A Deep Learning Model for Wireless Channel Quality Prediction

J. Dinal Herath, Anand Seetharam, Arti Ramesh
Department of Computer Science, SUNY Binghamton, USA
jherath1@binghamton.edu, aseethar@binghamton.edu, artir@binghamton.edu

***Abstract*—Accurately modeling and predicting wireless channel quality variations is essential for a number of networking applications such as scheduling and improved video streaming over 4G LTE networks and bit rate adaptation for improved performance in WiFi networks. In this paper, we propose an encoder-decoder based sequence-to-sequence deep learning model that is capable of predicting future wireless signal strength variations based on past signal strength data. We consider two different versions of the deep learning model; the first and second versions use LSTM and GRU as their basic cell structure, respectively. In contrast to prior work that is primarily focused on designing models for particular network settings, the deep learning model is highly adaptable and can predict future channel conditions for different networks, sampling rates, mobility patterns, and communication standards. We compare the performance (i.e., the root mean squared error of future predictions) of our model with respect to two baselines—*i)* auto-regression(1), and *ii)* linear regression for multiple networks and communication standards. In particular, we consider 4G LTE, WiFi, an industrial network operating in the 5.8 GHz range, Zigbee, and WiMAX networks operating under varying levels of user mobility and observe that the deep learning model provides significantly superior performance. Finally, we provide detailed discussion on key design decisions including insights into hyper-parameter tuning of the model.**

***Index Terms*—wireless channel prediction, deep learning, sequence-to-sequence models, LSTM, GRU**

I. Introduction

Modeling and accurately predicting wireless channel quality variations (e.g., received signal strength) has received significant attention in wireless communications and networking research, starting from the early Gilbert and Elliot two-state Markov channel model [6]. Most prior research has been focused on designing Markovian models that elegantly capture the impact of wireless channel characteristics such as multi-path fading, shadowing and path loss on the received signal strength [3], [11]. Though these models provide valuable insight, majority of these models are tied to particular network settings and are dependent on parameters such as sampling rate, mobility, and location. Thus, they cannot be seamlessly used for predicting signal strength across different wireless networks.

Revisiting the channel prediction problem in today's data-driven Internet-of-things era is extremely important, particularly due to the exponential growth in the number of diverse wireless devices that communicate with each other using a variety of technologies (e.g., WiFi, 4G LTE, Zigbee) in different wireless scenarios (e.g., home, commercial, industrial). Additionally, the rapid increase in computational power over the last decade and the availability of large amounts of data, coupled with advances in the field of machine learning provide us the opportunity to design models that provide superior prediction performance of wireless channel quality variations [10]. Multiple foreseeable applications motivate this research such as better scheduling and improved video streaming over 4G networks, bit rate adaptation for improved performance in WiFi networks, and energy efficient and bulk transfer of data in sensor networks.

In this paper, we design deep learning models to address the wireless channel quality prediction problem. Specifically, we design an encoder-decoder based sequence-to-sequence deep learning model, which is capable of predicting variations in wireless channel quality. Our goal is to design a deep learning model that can effectively capture channel quality variations in different network settings, in a variety of mobile scenarios, and works across communication standards and for different sampling rates. Our model is comprised of two main components—-*i)* an encoder and *ii)* a decoder, each of which separately is a multi-layer recurrent neural network (RNN). The encoder takes past signal strength measurements and computes a state vector that captures channel information. The decoder in turn uses this state vector to predict future channel variations. We propose two variations of the model based on the inner cell architecture used in the encoder and decoder, namely, a long short-term memory (LSTM) variant and a gated recurrent unit (GRU) variant.

