size. First, we observe the dashed black line, representing all the flows observed. 50% of the flows are active more than 0.8secs and only 15% last more than one minute.

Moreover, non surprisingly, the duration of the flows grows with the amount of information carried by them. It indicates

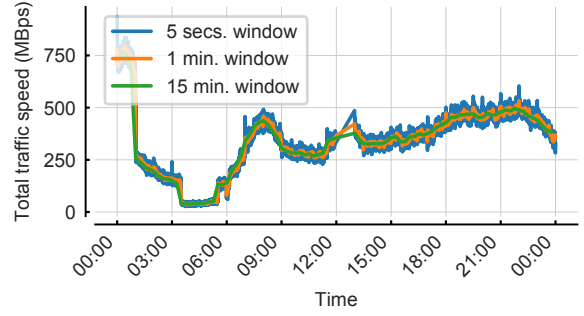


Fig. 3. Average traffic speed in each time windows for different window sizes.

that the number of long lasting idle connections is very small in the network under study. The median duration of the flows carrying more than 10kB, 100kB, 1MB and 10MB is 2.27, 10, 92 and 380 seconds, respectively.

C. Aggregated Network Traffic

Following, we aggregate all the traffic to characterize the load supported by the network in each moment. To this end, we split our dataset in windows of 5 seconds, 1 minute and 15 minutes. We assign to each window all the traffic that starts and ends in the same window. For the flows that span over more than one windows, we split the traffic evenly among all the windows.

Fig. 3 shows the average network speed (in MBps) for the aforementioned window lengths. As expected, the longer windows (15 min., green line) presents the general trend on the traffic consumption over the day. It grows from 6:00 until 8:00 (Commuting time) to decrease and stay stable during the morning and afternoon, then it steadily grows over the evening to decrease again over night. Just at 00:00, and during about 1 hour, we observe a peak on the amount of traffic consumed. We speculate there is a service periodically running at that time that causes this effect.

Most traffic prediction works [7]–[14] use a time window for their prediction in the order of the 15 minutes. This is because, as explained above, using that granularity the traffic follows stable trends. For smaller windows sizes (5 secs. and 1 min.), the traffic also follows the same general trend. However, if we focus on the details, we can observe a very noisy trend that could harm the ability of any resource allocation service to predict the traffic needs. For example, at around 20:00 hours, when the 15min. windows observes an average traffic of less than 500MBps, in a 5secs. window we observe a peak of almost 600MBps. Thus, traditional traffic prediction mechanisms would not be able to adapt to this sudden changes in traffic.

D. Users Contribution

Finally, we study the contribution of each user to the traffic in each traffic window. We use the shortest windows considered in the previous section, 5 seconds. This allows us to explore the variability on the users contribution with a very high granularity.

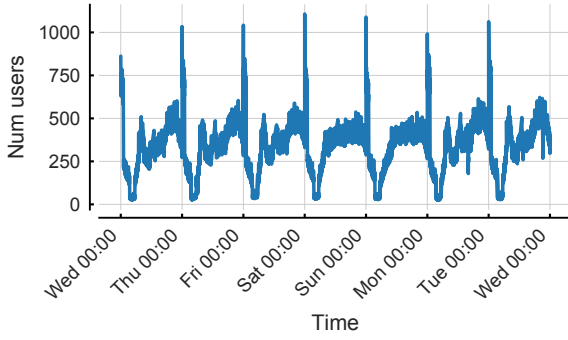


Fig. 4. Number of users active in each 5 seconds window.

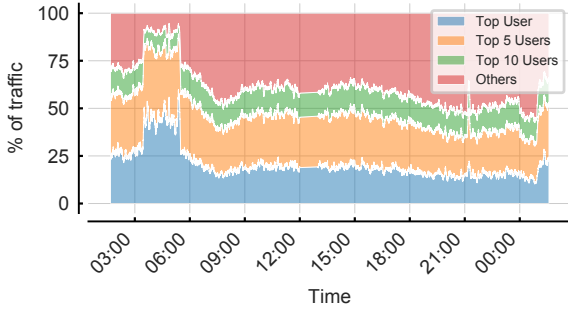


Fig. 5. Percentage of the traffic consumed by the top users in each moment.

