King's Research Portal

Document Version
Peer reviewed version

Link to publication record in King's Research Portal

Online Supervised Learning for Traffic Load Prediction in Framed-ALOHA Networks

Nan Jiang, Yansha Deng, Osvaldo Simeone, and Arumugam Nallanathan

Abstract

Predicting the current backlog, or traffic load, in framed-ALOHA networks enables the optimization of resource allocation, e.g., of the frame size. However, this prediction is made difficult by the lack of information about the cardinality of collisions and by possibly complex packet generation statistics. Assuming no prior information about the traffic model, apart from a bound on its temporal memory, this paper develops an online learning-based adaptive traffic load prediction method that is based on Recurrent Neural Networks (RNN) and specifically on the Long Short-Term Memory (LSTM) architecture. In order to enable online training in the absence of feedback on the exact cardinality of collisions, the proposed strategy leverages a novel approximate labeling technique that is inspired by Method of Moments (MOM) estimators. Numerical results show that the proposed online predictor considerably outperforms conventional methods and is able to adapt to changing traffic statistics.

Index Terms

Traffic load prediction, framed-ALOHA, online supervised learning, recurrent neural network.

I. Introduction

Framed-ALOHA (f-ALOHA) has been widely adopted as a key component of multiple access protocols in many state-of-the-art wireless communication systems, including Long-Term Evolution (LTE) and 5G New Radio (NR). In f-ALOHA, time is organized into time frames, with each frame containing multiple Random Access Opportunities (RAOs). RAOs refer to subsets of channel resources in time, frequency, or/and code domain, e.g., random access preambles in the LTE system. In each frame, devices select RAOs at random and transmit to the connected Base Station (BS) in an uncoordinated manner. Collisions cause
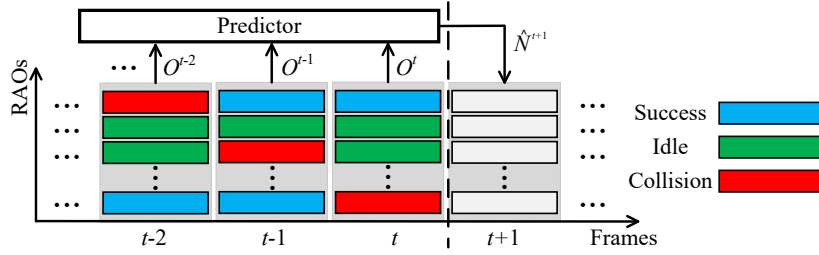
Fig. 1: Timeline of an f-ALOHA protocol and target predictor based on historical data about collided, successful, and idle RAOs in previous frames.

devices to retransmit in following frames, increasing the backlog of packets to be transmitted in a frame beyond the load due to newly generated packets. Adapting the number of RAOs per frame to the estimated current backlog, or traffic load, is an important step to relieve network congestion and reduce access delays. However, prediction is made difficult by the fact that the BS does not have access to the cardinality of the collisions, that is, to the number of devices that have selected the same RAO, and by the possibly complex nature of the incoming traffic statistics (see Fig. 1). As an example, the incoming traffic may consist of mixtures of different traffic types, including periodic, event-driven (bursty), multimedia streaming patterns, and etc. [1, 2].

Estimating traffic backlog in f-ALOHA can only rely on the observation on the number of RAOs in each frame that are idle, collided, or successful (see Fig. 1). Previous classical works have proposed Method of Moment (MOM)-based estimators that aim at matching the average number of such RAOs to the current measurements [3]. More recent works proposed to predict bursty traffic for event-driven applications, i.e., for massive devices being activated by an external event to request transmissions within a short period, using drift analysis [4], MOM [5], or Maximum-Likelihood (ML) estimation [6]. All these prior works estimate the current backlog only based on the latest observations of idle, collided, or successful RAOs, while ignoring historical data from prior frames. As argued in this work, information about RAOs in previous frames can be useful to learn features of the traffic statistics that enable an improved prediction.