To demonstrate the widespread applicability and efficacy of our model, we conduct experiments on received signal strength data collected over different kinds of networks including 4G LTE, WiFi, Zigbee, WiMAX, and in an industrial network setting. Additionally, we investigate the predictive capability of our model on data collected in these networks on different time granularities and in varying mobility scenarios. We compare the performance of our model with two baselines—auto-regression(1) and linear regression, and show that our deep learning model outperforms the baselines in all scenarios. Our experiments show that auto-regression(1), the model that uses the least historical information performs the worst. Interestingly, we observe that the deep learning model provides higher performance gains for network settings with higher signal strength variations and less seasonality, which demonstrate the superiority of the model. Finally, we provide a discussion on key design decisions in training methodology and insight into hyper-parameter tuning of the model.

II. Related Work

Wireless channel quality prediction is a well-studied domain, with the earliest work in this space being the two-state Gilbert and Elliot Markov model. Research in this field can be broadly categorized into—*i)* Markovian models that model variations in the received signal strength, and *ii)* machine-learning models for predicting future wireless channel conditions.

The networking literature is rife with Markovian models for wireless signal strength prediction. Sadeghi et al. [11] and Bui et al. [3] provide detailed surveys of finite state Markovian models designed for modeling the wireless channel and their evolution over time. In [12], the authors design a coarse time scale model for capturing the effect of shadowing on the received power. Other recent work utilizing Markovian models for channel prediction include spectrum sensing utilizing a hidden bivariate Markov chain [2] and modeling channel variations for vehicular networks [4]. While Markovian models offer insight into wireless channel variations, prior work by Wang et al. [13] note that higher order Markovian models that utilize more historical information are necessary to obtain better performance.

Prior work focusing on the use of machine learning for channel prediction include predicting link quality for wireless sensor networks [9], identifying critical links [8] and spatio-temporal modeling and prediction in cellular networks [14]. Additionally, in recent years, deep learning techniques have also been applied to solve various problems in wireless communications. A comprehensive survey by Mao et al. [10] identifies many opportunities for the use of deep learning in wireless networks and emphasizes the capability of deep learning models. Some examples of designing deep learning models include device-free wireless localization using shadowing effects [15] and spectrum sharing in heterogeneous wireless networks [16]. In contrast to prior work, we design a deep learning model for received signal strength prediction and demonstrate its applicability for a variety of network settings and communication standards.

III. Problem Statement

Several factors cause sudden variations in the wireless channel quality, thus posing challenges in developing a generalized framework for this prediction task. *In this work, we develop an encoder-decoder based sequence-to-sequence deep learning predictive model, which is capable of accurately predicting wireless channel variations irrespective of mobility pattern, communication standard, and sampling rate.* Sequence-to-sequence models are ideally suited for problems that require mapping input sequences to output sequences and have been extensively used for tasks such as video captioning and natural language translation [7]. Recent work [7] has also demonstrated the applicability of these models for forecasting and prediction purposes where the objective is to predict the future based on past time series data. Therefore, for the problem studied here, at each time T, our deep learning model uses a sequence of past signal strength measurement values in a window size of n (i.e., $X_T = [x_{T-n}, x_{T-(n-1)}, ..., x_{T-1}, x_T]$) and predicts channel variations for k steps into the future (i.e., $\hat{Y}_T = [\hat{y}_{T+1}, \hat{y}_{T+2}, ..., \hat{y}_{T+(k-1)}, \hat{y}_{T+k}]$).

IV. Sequence-to-Sequence Deep Learning Model

We design an encoder-decoder based sequence-to-sequence model for solving the channel quality prediction problem (Figure 1). Our model has two main components—an *encoder* and a *decoder*. Both the encoder and decoder use RNN as the underlying neural network architecture. RNNs are a deep neural network architecture particularly suited for sequence-to-sequence modeling. RNNs are a network of nodes organized into sequential layers, each node in a given layer having a directed connection to every other node in the next successive layer. Additionally, RNNs also possess internal memory that remembers state, thus making them attractive for sequence-to-sequence models for temporal data. We refer the reader to Goodfellow et al. [7] for additional details on the functionality of RNNs.

