Preamble Transmission Prediction for mMTC Bursty Traffic: A Machine Learning based Approach

Aasmund Søraa, Thilina N. Weerasinghe, Indika A. M. Balapuwaduge, and Frank Y. Li
Dept. of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway
Email: aasoeraa@gmail.com, {thilina.weerasinghe; indika.balapuwaduge; frank.li}@uia.no

***Abstract*—The evolution of Internet of things (IoT) towards massive IoT in recent years has stimulated a surge of traffic volume among which a huge amount of traffic is generated in the form of massive machine type communications. Consequently, existing network infrastructure is facing challenges when handling rapidly growing traffic load, especially under bursty traffic conditions which may more often lead to congestion. By proactively predicting the occurrence of congestion, we can implement necessary means and conceivably avoid congestion. In this paper, we propose a machine learning (ML) based model for predicting successful preamble transmissions at a base station and subsequently forecasting the possible occurrence of congestion under bursty traffic conditions. The model is composed of a recurrent neural network ML algorithm which is built based on the long short-term memory architecture. Through extensive simulations, we demonstrate that the proposed model achieves precise predictions on successful preamble transmissions relying merely on the data collected priori to congestion occurrence.**

I. Introduction

Machine type communications (MTC), which refer to facilitating communications among machines without (or with little) human intervention, have appeared as an integral component of fifth generation (5G) networks. Currently, the 3rd generation partnership project (3GPP) is developing specifications which enable 5G mobile and wireless networks to support massive machine type communications (mMTC). At a density of one million devices per square kilometer, mMTC will focus on providing wireless connectivity to a huge number of devices that may transmit data packets, which are typically small-sized, in a periodic or sporadic manner. Among potential scenarios for mMTC applications, critical infrastructure surveillance, environmental monitoring, intelligent transportation, smart city, and smart agriculture are a few examples.

With the increasing popularity of mMTC, how to upgrade the existing random access (RA) procedure adopted in long-term evolution advanced (LTE-A) to 5G networks becomes an imperative research task. For MTC, various RA techniques have been considered. Among them, multichannel slotted ALOHA schemes have been proposed for medium access of RA channels (RACH) [1]. However, the throughput of multichannel ALOHA schemes is not satisfactory in many mMTC scenarios due to the limit of the amount of available radio resources versus the number of devices [2]. Therefore it is necessary to improve such methods to accommodate medium access of mMTC devices.

A next generation nodeB (gNB) can receive numerous access requests simultaneously and it relies on a pre-determined set of orthogonal preambles to handle different requests and avoid collision during the initial access procedure. Under normal traffic conditions where the transmission intervals for MTC devices are pre-scheduled or known beforehand, the current RA scheme is sufficient to process all access requests. However, under abnormal conditions especially for *bursty traffic*, the access success probability could drop drastically due to the sudden jump of the number of access requests. A traffic burst occurs when a large number of devices attempt to transmit uplink data (almost) simultaneously, for instance, after a power failure or when a mission critical event occurs.

The LTE-A RA procedure lets each device randomly select one of the available preambles in a given RA slot, where the number of orthogonal preambles is limited. The same procedure for random access is adopted in 5G new radio (NR) Phase I as NR is also built based on the same medium access mechanism, i.e., orthogonal frequency-division multiplexing (OFDM). When a massive number of devices simultaneously request for medium access, a very high number of RA attempts would cause heavy preamble collisions. To diminish collision at the gNB, certain improvements which reduce the number of attempts per RA slot have been proposed. However, most of these schemes are *reactive*, meaning that no action is taken until congestion has occurred. In this paper, we propose a *proactive* solution which is built based on a recurrent neural network (RNN) machine learning (ML) algorithm to predict the number of successfully detected preambles in future RA slots. By proactive, it is meant that our prediction is performed based on gNB's observation over a short period of time *prior to the occurrence of traffic congestion*. Although other ML based solutions for network traffic prediction exist, they are not focused on bursty traffic in mMTC scenarios. Nor are they operated in a proactive manner. The accuracy of the model is validated via simulations under different traffic conditions.

The rest of this paper is structured as follows. An overview on RA and the RACH congestion problem is given in Sec. II together with a summary of the related work. Network scenario and assumptions are described in Sec. III and afterwards Sec. IV outlines the proposed ML based traffic prediction model. Performance evaluation of the proposed model is presented in Sec. V. Finally, the paper is concluded in Sec. VI.