In this work, we target modern Internet of Things (IoT) traffic scenarios with complex statistics, possibly encompassing mixtures of long- and short-memory processes, e.g., a mixture of random and periodic transmissions with long duty cycles [1, 2]. In order to capture the complex dynamics of the IoT traffic, we propose an online supervised learning method that adopts a Recurrent Neural Network (RNN) model based on the state-of-the-art Long Short-Term Memory (LSTM) architecture [7]. The most relevant prior work is [8], where the authors leverage LSTM to classify mobile encrypted traffic. To the best of our knowledge, the application of LSTM to traffic backlog prediction in f-ALOHA has not been considered

before. The proposed method is able to adapt at runtime by leveraging a novel supervised technique based on approximate labeling. In particular, given the absence of feedback on the exact cardinality of collisions, approximate target labels defining the current backlog are estimated in a manner inspired by the discussed memoryless MOM solutions [3, 5].

The rest of the paper is organized as follows. Section II describes the system model and problem formulation. Section III provides background material. Section IV introduces the proposed online supervised learning method. Section V presents numerical results, and Section VI concludes the paper.

II. System Model and Problem Formulation

We consider a f-ALOHA network system consisting of an arbitrary number of devices and a single BS. As illustrated in Fig. 1, time is divided into frames, where each frame contains F RAOs. At the given frame t, a number of devices N^t are *active*, that is, they either have tried unsuccessfully to access previous frames or they have newly generated packets to transmit. The number N^t represents the current *backlog* of the system, also referred to as *traffic load*. We assume an arbitrary process for the generation of new packets, which is unknown to the BS. Every device has only two possible states, either inactive or active, while it has a single packet to be transmitted in the latter case. This captures sporadic traffic random access in cellular-based networks as detailed in [2, 4–6].

Each active device chooses one of the F RAOs uniformly at random for transmission. As a result of the random selections of the RAOs, a transmission may fail if a collision occurs, i.e., two or more devices select the same RAO at the same frame. For each RAO, the BS can detect whether it contains a single transmission, a collision, or whether it is idle, i.e., it contains no transmitted packet. To take into account the effects of radio channels, e.g., path-loss, fading, and interference, we further assume that an RAO that contains a collision or a single transmission can be erroneously detected as idle with probability p_{ed} [2]. If a packet is successful detected without collision, the corresponding transmitting device receives an acknowledgement message from the BS. If a packet is either collided or undetected, it is re-transmitted on a randomly selected RAO in the next frame unless a maximum number γ_{max} of attempts is exceeded. These re-tranmissions result in backlog accumulation in future frames, which can in turn increase the collision probability. The processing latency of sequence transmission and acknowledgement are ignored as in prior works [2–6].

In this letter, the objective is to predict the number N^{t+1} of devices that are active at the beginning of each frame $t + 1$. To do so, the BS keeps a record of the observations $O^{t'}$ made in prior and current frames $t' = 1, 2, \cdots, t$. Each observation $O^{t'} = \{V_{\text{i}}^{t'}, V_{\text{s}}^{t'}, V_{\text{c}}^{t'}\}$ includes the number $V_{\text{i}}^{t'}$ of idle RAOs, the number $V_{\text{s}}^{t'}$ of successful detected RAOs, and the number $V_{\text{c}}^{t'}$ of collided RAOs, as detected by the BS (possibly in an erroneous manner, as discussed). Using the observed history $H^t = \{O^1, O^2, \cdots, O^t\}$ in frame t, the

goal of this paper is to predict the forthcoming traffic value N^{t+1} by learning a conditional distribution $\mathbb{P}\{N^{t+1} = n | H^t\}$ and then solving the Maximum A Posteriori (MAP) problem

$$(\text{P1}) : \hat{N}^{t+1} = \underset{n \in \{0, 1, \ldots, N_{\max}\}}{\arg \max} \quad \mathbb{P}\{N^{t+1} = n | H^t\}, \tag{1}$$

where the $N_{\max}$ is an upper bound on the backlog. Unless one makes simplistic assumptions as in [4–6], problem (1) remains generally intractable even in the presence of realistic traffic models, further justifying the data-driven solutions developed in prior art and in this letter. We emphasize that we focus solely on the problem of prediction and that we leave the problem of investigating the interplay between overload control via, e.g., frame size selection and access barring, and traffic prediction to future work.