The encoder receives the past signal strength measurements X_T and produces a context vector C (i.e., the encoded state) that summarizes the input sequence X_T. The decoder receives this as an input and in turn produces $\hat{Y}_T$, the predicted channel variations. An encoder-decoder based sequence-to-sequence model has the benefit of not being constrained to use the same sequence lengths for input and output (i.e., $n \neq k$) unlike standard RNN architectures [7].

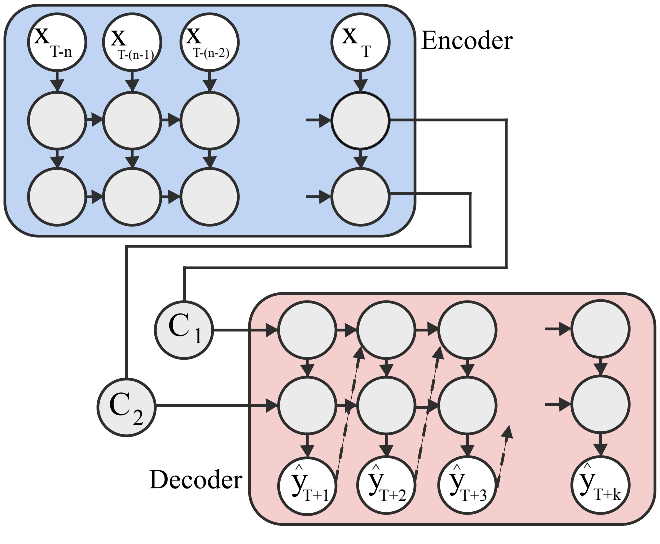


Fig. 1: Encoder-decoder based sequence-to-sequence architecture. In our model the "basic cell" is either LSTM or GRU.

In our model, both the *encoder* and the *decoder* operate as a deep RNN unit with two stacked layers of basic cell structures. We observe empirically that 2 stacked layers with 100 hidden units provide the best performance, and hence, we consider it as the architecture of our model. We present in-depth insight into the rationale behind using a stacked RNN structure in addition to other design decisions in Section VI-C. In a standard RNN, the nodes (the building blocks of a neural network architecture) are usually composed of basic activation functions such as *tanh* and *sigmoid*. Since RNN weights are learned by backpropagating errors through the network, the use of these activation functions can cause RNNs to suffer from the vanishing/exploding gradient problem, that causes the gradient to have either infinitesimally low or high values, respectively. This problem hinders RNN's ability to learn long-term dependencies [7]. To circumvent this problem, LSTM and GRU cells were proposed; they create paths through time with derivatives that do not vanish or explode [7] by incorporating the ability to "forget". Therefore, we consider two variations of our model (Figure 1) based on the basic cell structure used internally at each multi-layer RNN, namely, a LSTM version and a GRU version. As further elaborated in [7] both LSTM and GRU cells are composed of a number of gated units and they primarily differ in the number of gates and their interconnections. LSTM consists of three gates namely, the input gate, the output gate, and the forget gate that lets it handle long-term dependencies. In comparison, the GRU cell consists of two gates, a reset gate that combines the current input with previous memory and an update gate that determines the percentage of previous state to remember.

At each unfolded step $t \in [T-n, T]$ in the encoder, the signal strength measurement $x_t \in X_T$ is fed into the first cell of the *encoder*. The output of this cell is subsequently fed into the cell that follows. The final output from the encoder at each unfolded step t is discarded. Once the encoder computes the context vectors C_1 and C_2, they are fed into the decoder for prediction. Unlike machine translation tasks that use a flagged value for the initialization of the first *decoder* output [7], we use the final signal strength measurement (i.e., x_T), due to the correlation between $\hat{y}_{T+1}$ and x_T. At each unfolded decoder step $t^{'} \in [T+1, T+k]$, the decoder uses its previous predicted value $\hat{y}_{t^{'}-1}$ as its input until $\hat{y}_{T+k}$ is obtained. Our investigation in Section VI also shows that for predicting 10 time steps ($k = 10$) in the future using a historical data of 20 time steps ($n = 20$) provides the best prediction performance.