II. Background and Problem Statement

In this section, an overview of the LTE-A/NR RA procedure is provided. We present also an observed research gap related

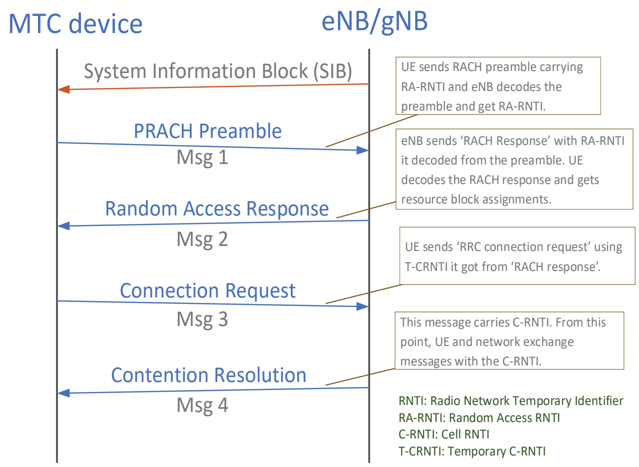


Fig. 1: Contention based RA procedure for LTE-A and NR.

to mMTC RA congestion and summarize existing work.

A. Random Access in LTE-A and 5G NR

For contention based RA, 5G NR supports the same 4-step RACH handshake procedure as adopted in LTE-A [3]. Specifically, the Zadoff-Chu sequences are used for generating RA preambles for initial access. When an MTC device needs to start a data transmission, it initiates RA during an allowable RA time slot by randomly picking a preamble. If more than one device selects the same preamble and the gNB does not sense a collision, a contention resolution process is required. A contention based RA procedure consisting of 4 steps is illustrated in Fig. 1.

The transmission of a preamble from a device to its gNB is known as *MSG-1*. If the gNB detects the message successfully, it replies with an RA response (RAR) message, *MSG-2*, with a specific radio access radio network temporary identifier (RA-RNTI). Afterwards, the device sends *MSG-3* as a connection request based on its RA-RNTI, followed by an acknowledgment from the gNB as *MSG-4*. If a device does not receive the contention resolution message, it declares a failure for its access attempt and the same procedure will start again.

Although increasing RACH resources seems to be a solution to improve the performance of RA, it is constrained by the number of orthogonal preambles available in a slot. According to [4], there are 64 preambles per slot that can be allocated in a cell with a coverage radius of 7.4 km and a delay spread of 6 μs. Among these 64 preambles, a small amount of them, typically 10, are reserved for contention-free transmissions.

B. Collision Observation and Traffic Prediction

If an MTC device fails to transmit a preamble, it will wait for a short random time interval within a backoff window and restart a new RA preamble transmission until its retry limit has been reached. Consequently, upon a sudden incident, for instance, a traffic burst after a power reset, an enormous number of devices might compete for channel access simultaneously causing network performance degradation in terms of both higher collision and increased access delay.

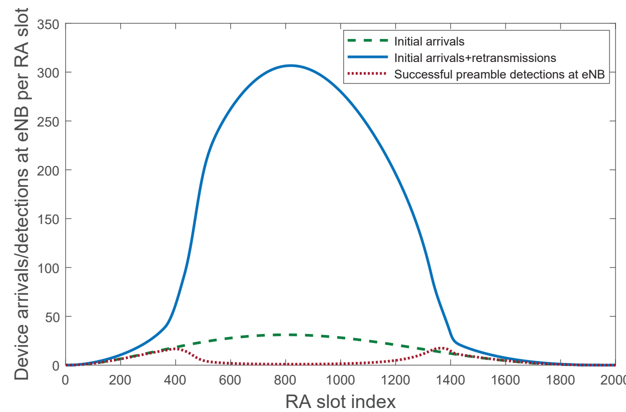


Fig. 2: Number of total arrivals and successful preamble detections in LTE-A RA under bursty traffic: 30k devices and 54 preambles [5].