III. CONVENTIONAL BACKLOG PREDICTION

In this section, we review two conventional traffic prediction methods, namely MOM-based algorithms [3] (see also [9, 10, Sec. III] [5, Sec. V-A]), and the current state-of-the-art ML estimator of [6]. Both these methods use an estimate $\tilde{N}^t$ for the backlog N^t in the current frame, given observations in frame t, as the prediction $\hat{N}^{t+1}$ for the backlog in frame $t + 1$. We emphasize that, throughout this letter, given information H^t available at the end of frame t, we use the notation $\tilde{N}^t$ to represent an estimate of the backlog in frame t, while $\hat{N}^{t+1}$ denotes a prediction for the backlog at the beginning of frame $t + 1$.

A. MOM-Based Estimator

Given a backlog $N^t = n$ in any frame t, neglecting the possibility of detection errors, the expected number of RAOs in idle, success, and collisions state are given respectively as

$$E_{\mathrm{i}}(n) = \mathbb{E}[V_{\mathrm{i}} | N^t = n] = F\left(1 - \frac{1}{F}\right)^n, \tag{2}$$

$$E_{\mathrm{s}}(n) = \mathbb{E}[V_{\mathrm{s}} | N^t = n] = n\left(1 - \frac{1}{F}\right)^{n-1}, \tag{3}$$

$$E_{\mathrm{c}}(n) = \mathbb{E}[V_{\mathrm{c}} | N^t = n] = F\left(1 - \left(1 - \frac{1}{F}\right)^n - \frac{n}{F}\left(1 - \frac{1}{F}\right)^{n-1}\right), \tag{4}$$

where we recall that F is the number of RAOs. These expectations can be easily computed by noting that each active of the n devices selects any of the F RAOs with equal probability.

MOM estimators $\tilde{N}^t_{\mathrm{MOM}}$ of the current backlog N^t aim at matching one or more of the moments in (2), (3), and (4) to the current observations V^t_{i}, V^t_{s}, and V^t_{c}, respectively. A MOM estimator hence generally finds a value of n that minimizes a measure of the discrepancy between the moments in (2), (3), and (4),

on the one hand, and the respective observations, on the other. For instance, one could consider the Mean Absolute Error (MAE)

$$\varphi^t(n) = \frac{1}{3}(|E_i(n) - V_i^t| + |E_s(n) - V_s^t| + |E_c(n) - V_c^t|),\tag{5}$$

which would yield the MOM-based estimator

$$\hat{N}_{\text{MOM}}^{t+1} = \tilde{N}_{\text{MOM}}^t = \underset{n \in \{0,1,...,N_{\max}\}}{\arg\min} \varphi^t(n).\tag{6}$$

Simplified, and generally less accurate, MOM-based estimators that enjoy closed-form solutions have been proposed in [5, 9]. As an example, in [9, Sec. 3], traffic load N^t is estimated by matching the single moment (2) with the given observation V_i^t. Imposing the equality $E_i(n) = V_i^t$ and using the rounded solution n as the estimate $\tilde{N}^{t+1}$ of the backlog for the current frame yields the estimator

$$\hat{N}_{\text{MOM}}^{t+1} = \tilde{N}_{\text{MOM}}^t = \text{round}\left\{\log_{(1-\frac{1}{F})}\left(\frac{V_i^t}{F}\right)\right\},\tag{7}$$

where round$\{\cdot\}$ is the nearest integer function.

B. ML Estimator

A more complex estimator can be obtained by using the ML estimator $\tilde{N}_{\text{ML}}^t$ of the current backlog [6]. This is given as

$$\hat{N}_{\text{ML}}^{t+1} = \tilde{N}_{\text{ML}}^t = \underset{n \in \{0,1,...,N_{\max}\}}{\arg\max} \mathbb{P}\{O^t | N^t = n\}.\tag{8}$$

Solving problem (8) requires the computation of the probability $\mathbb{P}\{O^t | N^t = n\}$ for each possible $n \in \{0, 1, ..., N_{\max}\}$ given the current observation O^t. Note that each value $\mathbb{P}\{O^t | N^t = n\}$ represents the likelihood of a value n given the current observation. Reference [6] proposes a numerical approach that computes the vector of probabilities, or likelihoods, $\mathbb{P}\{O^t | N^t = n\}$ for all $n \in \{0, 1, ..., N_{\max}\}$ as the steady-state probability vector of a Markov chain. This Markov chain is defined by letting each device sequentially, and independently, select an RAO at each step. We refer to [6] for details on the numerical procedure.