A. Training the Sequence-to-Sequence model

At training time, we find the best estimates for the hidden weight matrices and biases for each cell within the encoder and the decoder. In our sequence-to-sequence architecture both RNNs forming the encoder and decoder are trained jointly to minimize the loss function given by the MSE (mean squared error) of all predictions. All the parameters of this model are trained iteratively using the backpropagation algorithm, which propagates the error in the output layer through the recurrent layers. We propose a general criteria for attaining convergence at training time. Convergence is established when the observed loss for 5 consecutive epochs is less than an MSE of 9, which roughly corresponds to an average prediction error in received signal strength of 3 dBm. In order to make the models practically feasible, we limit the maximum number of training epochs to $50,000$. However, for some datasets and some models, it is possible that convergence is attained quickly. In these scenarios, we train the model for at least 1000 epochs. We note that we designed this generalized stopping criterion to make the model readily deployable rather than fine-tuning it for different datasets. We empirically observe that this generalized stopping criterion enables in achieving good prediction performance at test time (SectionVI).

In our experiments (at both training and test times), for a given signal strength measurement sample, we use a sliding window of one step to obtain X_T, thereby achieving the maximum overlap of sequences used. Additionally, we investigate three possible training schemes—i) guided, ii) unguided, and iii) curriculum, which are explained below. In the training schemes below, $y_{t'}$ refers to the actual signal strength measurement available during training time at each decoder unfolded step t'.

Guided Training: In this scheme, at each unfolded decoder step t' during training time, instead of feeding the previous predicted result $\hat{y}_{t'-1}$, we feed the actual signal strength measurement $y_{t'-1}$ as the input. This scheme aims to achieve faster convergence by guiding the model toward the nearest local minima. However, since at test time, we don't have access to the actual signal strength values at the previous time step, this scheme often suffers from poor generalizability at test time [7].

Unguided Training: In contrast to the scheme above, unguided training uses the previous predicted value $\hat{y}_{t'-1}$ as the input for the t' step of the decoder. This scheme provides the opportunity to explore the solution space better, thus increasing the generalizability of the model, often leading to better prediction performance at test time/deployment.

Curriculum Training: This scheme uses a combination of guided and unguided training to train the models. Here, we start off with guided training so that the model can make progress in the right direction initially when the model typically needs more guidance and then proceed to make it unguided so that the model can explore the solution space and produce a generalized solution. For example, we can implement this by splitting the training data into two sets comprised of 30% and 70% of the original training dataset, respectively. We then employ guided training for the first 30% data. After the model converges, unguided training is adopted for the remaining 70% of the data.

For all training schemes, once the initial training is complete, we re-train the model on random sequences of length n sampled from the training data amounting to 10% of the actual training dataset while ensuring adherence to the convergence criteria. This exercise improves the model's exploration of the solution space and its generalizability. These measures (unguided/curriculum training and training on random sequences) and the incorporation of $L2$ regularization reduce overfitting the model to training data. We will see in Section VI-C that unguided training yields the best results for the datasets used in the paper. Therefore, the training scheme used in our final evaluation is unguided training.

V. Datasets and data preprocessing

To demonstrate the widespread applicability of our model, we consider multiple received signal strength measurement datasets collected at the end hosts for five different networks—4G LTE, WiFi, an industrial network operating at 5.8 GHz channel gain within a factory environment, Zigbee and WiMAX. The 4G LTE network measurements and the WiFi measurements used in this paper are collected by us, while the other datasets are publicly available [1], [5]. We next describe the network settings, characteristics, and preprocessing steps undertaken for each dataset.