When RACH congestion happens, a gNB cannot estimate the total number of devices attempting for RA (which includes both initial arrivals and retransmissions) in a given RA slot. Instead, the gNB has only knowledge on how many preambles have been successfully detected at each RA slot *but not beyond the current RA slot*. As an example, we illustrate in Fig. 2 the total number of arrivals per RA slot which consists of both new arrivals and the retransmissions from unsuccessful devices versus the number of successful preamble transmissions. It is evident that, out of those total arrivals, the gNB can only decode correctly a few of them. In this figure we observe that congestion starts occurring around RA slot index 400, afterwards a spike of new arrivals and retransmission emerges.

Once congested, the gNB notices a sharp descend in number of successful attempts but it does not know how many devices are transmitting simultaneously since a collision could be caused by two or more concurrent transmissions. In Sec. IV below, we propose an RNN model to predict the number of successful preamble detections based on gNB's observation *before* congestion happens. It is worth mentioning that while our previous work [5] predicted the number of total arrivals including both initial and retransmissions, this paper focuses on predicting the number of preambles successfully detected at the gNB. Another difference between them is that the initial data set used for traffic prediction in this study is merely the data collected up to a point after which congestion might happen (e.g., when the RA slot index is close to 400 in Fig. 2) and the prediction can be long-term (e.g., for RA slot index of 2000), whereas all data accumulated up to the prediction instant are needed for arrival estimation performed in [5].

C. Related Work versus This Study

Congestion avoidance in LTE-A networks has been studied intensively. In [6], a list of improvements to alleviate traffic overload caused by MTC devices has been released where mechanisms such as access class barring (ACB) and extended access barring (ECB) are among the most eminent ones. ACB allows MTC devices to transmit their connection requests with different probabilities based on a barring rate broadcast by the associated eNB. In order to maintain the quality of human type communications (HTC), dedicated RACH for MTC devices was also proposed in [6]. However, in certain traffic scenarios, dedicated RACH could lead to poor performance due to RACH underutilization. In addition, ECB introduces more strict constraint on devices such that certain classes of devices are restricted from obtaining access when congestion occurs.

On the other hand, ML based solutions have recently attracted lots of attention as a means to predict wireless traffic and resolve congestion issues [7]. In [8], a Gaussian process based prediction model for 4G traffic data was proposed, however, without considering initial access. In [9], a delay-aware RA scheme for mMTC devices was proposed through an online hierarchical stochastic learning algorithm. In [10], reinforcement learning based eNB selection methods were studied as an effective solution to support efficient RA and avoid congestion for a large number of devices. Moreover, to solve the RACH congestion problem of MTC in LTE-A networks, [11] modeled the ACB decision process using a Q-learning algorithm. However, ACB schemes cannot provide higher access efficiency when the number of MTC devices accessing a gNB is very large. Although various approaches for achieving efficient MTC in LTE-A have been proposed, how to efficiently handle RA opportunities for a massive number of devices under bursty traffic conditions in NR networks requires further investigation.

In contrast to the aforementioned existing work, our approach exhibits the following three salient features. First, the proposed prediction model is based on an RNN model and is dedicated to dealing with bursty traffic. Second, it presents a prediction of preamble detections for initial access that happens priori to data transmission. Third, the prediction is performed in a proactive manner as mentioned above.

III. Network Scenario and Assumptions

Consider an MTC network under bursty traffic conditions based on a traffic model specified by 3GPP. Bursty traffic conditions occur when a large number of devices attempt to transmit event-driven data abruptly and concurrently.

Although the Poisson arrival process is regarded as probably the most popular traffic model for HTC traffic sources, it is not recommended for MTC traffic since it does not capture the burstiness of MTC traffic distributions. According to [6], under bursty conditions, the access intensity at RA slot i, $A(i)$, is given by

$$A(i) = N \int_{t_i}^{t_{i+1}} p(t)\, d(t)\,. \tag{1}$$

The above integral is done over an interval $[t_i, t_{i+1}]$, where t_i is the time at access opportunity i, t is defined in the range $[0, T]$, when T is the observation window and N is the number of devices that are active during time T. Each MTC device is activated at time t where $0 \le t \le T$ with probability $p(t)$ following beta distribution with parameters $\alpha = 3, \beta = 4$ as

$$p(t) = \frac{t^{\alpha-1}(T-t)^{\beta-1}}{T^{\alpha+\beta-1}B(\alpha,\beta)} \tag{2}$$