IV. ONLINE SUPERVISED LEARNING-BASED BACKLOG PREDICTION

In this section, we propose an online supervised learning approach for the training of a predictor of the forthcoming traffic load N^{t+1} given the observations H^t available at the end of frame t. Unlike the existing methods presented above, the proposed scheme aims at capturing not only the information present in the most recent observation O^t, but also the historical information in the previous observations in H^t in order to detect patterns in the traffic generation mechanism and in the communication protocol. To this end, the
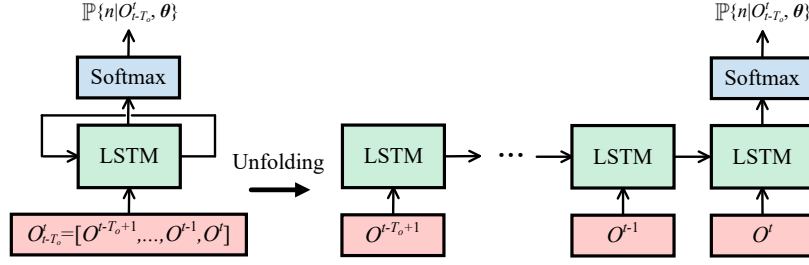
Fig. 2: Proposed LSTM architecture (left) with temporal unfolding (right). The LSTM block can contain multiple stacked LSTM layers that are fully connected.

method leverages an RNN, and specifically an LSTM architecture, which is capable of recognizing patterns in temporal data.

As illustrated in Fig. 2, the RNN is sequentially fed the observation O^t and is tasked with the goal of reproducing $\hat{N}^{t+1}$ at any time t. Since the latter value is never observed at the BS, the proposed method substitutes it with an estimator $\tilde{N}^{t+1}$ obtained in the next frame $t+1$ using one of the methods described in Sec. III. In this way, with a delay of one frame, the weights of the RNN are adapted in order to minimize the error of the predictor $\hat{N}^{t+1}$ output by the RNN at frame t with respect to the estimate $\tilde{N}^{t+1}$ computed in the following frame. Note that this estimate uses information not available in frame t and hence it allows the RNN to observe and detect patterns in the history H^t that correlate with future traffic loads. Details are provided next.

The proposed LSTM architecture is illustrated in Fig. 2. An LSTM layer, consisting of multiple standard LSTM units [7] receives at each frame t the current observation vector O^t and is connected to an output layer consisting of a softmax non-linearity with $N_{\max}+1$ output values. The softmax layer outputs an estimate $\mathbb{P}\{\hat{N}^{t+1} = n|O^t_{t-T_o}, \boldsymbol{\theta}\}$ of the probability of each possible value n for the forthcoming backlog N^{t+1} given a window T_o of previous observations $O^t_{t-T_o} = [O^{t-T_o+1}, \cdots, O^{t-1}, O^t]$. The probability $\mathbb{P}\{\hat{N}^{t+1} = n|O^t_{t-T_o}, \boldsymbol{\theta}\}$ is parameterized by the vector θ that contains both the LSTM internal parameters and the weights of the softmax layer.

More precisely, we adopt a *stateless* implementation of LSTM whereby, as shown in the right-hand side of Fig. 2, at each frame t the LSTM network's network state is re-initialized and the network is progressively fed the observations $O^{t-T_o+1}, \cdots, O^{t-1}$, and O^t, producing the vector of probabilities $\mathbb{P}\{\hat{N}^{t+1} = n|O^t_{t-T_o}, \boldsymbol{\theta}\}$ upon observing vector $O^t_{t-T_o}$. Parameter T_o should generally be selected so as to capture the expected memory for the packet generation process and communication protocol. Examples will be given in the next section. We note that stateless implementations are generally known to be easier to train and more stable than their

stateful counterparts, in which the state of the LSTM layer is not re-initialized at each step [11].