1) 4G LTE Measurements: We collect Reference Signal Received Power (RSRP) measurements using a Motorola G5 smartphone over T-Mobile and AT&T 4G LTE networks in vehicular and pedestrian mobility scenarios. The vehicular and pedestrian mobility traces are approximately 50 and 20 minutes in duration and are collected at a granularity of 1 second. Prior work [12] has demonstrated the need for wireless channel prediction on the seconds' timescale for improved video streaming over cellular networks.

2) WiFi Measurements: We collect two datasets containing received signal strength indicator (RSSI) using a Motorola G5 smartphone on a campus WiFi network at sampling rates of 1 and 2 seconds respectively. Each measurement is carried out for approximately 50 minutes amidst pedestrian mobility (indoor and outdoor). Prior work [12] has demonstrated the need for wireless channel prediction on the seconds' timescale for designing block-based bit rate adaptation algorithms for WiFi networks.

3) Industrial Network Measurements: This dataset contains wireless channel measurements collected over a time-variant and frequency-variant 5.8 GHz channel gain within a factory environment in the presence of pedestrian mobility [1]. We consider three such datasets, each collected using a stationary pair of antennas separated by a distance of 3.1m, 10.0m and 20.4m respectively. Each dataset contains approximately 1000 samples.

4) Zigbee Measurements: We consider signal strength measurements collected over a wireless sensor network operating under Zigbee containing around 2000 samples [5]. The datasets were collected using two sensor nodes communicating with each other over fixed distances of 10m and 15m for a power level of 31 (0 dBm). We fill potential missing values indicative of packet loss with random signal strength values obtained between the smallest recorded RSSI and 10 units below that.

5) WiMAX Measurements: We consider three separate datasets containng RSSI measurements collected over a (802.16e) WiMAX network [12], one vehicular and two pedestrian (one indoor and the other outdoor) mobility datasets. In each of the datasets, RSSI measurements are recorded at the granularity of one second. The indoor pedestrian, outdoor pedestrian and the vehicular mobility traces are approximately 10, 38, and 26 minutes in duration, respectively.

VI. Experimental Evaluation

In this section, we present experimental results that demonstrate the widespread applicability and robustness of the deep learning model. The main metric used for evaluation is the root mean squared error (RMSE) that captures the error of the absolute prediction. Let y_{ij} be the i^{th} test sample for the j^{th} prediction step where $j \in [1, k]$, and $\hat{y}_{ij}$ be the predicted value of y_{ij} and h the number of test samples. The RMSE is given by Equation 1.

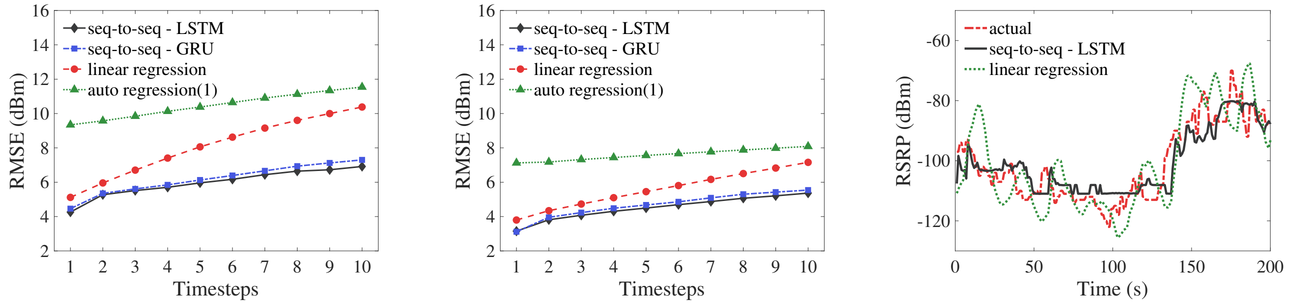
$$RMSE_j = \sqrt{\frac{\sum_{i=1}^{h} \left(\hat{y}_{ij} - y_{ij}\right)^2}{h}} \tag{1}$$

We compare the performance of our model with respect to two baselines—linear regression and auto regression(1). In our experiments, the linear regression baseline considers a history of previous 20 samples to predict the future. We consider the auto regression(1) baseline because prior work related to channel modeling has been mainly focused on designing Markov chains to capture the underlying channel correlation. The default parameters used for our models are—history window size = 20 , number of future prediction steps = 10, number of stacked layers = 2, and number of hidden units per cell = 100.