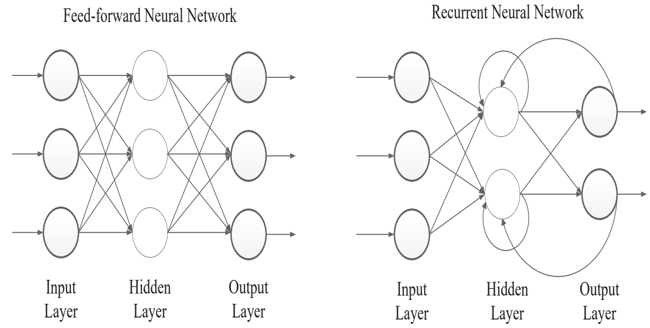


Fig. 3: Typical structures of feed-forward and recurrent NNs.

where $B(\cdot)$ is the beta function. For bursty traffic, T is considered to be 10 and 60 seconds long respectively.

The number of active devices in our MTC scenario is considered to be 10k, 20k, or 30k per slot respectively. Furthermore, we do not consider any channel impairment such as path loss, fading, or inter-cell interference in this study.

IV. Proposed RNN based Traffic Prediction Model

In this section, we propose an ML based traffic prediction model for bursty traffic in mMTC networks by exploiting the capability of recurrent neural networks.

A. *Properties of Recurrent Neural Networks*

A neural network (NN) is a network of computational nodes (neurons) consisting of three or more layers. The first layer in an NN is the input layer, and the nodes in this layer contain the input data used for training. The last layer is the output layer which produces the outputs of the entire neural network. Between these two layers, there exists one or multiple hidden layers with multiple hidden nodes which perform computations and transfer information from input nodes to output nodes. In regular NN architectures, data is processed in a *feed-forward only* fashion, where the input data is computed in the first hidden layer, and the result in each node will be passed to the corresponding nodes in the next layer. Feed-forward neural networks (FFNNs) have been applied to time series prediction for data sets.

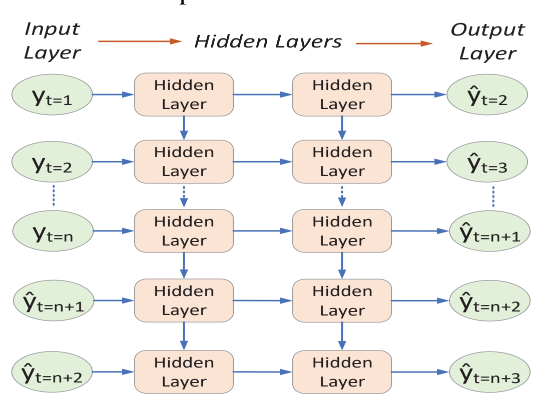


Fig. 4: Structure of the LSTM-RNN model with two hidden layers.

Unlike FFNNs, RNNs use feedback loops to process a sequence of inputs, and the output from one layer could be fed back to the same or a previous layer. Fig. 3 illustrates the types of layers and the differences between an RNN and an FFNN. Improved performance for time series tasks has been achieved in RNNs [12]. In this work, we adopt a long short-term memory (LSTM) network which is an enhanced version of RNNs as the basis for our prediction model, since it allows learning of long-term dependencies of a system. The length of a sequence that an RNN can interpret is rather limited, especially in comparison with LSTMs. Therefore, an LSTM based prediction model is a better choice for this study since a bursty traffic condition brings out a comparatively long sequence of data. One of the important features of LSTM networks is that it introduces gates to specify how much past information could be let through. In our proposed model, this feature is adopted for MTC traffic prediction.

B. LSTM Prediction Model

To estimate the number of successful preamble detections at the gNB and predict the congestion status, an LSTM model is developed. The model is structured as a multi-step prediction with two hidden layers where a set of *input seeds* from traffic arrival values are used as the basis for new, predicted values. The input seed, denoted as Y_n, consists of a set of n number of input values $Y_n = (y_1, y_2, ..., y_n)$ and it is used to predict the next value, denoted as $\hat{y}_{n+1}$. In order to predict the value after $\hat{y}_{n+1}$, i.e., $f(Y_{n+1}) = \hat{y}_{n+2}$, the set of input values will include the previous *estimated-output* so that $Y_{n+1} = (y_1, y_2, ..., y_n, \hat{y}_{n+1})$. This approach is implemented by introducing a sliding window, where the window size $|Y|$ either is fixed or increases by one at each prediction. Specifically, (3) is the function for a fixed sliding window at two consecutive time steps (denoted as t_0 and t_1), showing that $\hat{y}_{n+2}$ is estimated based $n-1$ observed values, $y_2, y_3, ..., y_n$, plus one estimated value, i.e., $\hat{y}_{n+1}$.