In order to adapt the model parameter θ, we adopt standard stochastic gradient descent implemented via BackPropagation Through Time (BPTT) [12]. In particular, at each frame $t+1$, the BS estimates the current backlog $\tilde{N}^{t+1}$ using one of the methods discussed in Sec. III. Then, it updates the weights θ in the direction of the negative gradient of the cross-entropy loss

$$L^t(\boldsymbol{\theta}) = -\log\left(\mathbb{P}\{\hat{N}^{t+1} = \tilde{N}^{t+1} | O_{t-T_o}^{t+1}, \boldsymbol{\theta}\}\right), \tag{9}$$

where we recall that the probability $\mathbb{P}\{\hat{N}^{t+1} = \tilde{N}^{t+1} | O_{t-T_o}^{t+1}, \boldsymbol{\theta}\}$ is defined by the LSTM and by the softmax layer. The gradient can be computed via BPTT using standard methods. In practice, rather than applying the gradient of Eq. (9) at frame $t + 1$, it is preferable to consider a window, or random mini-batch, of T_b previous values and evaluate the gradient of the average loss

$$L^t(\boldsymbol{\theta}) = -\sum_{t'=t-T_b+1}^{t} \log\left(\mathbb{P}\{\hat{N}^{t'+1} = \tilde{N}^{t'+1} | O_{t'-T_o}^{t'+1}, \boldsymbol{\theta}\}\right). \tag{10}$$

This can generally reduce the variance of the stochastic gradient and improve stability of training [13].

In order to reduce the time and computational resource needed for convergence of LSTM training, it is useful to initialize the weights of the LSTM by first running offline experiments based on available traffic models, which can be mismatched to online traffic statistics. This may be considered as an example of meta-learning [14]. We will provide an example in the next section.

V. NUMERICAL RESULTS

In this section, numerical experiments are conducted to evaluate the traffic load prediction accuracy of the proposed online supervised learning method. We mostly assume the presence of $N_u = 1000$ devices generating a packet in any frame independently with probability 0.005, as well as of additional $N_p = 20$ devices generating one packet every $T_p = 10$ frames in a deterministic (periodic) manner. This scenario captures the coexistence of services with both random and periodic traffic types [1, 2]. We will also consider random bursty traffic [2, Ch. 6.1], as detailed latter. Unless stated otherwise, other parameters are set according to 3GPP technical report for Machine-Type Communication [2] as follows: error detection probability $p_{ed} = 0.05$; retransmission constraint $\gamma_{max} = 10$; and number of RAOs $F = 54$.

We compare the performance in terms of prediction error among the MOM predictor (7), the ML predictor (8), and three LSTM-based predictors. The first, referred to as "Offline LSTM", trains the LSTM during an offline phase with 10^5 frames of synthetically generated traffic. For the offline phase, the exact backlog N^{t+1} can be used when training using the cross-entropy criterion. The statistics of the offline traffic are different from the online traffic in that the former only contains the underlying random traffic, while the latter

TABLE I: Supervised Learning Hyperparameters

Hyperparameters	Value
Memory T_o	20
RMSProp learning rate α	0.0001
LSTM drop-out rate	0.2
Minibatch size	64
Historical samples size T_b	1000

also models periodic traffic. This offline scheme is treated as the baseline, and its weights are transferred to the online LSTM-based predictors as initialization. The second scheme, referred to as "Online LSTM", implements the proposed online scheme by using the MOM estimator (5) in the cross-entropy criterion (10), while adapting to the online traffic statistics. The third, referred to as "Genie-aided LSTM", trains the LSTM by using criterion Eq. (10) with the correct value N^{t+1}. The performance of this scheme provides an upper bound, due to its use of an ideal supervision in the form of signal N^{t+1}.
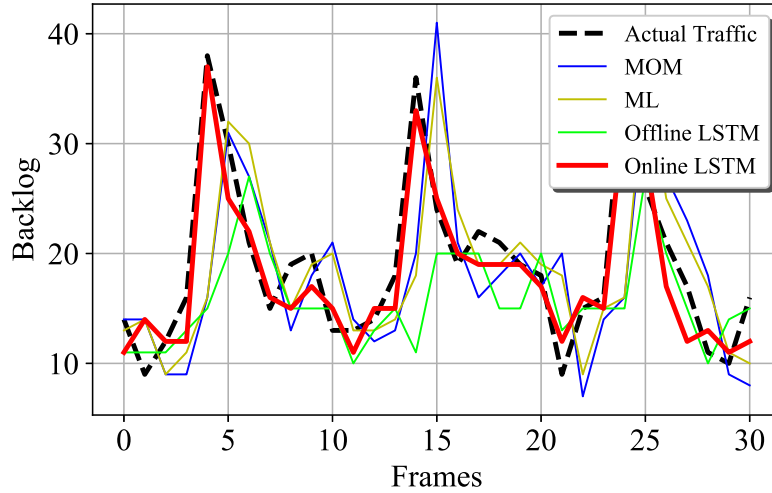