A. RMSE Results for 4G LTE

In this subsection, we discuss RMSE results in detail for the 4G LTE network to demonstrate the superior performance of our model. We then discuss its performance on other networks. Figures 2 and 3 show the performance of the deep learning model and the baseline approaches for 4G LTE networks (T-Mobile and AT&T) for pedestrian and vehicular mobility scenarios. We observe from Figures 2(a) and 2(b) that the deep learning model with either LSTM or GRU cells significantly outperforms the linear regression and auto-regression(1) models in both mobile settings. We observe that in comparison to linear regression and auto regression(1), the RMSE values for the deep model increase slowly as the number of time steps increases. This means that the our model is able to predict further into the future considerably better than the baseline approaches. Additionally, based on these results and those from all networks, we observe that there is no clear winner between the deep learning models, with both variants outperforming one another depending on the network.

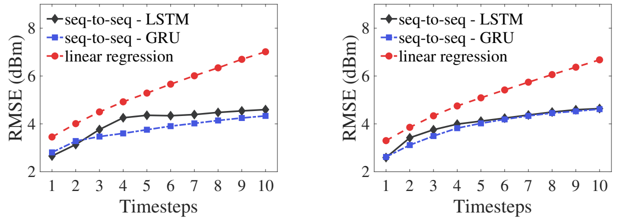
We also observe from our experiments on different networks that auto-regression(1) performs the worst, with its first step prediction being significantly worse in comparison to the other approaches. Figures 2(a) and 2(b) also show that the prediction performance of auto-regression(1) gradually deteriorates (or almost remains constant) over time. These results suggest that making future predictions based solely on the signal strength measurement obtained in the previous time step is insufficient and not useful. As auto-regression(1) fails to successfully capture the temporal correlation of the wireless channel and provides poor predictive performance, in the remaining figures, we only plot the performance results of our deep learning



(a) Pedestrian Mobility (b) Vehicular Mobility (c) Pedestrian Mobility - 5 step prediction

Fig. 2: 4G LTE T-Mobile experiments

models and linear regression. Figure 2(c) shows a qualitative comparison of the actual vs. the prediction results of the LSTM model with respect to linear regression for the pedestrian mobility scenario for the T-Mobile network. The figure depicts a prediction timeframe of 200 seconds considering the 5^{th} time step prediction for each model variant. We observe that the performance of linear regression is significantly poor compared to the deep learning model. This correlates with RMSE variations shown in Figure 2(a). Similarly, from Figure 3, we observe that the deep learning models outperform linear regression for the 4G LTE AT&T network.

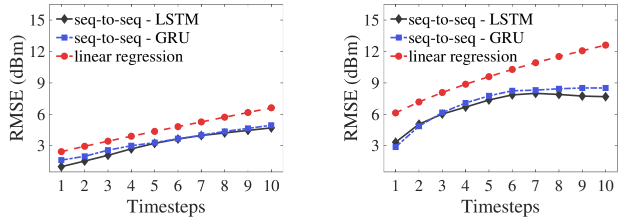


(a) Pedestrian Mobility (b) Vehicular Mobility

Fig. 3: 4G LTE AT&T experiments

B. RMSE Results for Other Networks

In this subsection, we present performance results comparing the deep model with linear regression for all other networks (i.e., WiFi, industrial network operating in the 5.8 GHz range, Zigbee and WiMAX) described in Section V. Figures 4(a), and 4(b) outline the prediction performance for different sampling rates (1s, and 2s) in a WiFi network for a pedestrian mobility scenario. Once again, we observe that the deep learning model outperforms linear regression in all cases. As all measurements are undertaken in a pedestrian mobility scenario, there is little variation in physical position and mobility between consecutive samples at the lower sampling rate. This makes it an easier prediction task, thus resulting in linear regression and the deep learning model having comparable performance (Figure 4(a)).