$$\begin{aligned} t_0 &: \ f\left(y_1, y_2, ..., y_n\right) = \hat{y}_{n+1}, \\ t_1 &: \ f\left(y_2, y_3, ..., \hat{y}_{n+1}\right) = \hat{y}_{n+2}. \end{aligned} \tag{3}$$

Furthermore, (4) shows the function when an increasing window is adopted. Note that the difference between (3) and (4) is that the y_1 term at time t_1 is included as an input in (4) only. In other words, with the fixed sliding window prediction, *after n steps, the new predictions are purely based on the previous predicted values*. On the other hand, in the increased sliding window prediction, the initial input will always be used for predictions at each step. Thus higher computational complexity is expected in the latter case.

$$\begin{aligned} t_0 &: \ f\left(y_1, y_2, ..., y_n\right) = \hat{y}_{n+1}, \\ t_1 &: \ f\left(y_1, y_2, ..., y_n, \hat{y}_{n+1}\right) = \hat{y}_{n+2}. \end{aligned} \tag{4}$$

One salient feature of the developed LSTM model in this study is that the model exploits the advantages of both fixed and increasing sliding window functions. The *hidden states* in our LSTM model keep track of the previous computations at the current time-step, such that when predicting the value at time $t+1$, the function adopts only the summarized values stored in the hidden states. This ensures that the model always includes the information from the originally collected values when predicting, however, without increasing computational complexity. The architecture of the adopted LSTM-RNN is illustrated in Fig. 4.

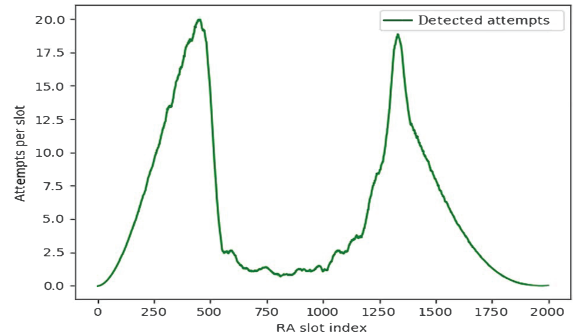


Fig. 5: Successful (detected) attempts of a bursty traffic scenario when 30k MTC devices in the network.

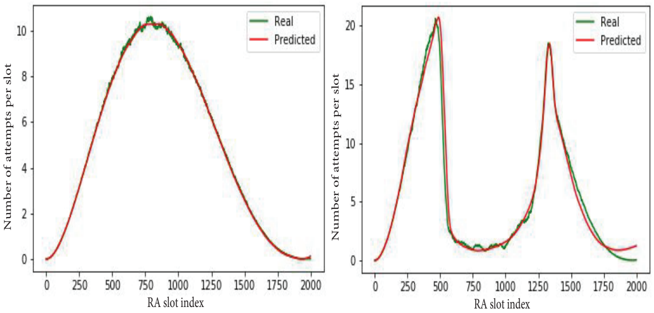
In addition to the input-output relationship, Fig. 4 illustrates also how the evaluation for the next time-step is based on the previous time-steps. Herein, we provide the collected values for the first n number of inputs. Each of the first n inputs to the LSTM-RNN model generates a single output. For instance, the input at time $t = 1$, i.e., $y_{t=1}$ generates output value $\hat{y}_{t=2}$ at time $t = 2$. During each round, the impact of the previous input data series exists due to the feedback process of hidden layers. However, *after the first n inputs, there is no detected data. That is, only predicted data exist*. Therefore, the previous prediction is fed to the LSTM-RNN model as the input. For instance, the predicted value at time $t = n + 1$ is fed as the input at $t = n + 2$. Another feature of our model is that even without detected values after n rounds, the predicted values still have impact from the first n number of collected data.