First, we analyze the number of users active in each one of the 5 second windows. Fig. 4 shows the number of users active during one week starting at midnight of a Wednesday. We observe a clear day/night pattern with between 200 and 600 active users simultaneously during the day, a higher activity in the afternoon than in the morning and peaks of more than 1000 users around midnight. As it was happening with the network speed, we speculate that peaks are produced by a pre-installed service pooling every day at the same time in more than 1000 devices. Moreover, the number of users quickly drop after midnight, having some windows with less than 50 active users between the 3:00 and the 4:30 hours. Finally, during the weekend, we don't observe the differences between morning and night observed in the weekdays.

Then, we try to understand the contribution of each user to the network traffic. Fig. 5 shows the percentage of traffic in each window that is used by the top users. The figure shows the contribution of the top, top 5 and top 10 users in each time windows during one day. Surprisingly, the contribution of the top user in each window is over 10% during the whole day with peaks of almost 50% of the traffic consumed by a single user in the hours with less users in the system.

Moreover, the top 5/10 users, consume about 50/60% of the traffic, respectively. This percentage slightly decreases in the evening together with the grow in number of concurrent users observed in Fig. 4.

All together, the results indicate that a very small number of users is responsible for a very high percentage of the

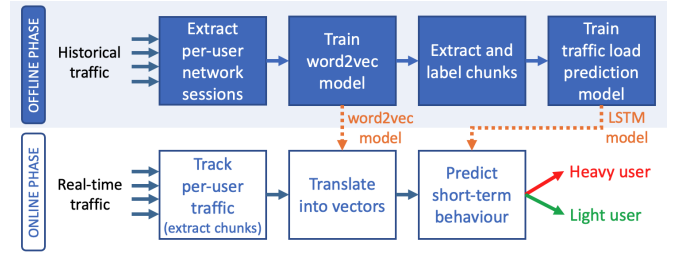


Fig. 6. prediphant high-level design

traffic served in a specific moment. Thus, being able to identify users consuming a high amount of traffic, namely, the *Elephants*, may help better shaping the traffic needs in the short term in better ways than existing methods that try to predict the aggregated traffic.

III. PREDIPHANT

A. Concept

Navigation patterns (i.e., the sequence of hosts contacted by a single user), have been recently used for user profiling [18] or for the detection of malicious activities [19], [20]. Hosts can be either network domain names or IP addresses. In this work we investigate the use of navigation patterns as indicator of the amount of traffic load that a user will generate in the short term. The intuition behind this is that a user, before accessing a content that will generate a high traffic load (e.g., watching a video or downloading a file/application), will spend some time to look and to choose such content. That is, the per user short-term network traffic behavior directly depends on which hosts the user has visited before, i.e. the context of such visits.

Starting from this intuition we developed *prediphant*, a system capable of detecting short-term heavy hitter behaviour (elephant users) given the history of visited network hosts. As common in systems based on Machine Learning, *prediphant* operates in two phases: an *offline phase* and an *online phase* (cf. Fig. 6). During the offline phase it processes historical traffic network traces to associate sequences of visited network hosts (end hosts) to the amount of traffic generated by such visits, and then it builds a ML model that maps each sequence to high or low traffic loads. In the online phase, the trained model is used to detect whether a user is going to become an heavy user in the short term.

B. Offline phase

The offline phase follows three different steps: i) Host-to-vector: unsupervised learning is used to build a vector representation of the end hosts (embeddings); ii) Hosts-to-Load: sequences of end hosts are associated with high or low traffic volume; iii) Traffic load prediction: a supervised LSTM model is trained to recognize high/low traffic sequences.

Hosts-to-vector. In the first step, hosts are transformed into a vector representation that captures relationships among hosts and describes how they occur together. To do so we borrow a technique from Natural Language Processing

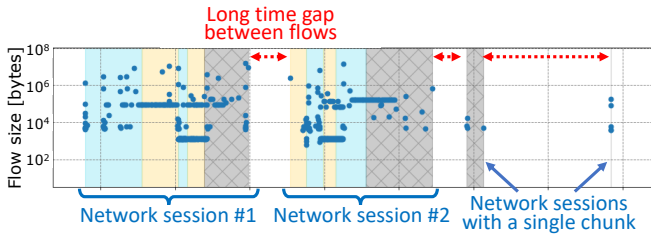


Fig. 7. Example of traffic traces of a single user split in network sessions. Blue and yellow bands represent chunks used to generate a prediction. The last chunk of a session (in gray) instead cannot be used since the next chunk belongs to a different session.