(a) Sampling Rate = 1 s (b) Sampling Rate = 2 s

Fig. 4: WiFi experiments

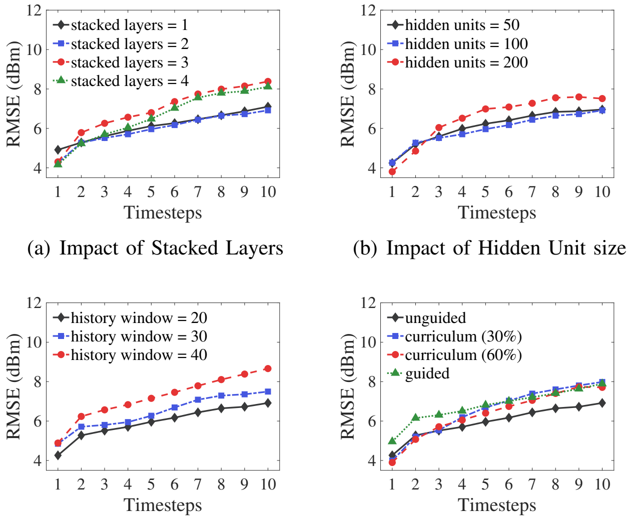
Network Type		Seq-to-seq LSTM (%)	Seq-to-seq GRU (%)
Industrial Network	Distance = 3.1m	18.8	15.0
	Distance = 10m	27.1	30.9
	Distance = 20.4m	39.7	24.9
Zigbee	Distance = 10m	10.5	14.0
	Distance = 15m	5.90	6.80
WiMAX	Pedestrian (Outdoor)	4.50	3.10
	Pedestrian (Indoor)	33.1	25.6
	Vehicular	38.0	45.3

TABLE I: Predictive performance of other networks. Average performance increase (%) with respect to linear regression.

In the interest of space, in Table I, we present the average predictive performance improvement (over 10 future steps) for both LSTM and GRU based models with respect to linear regression for the remaining networks. While it is clear that the deep model outperforms the base case, it is also evident that there is no clear winner between the LSTM or GRU versions. From our experiments, we also observe that for the stationary industrial setting, the performance gap increases with distance, primarily to the presence of higher number of obstacles that result in increased signal strength variation. While deep learning beats linear regression in all cases, we note that the performance gap is less for the Zigbee and WiMAX outdoor pedestrian mobility experiments. We hypothesize the seasonal nature of the channel variations and low overall fluctuations in the channel in these datasets as the main reason for increased performance of linear regression.

C. Discussion on Design Decisions

In this subsection, we discuss hyper-parameter tuning and the rationale behind key design decisions in training. Here, we show all findings for the LSTM variant of our model for the 4G LTE T-Mobile pedestrian mobility dataset. However, we note that these insights hold true for the GRU version as well as for other networks.



(a) Impact of Stacked Layers

(b) Impact of Hidden Unit size

(c) Impact of History Window size

(d) Impact of Training method

Fig. 5: Impact of parametric changes and training method on performance

1) Hyper-parameter tuning: Figures 5(a), 5(b), and 5(c) show the predictive performance of our model for variations in parameters such as number of stacked layers, number of hidden units in a cell, and the history window size, respectively. We experimentally validate that number of stacked layers = 2, number of hidden units in a cell = 100 and history window size = 20 in general provides the best performance for our model. We hypothesize from these results that overly complex models with larger number of hidden units per cell and greater depth may be attempting to learn additional complexity in the channel variations than what is present, thus generalizing poorly to test data. Similarly, Figure 5(c) shows that considering the past 20 signal strength measurements is sufficient for forecasting future channel variations. Interestingly, it illustrates that "more" history does not always carry more information about the channel.