TABLE I: Simulation parameters of the LTE-A RA process

RACH Parameter	Value
Number of contention-based preambles	54
Physical RACH configuration index	6
Max. number of transmissions	10
Backoff value (in terms of subframes)	20

V. Traffic Analysis and Performance Evaluation

Real-life datasets for bursty traffic are not easily available for ML based traffic prediction studies. Therefore, we follow the model proposed by 3GPP [6] to generate traffic based on simulations through proper parameter configurations. The proposed prediction model has been implemented in a custom-built simulator written in Python (v3.6.7) [13]. Based on the parameters listed in Table I, different traffic patterns have been generated in our simulations. In Fig. 2 above, we have illustrated the generated bursty traffic for 30k MTC devices including both initial and retransmissions together with the successful preamble detections. In Fig. 5, the successful detections are highlighted. Note that the data shown herein is smoothened through a Savitzky-Golay [14] filter with a window size of 97 and a 2nd degree polynomial in order to improve the quality of the generated data. This filtering mechanism is performed to create continuous values from the



(a) 10k MTC devices (b) 30k MTC devices

Fig. 6: Scenario 1: Multi-step prediction using 100 RA slots as input seed with 10k and 30k MTC devices respectively.

discrete simulation output. As detected data, we input only a part of the data shown in Fig. 5 to the developed LSTM network. Accordingly, we define a parameter, *seed length* L corresponding to the number of input seeds, in Fig. 4, i.e., $L = n$. When $L = 100$, for instance, we only take data corresponding to the first 100 RA slot indexes as data inputs.

Three traffic groups, each consisting of 10k, 20k, and 30k MTC devices respectively, are considered in our simulations. This is a reasonable configuration in terms of number of mMTC devices since it is meant for the number of active devices arriving within one RA slot which varies from 62.5 μs to 1 ms depending on the numerology adopted in 5G NR. Two test scenarios, *homogeneous* (Scenario 1) and *heterogeneous* (Scenario 2), are defined. The LSTM network is configured with 2 hidden layers. The model was trained through an *Adam* optimization function with a learning rate of 0.0001 per iteration. This rate is selected from a set of learning rates considering the tradeoff between learning output and efficiency. In addition, the hyperbolic tangent (tanh) function is used as the activation function of the model.

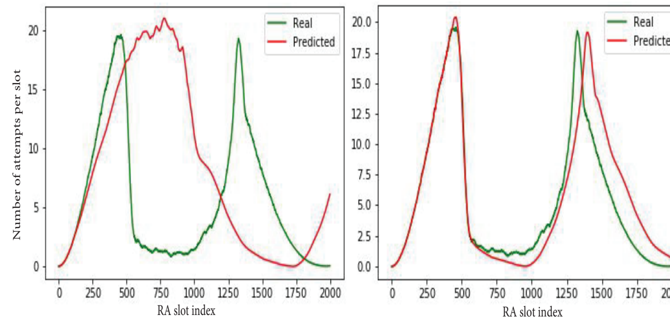
A. Homogeneous Scenario

In test Scenario 1, the multi-step prediction model is trained on each traffic group independently. This training model is tested based on the data from the respective group. Thus the model predicts upcoming access attempts in an mMTC network where the number of active devices in a cell is known. To assure that the model is able to learn the features of the data set, the data sets of each group were trained separately.

Fig. 6 depicts the prediction results of the LSTM network corresponding to 10k and 30k devices respectively. In both configurations, the predicted curves fit the original ones precisely showing that very high accuracy has been achieved by the model. Owing to a lower traffic intensity with 10k devices, the prediction accuracy is even higher than the 30k MTC case. To verify the accuracy, the mean squared errors (MSEs) between the collected and predicted traffic are calculated and listed in Table II.

B. Heterogeneous Scenario

In test Scenario 2, the multi-step prediction model is trained based on the traffic generated simultaneously from a combination of two groups, e.g., 10k & 20k and 10k & 30k etc. Accordingly, the model is evaluated by predicting the number of successful detections in a *dynamic* network environment where the number of active devices is unknown. This represents a more complicated situation where the number of *active nodes*, i.e., nodes with packets to send, is unknown to the gNB.



(a) 30k devices, $L = 100$ (b) 30k devices, $L = 400$

Fig. 7: Scenario 2: Multi-step prediction trained based on a combination of data from 20k and 30k devices but tested on data from 30k devices with different seed lengths.