(NLP). Indeed, we use word2vec [21] a well known NLP algorithm that is used to learn a mathematical representation of the meanings associated with single words, where words with the same meaning have a similar representation. That is, by replacing words with end hosts, and sentences with network sessions, we are able to apply NLP techniques to represent domains as vectors.

A network session is a sequence of correlated network flows generated by the same user, i.e., flows that belong to the same browsing session or user activity. Hosts appearing in the same network session are correlated, while hosts part of different sessions (even from the same user) do not have any correlation. Thus, the word2vec algorithm is configured to capture relationships that occur only in the same network session. We divide traffic into network sessions by checking time distance across flows (Fig. 7). The time gap value can be extracted from a statistical analysis on the dataset: for the experiment in this paper it is set to the 99-th percentile of inter-arrival time across flows, i.e. 120 seconds.

Hosts-to-Load. Network sessions alone cannot be directly used to predict the amount of traffic load that a user will generate in the future. Indeed, by definition, there is no direct correlation between two contiguous network sessions. Instead there is a high correlation between hosts visited in a network session and the short term traffic load that the user will generate in the same session. Therefore, we divide each network session in smaller subsequences, called *chunks*, that will be then used as input to the traffic prediction model.

In the second step of the offline phase, network sessions are split into chunks and associated with the corresponding generated traffic load. Sessions can be split in chunks with two different approaches: based on size or based on time. When the size-based approach is used, the sequence of visited hosts is divided into chunks of the same size (e.g.: 10, 50, 100 contacted end hosts). Instead, when the temporal-based approach is used, each chunk will contain the end hosts visited in a given interval of time. In the rest of this paper we will focus only on the first approach.

The selection of the chunk size parameter is crucial for the configuration of the system. In fact, *prediphant* predicts the traffic load that will be generated in the next chunk, hence the size of the chunk has a direct impact on the time granularity of the prediction. Table I shows the prediction granularity in our dataset when considering different values

TABLE I
DURATION OF THE CHUNKS, IN SECONDS,
FOR DIFFERENT VALUES OF CHUNK SIZE

Chunk Size	10	50	100
50-th percentile	2.0	12.1	27.3
75-th percentile	6.5	56.4	108.8
95-th percentile	51.2	215.9	364.9
99-th percentile	151.3	459.4	714.0
Average	10.8	51.9	91.2

of the chunk size. On average, with smaller chunks (10 hosts) *prediphant* predicts the load of the next 10 seconds, while bigger chunks (100 hosts) size can provide predictions about the next 90 seconds.

After the network sessions have been split in chunks, each chunk is assigned a high/low class based on the traffic load. Classes are selected by comparing against a threshold the sum of all the traffic transmitted and received in the flows part of the chunk.

Traffic load prediction. In the third step, a supervised traffic load prediction model is trained. We use a Long Short-Term Memory (LSTM) [22], a type of Recurrent Neural Network frequently employed in NLP thanks to its ability in learning temporal dependencies in the input sequences. The LSTM network is fed with the sequence of (vectorised) end hosts part of a chunk and predicts the class of the *next* chunk as a classification task. That is, given a chunk it predicts the traffic load that will be generated by the next chunk of the same network session as high load or low load class. Note that not all the chunks can be used to train the LSTM model. In fact, both the last chunk of each network session and chunks part of network sessions with a single chunk cannot be used for the training (cf. Fig. 7).

C. Online phase

During the online phase, *prediphant* uses the trained LSTM model to analyze live traffic and predict per-user short-term traffic load. *prediphant* keeps track of the end hosts visited in the current chunk by each user in the network, and feeds this information to the prediction model. The output of the model is a list of users that will have a high traffic load in the next chunk.

The operations performed during this phase are the following. *prediphant* processes network packets and extracts (*user*, *end host*, *timestamp*) triplets. Both destination IP address or domain can be used to represent an end host, depending on their availability. For each user in the network, it records the number of visited end host, the sequence of visited end hosts and the time elapsed since the start of the chunk. Once the chunk is complete (i.e., the sequence of visited end hosts reaches the chunk size), hosts in the chunk are translated to their vector representation using the learned word2vec model, and then the chunk is fed to the LSTM prediction model which finally classifies the user as future heavy/low hitter (cf. Fig. 6).