Fig. 3: Actual traffic and predicted backlog versus the number of frames after 10^5 online training frames.

We start by illustrating the operation of the proposed and reference predictors in Fig. 3. This figure plots the actual and predicted backlog after 10^5 frames along time frames. We observe that the only method that is able to predict the backlog spikes due to periodic traffic is the proposed Online LSTM method (the Genie-aided LSTM scheme is not shown). In fact, both MOM and ML are not capable of capturing historical trends in the traffic, and Offline LSTM has only observed data without packets from the periodic traffic.

Fig. 4 shows the evolution (averaged over 500 training trails) of the absolute prediction error $|\hat{N}^{t+1} - N^{t+1}|$ as a function of the frames observed in the online phase for the proposed online LSTM scheme and the genie-aided LSTM scheme. It is seen that the proposed online scheme is able to fairly quickly adapt to the

traffic conditions, improving over the performance of MOM and ML strategies and converging to those of the ideal genie-aided scheme. It is also noted that the LSTM scheme outperforms MOM and ML even with the initialized weights obtained from the offline training phase based only on random traffic.
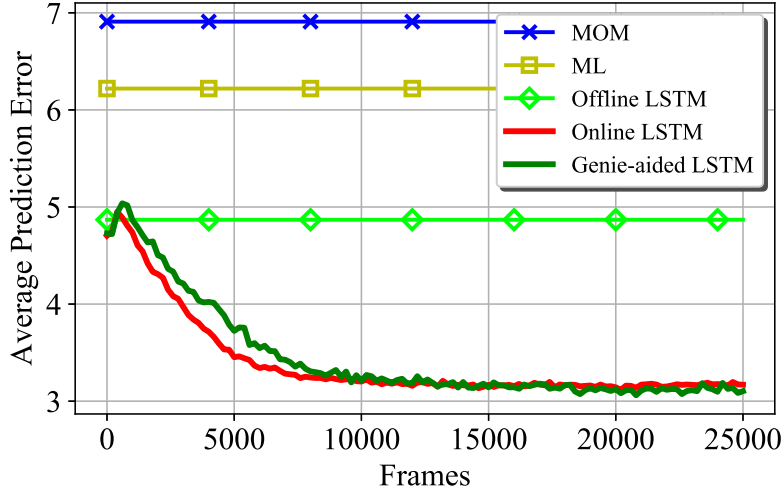

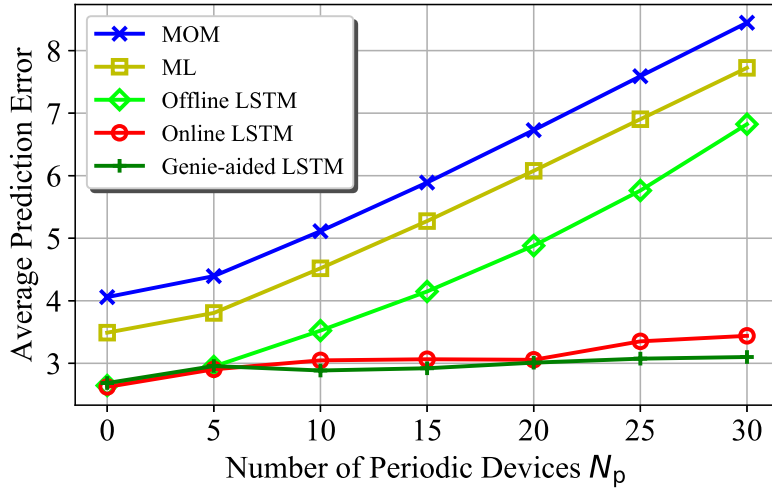Fig. 4: Average prediction error per frame as a function of the frames in the online adaptation phase.