2) Discussion on training: We next discuss the rationale behind adopting a particular training methodology for our model. Figure 5(d) shows the prediction performance of the deep learning model for four training methodologies—unguided learning, curriculum learning (with first 30% of data as guided), curriculum learning (with first 60% of data as guided), and guided learning. Even though unguided training usually requires more epochs to converge, we observe that it provides the best performance at test time for all network settings. This is due to the larger solution space explored by this method in comparison to the other methods.

VII. Conclusion

In this paper, we investigated the wireless channel quality prediction problem in wireless networks. We developed an encoder-decoder based sequence-to-sequence deep learning model that takes prior channel quality (i.e., received signal strength) into account to predict future signal strength variations. We compared the performance of the deep learning model with the auto-regression(1) and linear regression baselines and observed that our model significantly outperforms these baseline models for different network settings. In future, we intend to study the robustness of our model by investigating the performance of the trained model on previously unseen data. We also plan to explore the channel quality prediction problem from a more interpretable graphical modeling perspective.

References

[1] D. Block, N. H. Fliedner, and U. Meier. CRAWDAD dataset init/factory (v. 2016-06-13). Downloaded from https://crawdad.org/init/factory/20160613/factory1-channel-gain, June 2016. traceset: factory1-channel-gain.

[2] J. M. Bruno, Y. Ephraim, B. L. Mark, and Z. Tian. Spectrum sensing using markovian models. *Handbook of Cognitive Radio*, pages 1–30, 2017.

[3] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer. A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques. *IEEE Communications Surveys & Tutorials*, 19(3):1790–1821, 2017.

[4] P. Fazio, M. Tropea, C. Sottile, and A. Lupia. Vehicular networking and channel modeling: a new markovian approach. In *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*, pages 702–707. IEEE, 2015.

[5] S. Fu and Y. Zhang. CRAWDAD dataset due/packet-delivery (v. 2015-04-01). Downloaded from https://crawdad.org/due/packet-delivery/20150401, Apr. 2015.

[6] E. N. Gilbert. Capacity of a burst-noise channel. *Bell system technical journal*, 39(5):1253–1265, 1960.

[7] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[8] L. Liu, B. Yin, S. Zhang, X. Cao, and Y. Cheng. Deep learning meets wireless network optimization: Identify critical links. *IEEE Transactions on Network Science and Engineering*, 2018.

[9] T. Liu and A. E. Cerpa. Temporal adaptive link quality prediction with online learning. *ACM Transactions on Sensor Networks (TOSN)*, 10(3):46, 2014.

[10] Q. Mao, F. Hu, and Q. Hao. Deep learning for intelligent wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2018.

[11] P. Sadeghi, R. A. Kennedy, P. B. Rapajic, and R. Shams. Finite-state markov modeling of fading channels-a survey of principles and applications. *IEEE Signal Processing Magazine*, 25(5), 2008.

[12] A. Seetharam, J. Kurose, and D. Goeckel. A markovian model for coarse-timescale channel variation in wireless networks. *IEEE Trans. Vehicular Technology*, 65(3):1701–1710, 2016.

[13] H. S. Wang and P.-C. Chang. On verifying the first-order markovian assumption for a rayleigh fading channel model. *IEEE Transactions on Vehicular Technology*, 45(2):353–357, 1996.

[14] 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 *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2017.

[15] J. Wang, X. Zhang, Q. Gao, H. Yue, and H. Wang. Device-free wireless localization and activity recognition: A deep learning approach. *IEEE Transactions on Vehicular Technology*, 66(7):6258–6267, 2017.

[16] Y. Yu, T. Wang, and S. C. Liew. Deep-reinforcement learning multiple access for heterogeneous wireless networks. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2018.