It is worth to notice that the aforementioned operations have to be performed in a specific time frame, i.e., right before the start of the next chunk, otherwise the generated prediction would not be available in useful time. We computed the chunk interarrival time to estimate the available time budget. In our dataset the average interarrival time is about 900 milliseconds.

IV. EVALUATION

We evaluate `prediphant` using the dataset presented in Section II-A with three different configurations for the chunk size: 10, 50 and 100 hosts. Depending on the chunk size the dataset is composed by 6.8M, 1.3M and 650k chunks, respectively. The threshold used to label chunks as high or low traffic is set to 1 MB. Furthermore, since just 1% of flows are larger than 1.5MB (cf. Fig. 1), the datasets are naturally unbalanced towards low load chunks: the high load chunks represents the 5%, 25% and 45% of the dataset when considering 10, 50 and 100 hosts per chunk, respectively. The model is trained using the first two weeks of traffic traces and then tested on the remaining two weeks.

As evaluation metrics we computed Precision and Recall. Given the class imbalance in the data, in order to provide a comprehensive evaluation of the performance [23], we computed the two metrics for both classes¹. We refer to the positive (negative) class as the chunks associated to a subsequent high (low) traffic load, respectively. The Recall for the positive (negative) class tells how many high (low) traffic load chunks out of the total number of high (low) load chunks are correctly predicted by the model, i.e. it reports the True Positive (Negative) Rate, respectively. The Precision indicates instead the probability that a chunk predicted as positive (negative) is actually associated to a subsequent high (low) traffic load.

Results are presented in Fig. 8. `prediphant` achieves over 80% Recall for the prediction of heavy users and about 80% Recall for the prediction of light users. For all the three different chunk size configurations the model correctly predicts if a user will be an heavy/light hitter in the next chunk in around 80% of the cases. When considering chunks of size 10, the Precision of heavy user prediction is around 20%. Under this configuration, the model would introduce a considerable amount of false positives. On the other hand, this configuration has the best Precision for light user prediction, close to 99%. The other two configurations (50 and 100 hosts per chunk) present instead a more balanced performance across the two classes.

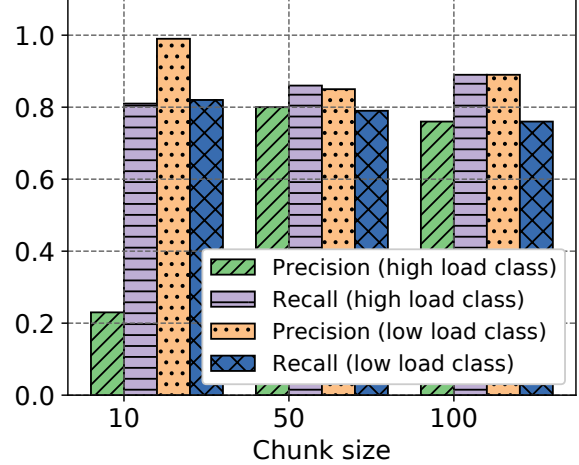


Fig. 8. Performance of `prediphant` for different values of Chunks Size

V. DISCUSSION AND FUTURE WORK

Aggregated traffic forecasting over long periods of time is popular in the research community and allows the predictive management of resources in mobile networks. It is even more important nowadays with the emergence of virtualized network solutions and infrastructures shared among multiple tenants. However, this aggregated forecasting does not solve some of the problems faced when it is applied to realistic scenarios. First, it does not capture the high short-term variability of the traffic demands. As shown with the analysis of real world traces, the short-term variability on traffic demands can vary more than 20% over the average traffic demand in a window of 15 minutes.

Moreover, our analysis confirms the intuitions that only a few final users consume most of the network resources. This fact, that was well known in fixed networks, stays similar for mobile users with the top 5 users consuming half of the available resources. Thus, knowing the users that will become *elephants* in the short time can help in the prediction of the global traffic needs, and, therefore the correct provisioning of resources.

In this paper we have presented `prediphant`. A method that monitors the hosts visited by users to anticipate the consumption of huge amount of data (i.e., a user that start downloading a file or watching streaming video). Using standard ML methods such as word2vec and LSTM it is able to predict when a user will start/stop being an *elephant* with an accuracy over 80% in most scenarios.

We speculate standard traffic forecasting methods may be combined with `prediphant` to predict the traffic consumption in a fine grain manner. While the standard traffic forecasting can easily guess the global trend of the traffic consumption, the prognosis of data hunger users could be used to improve the global results on the short term. We leave this as future work.