TABLE II: Effect of using different input lengths measured in MSE between real and predicted traffic

Seed length	MSE for 10k group	MSE for 30k group
100	0.0070	1.9281
300	0.0059	1.7874
400	0.0054	0.4937

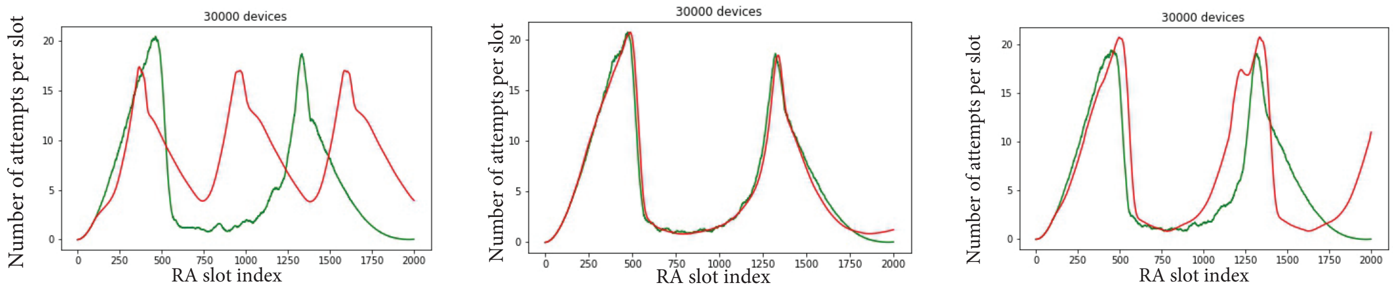
To predict successful preamble detections in the heterogeneous scenario needs to deal with the data generated from two groups without group identification. Herein, we present the results for a heterogeneous group with 20k and 30k devices where the model was trained based on the whole group.

The performance of the predicted model is illustrated in Fig. 7 considering that the traffic is only generated from the 30k device group. As shown in the figure, with a short seed length of 100 RA slots, the prediction is not accurate. However, when the seed length is increased to 400, the obtained prediction result has achieved very high accuracy. From this observation it is clear that when more training data or input seeds are utilized in the LSTM network, the model could reach more accurate predictions.

C. Number of Nodes/Neurons in a Hidden Layer

The number of nodes in a hidden layer, referred to as *width*, is a crucial parameter which affects the performance of the LSTM network. In general, there is no single rule on how to determine the width of an LSTM since it depends on several factors such as the complexity of the dataset, the amount of features, and the number of data points.

In Fig. 8, we exhibit the impact of width on the prediction results in this study. As shown in the figure, the accuracy of the prediction is heavily depending on this parameter, W of the LSTM network. When $W = 1100$, the model predicts more accurately the number of successful preamble detections than that of $W = 500$. However, the prediction accuracy does not improve when further increasing W. This is because too



(a) 500 nodes/neurons in a hidden layer. (b) 1100 nodes/neurons in a hidden layer. (c) 1500 nodes/neurons in a hidden layer.

Fig. 8: Effect of different number of hidden nodes in 2-layer LSTM network for forcasting on 30k MTC devices. The green and red curves in the sub-figures represent the detected and predicted values respectively.

many nodes may overfit the data causing poor generalization on data not used for training, while too few hidden nodes underfit the model, and is not sufficiently accurate [15]. Note that the accuracy of the LSTM model is further improved by its ability to decide which new information to integrate into the model and which unnecessary information to ignore.

D. Congestion Status Assessment

Finally, we present a traffic congestion assessment method which could be used by a gNB to forecast congestion status based on its predicted successful preamble detection. Such an assessment is made based on a simple single-layer FFNN in which logistic regression is applied to predict a binary class, i.e., congestion or non-congestion by utilizing the well-known sigmoid function $f(x) = (1 + e^{-x})^{-1}$ where x is the input to the function which provides the probability estimate.

The results shown in Table III are obtained based on a heterogeneous scenario with a combination of 20k and 30k devices, for both short- (1000 slots) and long-term (2000 slots) forecast. Clearly, when the seed length is too short, the assessment is not precise enough. With a longer input seed, however, our method is able to predict the correct class with sufficiently high accuracy. Therefore, this method provides the gNB with a capability for congestion status assessment solely based on its prediction of successful preamble detection.