Fig. 5: Average prediction errors per frame versus the number of devices with periodic traffic.

Fig. 5 plots the prediction error versus different number N_{p} of devices with periodic traffic. It can be seen that the ideal Genie-aided LSTM scheme slightly outperforms the proposed Online LSTM scheme, while their performance gap is small. Apart from Genie-aided LSTM, we observe that increasing N_{p} degrades the prediction accuracy of all methods, but the proposed online method demonstrates a significantly smaller

degradation as compared to the other strategies. This is due to the capability of the proposed scheme to adapt to the traffic statistics. It is also interesting to note that, even in the absence of statistical regularities in the traffic, i.e., with $N_\mathrm{p} = 0$, the LSTM-based schemes outperform MOM and ML. This is also because the random access communication protocol itself presents temporal correlations due to retransmissions of the same devices after collisions.

Fig. 6 shows the prediction error as a function of the packet generation period T_p of the periodic traffic. Increasing the period T_p results in a smaller traffic load, which improves the prediction accuracy of MOM, ML, and Offline LSTM. In contrast, as long as T_p is not too large, the prediction accuracy of the proposed LSTM is not significantly affected by the change in T_p due to its capability to adapt to the traffic statistics. Specifically, this is only true if $T_\mathrm{p} \leq T_o = 20$, that is, if the traffic periodicity is smaller than the memory of the LSTM predictor. In contrast, when $T_\mathrm{p} > 20$, the prediction accuracy of LSTM suddenly degrades to the same level of Offline LSTM scheme. This is because, with a memory equal to $T_o = 20$, the LSTM predictor cannot capture any traffic correlation pertaining to frames occurring more than 20 frames before the frame of interest. Therefore, if $T_\mathrm{p} > 20$, online adaptation cannot improve the prediction accuracy. This degradation can be eliminated by increasing the memory of LSTM T_o, but at the cost of increasing the required computational and data resources for both training and prediction.
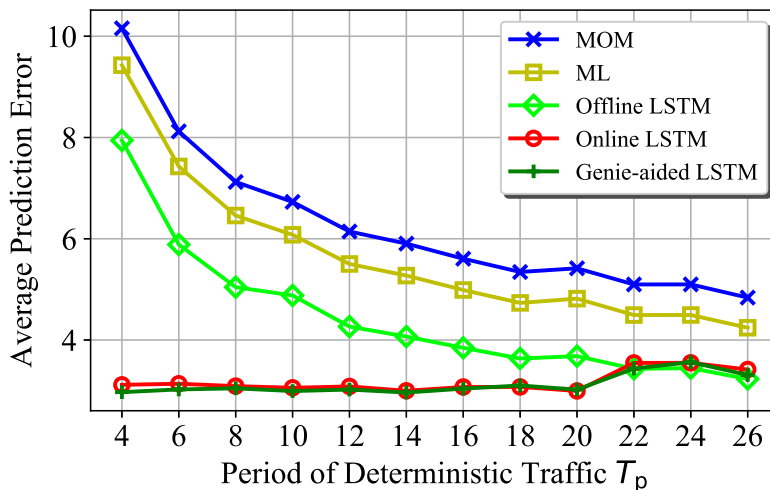

Fig. 6: Average prediction errors per frame versus the packet generation period of deterministic traffic.

We now consider a more general scenario, in which the $N_\mathrm{p} = 100$ devices with periodic traffic generate packets at random according to the time-limited Beta profile [2, Ch. 6.1] with period $T_\mathrm{p} = 10$. The time-limited Beta profile defines a probability of packet generation that peaks in the middle of each period with