¹The Recall for the negative class is the Selectivity, also called True Negative Rate (TNR); The Precision for the negative class is instead the Negative Predictive Value (NPV)

ACKNOWLEDGEMENT

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101017171 ("MARSAL") and No. 883335 ("PALANTIR"). This paper reflects only the authors' views and the European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

- [1] J. Barros, M. Araujo, and R. J. Rossetti, "Short-term real-time traffic prediction methods: A survey," in *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE, 2015, pp. 132–139.
- [2] "TMS (traffic management solution)," <https://www.nec.com/en/global/solutions/nsp/tms/index.html>, accessed: 2022-03-01.
- [3] J. A. Ayala-Romero, A. Garcia-Saavedra, M. Gramaglia, X. Costa-Perez, A. Banchs, and J. J. Alcaraz, "vrain: A deep learning approach tailoring computing and radio resources in virtualized rans," in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.
- [4] "Rakuten Mobile, NEC and Intel demonstrate industry-leading performance in containerized 5g core lab trial," https://corp.mobile.rakuten.co.jp/english/news/press/2021/0625_01/, accessed: 2022-03-01.
- [5] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Aztec: Anticipatory capacity allocation for zero-touch network slicing," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 794–803.
- [6] N. Alliance, "Description of network slicing concept," *NGMN 5G P*, vol. 1, no. 1, pp. 1–11, 2016.
- [7] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2018, pp. 231–240.
- [8] F. Xu, Y. Lin, J. Huang, D. Wu, H. Shi, J. Song, and Y. Li, "Big data driven mobile traffic understanding and forecasting: A time series approach," *IEEE transactions on services computing*, vol. 9, no. 5, pp. 796–805, 2016.
- [9] L. Fang, X. Cheng, H. Wang, and L. Yang, "Mobile demand forecasting via deep graph-sequence spatiotemporal modeling in cellular networks," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3091–3101, 2018.
- [10] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, and P. Bertin, "Improving traffic forecasting for 5g core network scalability: A machine learning approach," *IEEE Network*, vol. 32, no. 6, pp. 42–49, 2018.
- [11] X. Wang, Z. Zhou, F. Xiao, K. Xing, Z. Yang, Y. Liu, and C. Peng, "Spatio-temporal analysis and prediction of cellular traffic in metropolis," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 2190–2202, 2018.
- [12] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.
- [13] L. Chen, D. Yang, D. Zhang, C. Wang, J. Li *et al.*, "Deep mobile traffic forecast and complementary base station clustering for c-ran optimization," *Journal of Network and Computer Applications*, vol. 121, pp. 59–69, 2018.
- [14] J. Feng, X. Chen, R. Gao, M. Zeng, and Y. Li, "Deeptp: An end-to-end neural network for mobile cellular traffic prediction," *IEEE Network*, vol. 32, no. 6, pp. 108–115, 2018.
- [15] L. Guo and I. Matta, "The war between mice and elephants," in *Proceedings Ninth International Conference on Network Protocols. ICNP 2001*. IEEE, 2001, pp. 180–188.
- [16] K. Papagiannaki, N. Taft, S. Bhattacharyya, P. Thiran, K. Salamatian, and C. Diot, "A pragmatic definition of elephants in internet backbone traffic," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, 2002, pp. 175–176.
- [17] P. Megyesi and S. Molnár, "Analysis of elephant users in broadband network traffic," in *Meeting of the European Network of Universities and Companies in Information and Communication Engineering*. Springer, 2013, pp. 37–45.
- [18] R. Gonzalez, C. Soriente, J. M. Carrascosa, A. Garcia-Duran, C. Iordanou, and M. Niepert, "User profiling by network observers," in *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, 2021, pp. 212–222.
- [19] G. Siracusano, M. Trevisan, R. Gonzalez, and R. Bifulco, "Poster: On the application of nlp to discover relationships between malicious network entities," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2019, pp. 2641–2643.
- [20] M. Sharif, J. Urakawa, N. Christin, A. Kubota, and A. Yamada, "Predicting impending exposure to malicious content from user behavior," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 1487–1501.
- [21] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *ICLR*, 2013.
- [22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [23] H. Van Le and H. Zhang, "Log-based anomaly detection with deep learning: How far are we?" in *2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE)*. IEEE, 2022, to appear.