TABLE III: Accuracy of congestion classification assessment

Seed length	Classification Accuracy	
	Length of the traffic forecast = 1000 slots	Length of the traffic forecast = 2000 slots
100	0.5%	5%
200	10%	10%
300	80%	80%
400	90%	90%

VI. Conclusions

Based on the principle of recurrent neural networks, we have proposed a traffic prediction model for successful preamble detections in mMTC networks under bursty traffic conditions. Although RNNs have been a popular tool for time series data analysis, its applicability to mMTC networks remains unclear as the data in our scenarios is univariate. To ensure the best foundation for our model, we first analyze simulated traffic patterns to learn more about different traffic conditions. Based on the information gathered, an LSTM network is trained in one-step prediction, and the predictions are utilized in a recursive multi-step prediction scheme for successful preamble detections. According to the obtained simulation results, the proposed LSTM-RNN model with appropriate network configurations can precisely predict the number of preamble detections. Moreover, the model is capable of assessing the occurrence of congestion right before a traffic burst.

References

[1] C.-H. Wei, R.-G. Cheng, and S.-L. Tsao, "Modeling and estimation of one-shot random access for finite-user multichannel slotted ALOHA systems," *IEEE Commun. Lett.*, vol. 16, no. 8, pp. 1196–1199, Aug. 2012.

[2] J. Choi, "On the stability and throughput of compressive random access in MTC," in *Proc. IEEE ICC*, May 2017, pp. 1–6.

[3] 3GPP TS 36.321, "Evolved universal terrestrial radio access (E-UTRA), Medium access control (MAC) protocol specification," R16, v16.0.0, Mar. 2020.

[4] M. Rahnema and M. Dryjanski, *From LTE to LTE-Advanced Pro and 5G*, Artech House, 2017.

[5] T. N. Weerasinghe, I. A. M. Balapuwaduge, and F. Y. Li, "Supervised learning based arrival prediction and dynamic preamble allocation for bursty traffic," *IEEE INFOCOM Workshops*, Apr. 2019, pp. 1-6.

[6] 3GPP TR 37.868, "Study on RAN improvements for machine-type communications," R11, v11.0.0, Sep. 2011.

[7] H. Chergui and C. Verikoukis, "Offline SLA-constrained deep learning for 5G networks reliable and dynamic end-to-end slicing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 350–360, Feb. 2020.

[8] Y. Xu, W. Xu, F. Yin, J. Lin, and S. Cui, "High-accuracy wireless traffic prediction: A GP-based machine learning approach," in *Proc. IEEE GLOBECOM*, Dec. 2017, pp. 1–6.

[9] Y. Ruan, W. Wang, Z. Zhang, and V. K. N. Lau, "Delay-aware massive random access for machine-type communications via hierarchical stochastic learning," in *Proc. IEEE ICC*, May 2017, pp. 1–6.

[10] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in LTE-Advanced networks: Issues and approaches," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 86–93, Jun. 2013.

[11] J. Moon and Y. Lim, "A reinforcement learning approach to access management in wireless cellular networks," *Wirel. Commun. and Mobile Comput.*, vol. 2017, article ID. 6474768, May 2017.

[12] D. Andreoletti, S. Troia, F. Musumeci, S. Giordano, G. Maier, and M. Tornatore, "Network traffic prediction based on diffusion convolutional recurrent neural networks," in *Proc. IEEE INFOCOM Workshops*, Apr. 2019, pp. 246–251.

[13] A. Søraa, "A machine learning based prediction model for bursty traffic in 5G mMTC networks," *Master's Thesis in Information and Communication Technology*, University of Agder, Norway, 2019, [Online] Available: https://uia.brage.unit.no/uia-xmlui/handle/11250/2618717.

[14] S. R. Krishnan and C. S. Seelamantula, "On the selection of optimum Savitzky-Golay filters," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 380–391, Jan. 2013.

[15] L. Fletcher, V. Katkovnik, F. E. Steffens, and A. P. Engelbrecht, "Optimizing the number of hidden nodes of a feedforward artificial neural network," in *Proc. IEEE Int. Joint Conf. on Neural Networks*, May 1998, pp. 1–5.