burstiness defined by the magnitude of parameters (α, β) [2, Ch. 6.1]. Note that the deterministic packet generation considered so far is a special case of the current model, that is obtained by setting $\alpha = \beta \to \infty$. In Fig. 7, we plot the average prediction errors per frame versus a function of the burstiness parameter (α, β). We observe that increasing (α, β) degrades the prediction accuracy of all methods, since a higher burstiness results in a heavier traffic accumulation, which make prediction more difficult. The prediction accuracy of Offline LSTM is especially degraded, due to its lack of training observations under bursty traffic. In contrast, the proposed Online LSTM scheme only suffers from a minor performance degradation, demonstrating its capability to adapt to the traffic statistics.
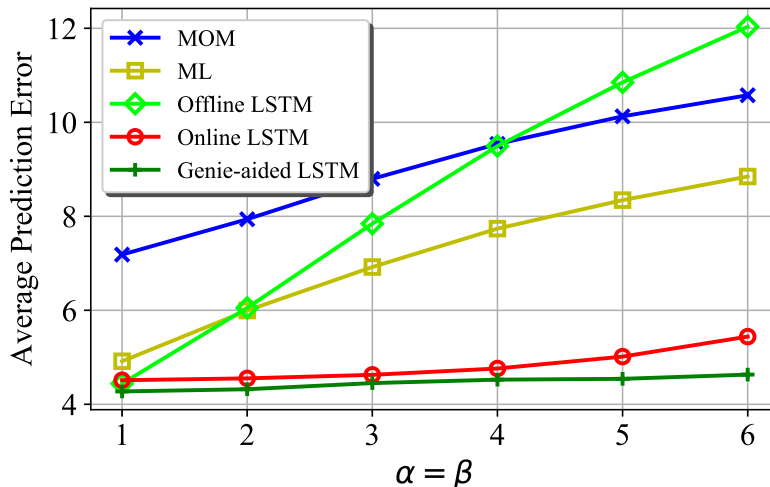

Fig. 7: Average prediction errors per frame versus the burstiness parameter (α, β) for time limited Beta distributed traffic.

VI. CONCLUSIONS

In this paper, we developed a traffic load prediction method based on online supervised learning in f-ALOHA networks. In the proposed method, LSTM RNNs are leveraged to capture temporal correlations due to traffic generation and protocol mechanisms. The scheme is based on a novel approximate labeling method based on backlog estimation. Numerical results demonstrate that the proposed method considerably outperforms conventional memoryless solutions, and that it can effectively adapt to new traffic statistics. A promising future direction is to develop lifelong learning and meta-learning techniques for online traffic prediction (see, e.g., [14]).

REFERENCES

[1] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "A first look at cellular machine-to-machine traffic: large scale measurement and characterization," *ACM SIGMETRICS Performance Evaluation Rev.*, vol. 40, no. 1, pp. 65–76, Jun. 2012.

[2] "Study on RAN improvements for machine-type communications," *3GPP TR 37.868 V11.0.0,*, Sep. 2011.

[3] F. Schoute, "Dynamic frame length ALOHA," *IEEE Trans. commun.*, vol. 31, no. 4, pp. 565–568, Apr. 1983.

[4] H. Wu, C. Zhu, R. J. La, X. Liu, and Y. Zhang, "Fast adaptive S-ALOHA scheme for event-driven machine-to-machine communications," in *IEEE Veh. Technol. Conf. (VTC Fall)*, Sep. 2012, pp. 1–5.

[5] S. Duan, V. Shah-Mansouri, Z. Wang, and V. W. Wong, "D-ACB: Adaptive congestion control algorithm for bursty M2M traffic in LTE networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9847–9861, Dec. 2016.

[6] H. He, P. Ren, Q. Du, L. Sun, and Y. Wang, "Traffic-aware overload control scheme in 5G ultra-dense M2M networks," *Trans. Emerging Telecommun. Technol.*, vol. 28, no. 9, p. e3146, Sep. 2017.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[8] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges," *IEEE Trans. Netw. Service Manag.*, Feb. 2019.

[9] N. Jiang, Y. Deng, O. Simeone, and A. Nallanathan, "Cooperative deep reinforcement learning for multiple-group NB-IoT networks optimization," *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP) invited paper*, May 2019.

[10] N. Jiang, Y. Deng, A. Nallanathan, and J. A. Chambers, "Reinforcement learning for real-time optimization in NB-IoT networks," *IEEE J. Sel. Areas Commun.*, 2019.

[11] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[12] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, USA: Springer print, 2006.

[14] S. Park, H. Jang, O. Simeone, and J. Kang, "Learning how to demodulate from few pilots via meta-learning," *arXiv preprint arXiv:1903.02184*, 2019.