This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2949954, IEEE Transactions on Vehicular Technology

DeepChannel: Wireless Channel Quality Prediction using Deep Learning

Adita Kulkarni, Anand Seetharam, Arti Ramesh, J. Dinal Herath
Department of Computer Science, SUNY Binghamton, USA
akulka17@binghamton.edu, aseethar@binghamton.edu, artir@binghamton.edu, jherath1@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 design DeepChannel, 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 DeepChannel; 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, DeepChannel 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, mean absolute error and relative error of future predictions) of DeepChannel with respect to two baselines—*i)* linear regression, and *ii)* ARIMA for multiple networks and communication standards. In particular, we consider 4G LTE, WiFi, WiMAX, an industrial network operating in the 5.8 GHz range, and Zigbee networks operating under varying levels of user mobility and observe that DeepChannel provides significantly superior performance. Finally, we provide a detailed discussion of the key design decisions including insights into hyper-parameter tuning and the applicability of our model in other networking scenarios.**

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 [1]. Multiple foreseeable applications motivate this research such as better scheduling and improved video streaming over 4G networks [2], [3], bit rate adaptation for improved performance in WiFi networks [4], [5], and energy efficient and bulk transfer of data in sensor networks [6], [7].

Most prior research in this domain has been focused on designing Markovian models that capture the impact of wireless channel characteristics such as multi path fading, shadowing and path loss on the received signal strength [8], [9]. 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], [11].

In recent years, a particular class of machine learning models (i.e., deep sequence-to-sequence models) have been shown to be well-suited for a variety of time series forecasting and prediction problems where the input data is correlated and varies randomly. As wireless channel quality also exhibits this property, in this paper, we explore deep learning models to address the wireless channel quality prediction problem. *We primarily investigate the received signal strength metric to study wireless channel quality variation, though we also explore other applications of our model.*

Specifically, we design *DeepChannel*, an encoder-decoder based sequence-to-sequence deep learning model, which is capable of predicting variations in wireless signal strength. With DeepChannel, our goal is to design a deep learning model that can effectively capture and predict channel quality variations in different network settings and mobility scenarios, and works across communication standards and sampling rates. DeepChannel comprises 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 signal strength variations. We develop two variants 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 DeepChannel, we conduct experiments on received signal strength data collected over different kinds of networks including 4G LTE, WiFi, WiMAX, Zigbee 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 (e.g., 0.2s, 1s, 2s) and in pedestrian and vehicular mobile scenarios. We compare the performance of DeepChannel with linear regression and ARIMA, and show that DeepChannel outperforms the baselines in all scenarios.

Interestingly, we observe that DeepChannel provides higher performance gains for network settings with higher signal strength variations and less seasonality, which demonstrates the superiority of the model.

To derive more insights into the functionality of DeepChannel, we investigate the impact of parameters such as sequence length, number of hidden layers and the type of training methodology used on prediction performance. While the optimal parameter configuration for DeepChannel varies with the dataset in consideration, we observe that a model consisting of 1 or 2 layers with 50 to 200 units in each layer provides the best performance, depending on the dataset. Our experiments demonstrate that superior performance for the wireless signal strength prediction problem can be achieved by experimenting with a limited number of parameter configurations.

Interestingly, we observe that sequence lengths of size 20 capture most of the useful information in the data and sequences of greater length do not improve performance. We hypothesize the simplicity of our datasets to be main reason as to why smaller sequence lengths are sufficient. Additionally, we also observe that a simple unguided learning strategy, which uses the model's predictions in the previous step at training time achieves better predictive performance than a more complex training methodology such as curriculum learning that carefully balances between using actual data and model's predictions in the previous step at training time. The unguided strategy results in a greater exploration of the solution space, and thus generalizes better to test data, achieving better prediction performance. We also provide preliminary results on the applicability of our model in other networking scenarios. We then explore avenues for future research by discussing the performance of a trained model on previously unseen data.

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 quality.

The networking literature is rife with Markovian models for wireless signal strength prediction. Sadeghi et al. [8] and Bui et al. [9] 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 prediction of slow channel processes in LTE networks [13], spectrum sensing utilizing a hidden bivariate Markov chain [14] and modeling channel variations for vehicular networks [15]. While Markovian models offer insight into wireless channel variations, prior work by Wang et al. [16] 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 [17], [18], extracting useful features of the wireless channel [19], [20], identifying critical links [21] and spatio-temporal modeling and prediction in cellular networks [22]. For example, Mekki et al. [23] combine Kalman filters with expectation maximization to accurately predict channel gains.

Recent years has seen an increased use of deep learning models to solve various problems in wireless communications [10], [24]–[27]. Deep learning based resource allocation for 5G networks and automatic modulation recognition in cognitive radio networks is performed in [28] and [29], respectively. Similarly, the authors in [30], [31] demonstrate the benefits of designing deep learning models for addressing hybrid precoding and beam forming issues in MIMO systems, respectively. Some other examples of designing deep learning models for wireless communications include caching and interference alignment [32], device-free wireless localization using shadowing effects [33] and spectrum sharing in heterogeneous wireless networks [34]. A comprehensive survey on the application of deep learning models for traffic control and automatic network configuration and management is provided in [35], while [36] outlines the challenges that still need to be overcome to enable the seamless adoption of deep learning models for solving network traffic control. Similarly, Mao et al. [11] identify many opportunities for the use of deep learning in wireless networks and emphasize the capability of deep learning models.

In contrast to prior work on wireless signal strength prediction and deep learning models in wireless communications, in this paper, we design a deep sequence-to-sequence model, specifically tailored for received signal strength prediction. Additionally, instead of resorting to simulation as has been done in most prior research, we demonstrate the applicability of our model for a variety of network settings and communication standards via exhaustive experiments on real-world measurement data.

III. Problem Statement and Motivation

Several factors such as the environment, user mobility, and communication technology cause sudden variations in the received signal strength, thus posing challenges in developing a generalized framework for this prediction task. *In this work, our goal is to design a predictive model, which is capable of accurately predicting received signal strength variations irrespective of mobility pattern, communication standard, and sampling rate.* This problem can be modeled as a classic time series prediction problem, where the goal at time T is to predict signal strength variations for k steps into the future (i.e., $\hat{Y}_T = [\hat{y}_{T+1}, \hat{y}_{T+2}, ..., \hat{y}_{T+(k-1)}, \hat{y}_{T+k}]$) based on past signal strength measurement values in a window size of n (i.e., $X_T = [x_{T-n}, x_{T-(n-1)}, ..., x_{T-1}, x_T]$). We note that $\hat{Y}_T$ are the predictions for the actual future signal strength values Y_T.

With the proliferation of wireless devices, accurately predicting the quality of the wireless channel has become an extremely important problem and has a number of applications related to providing Quality of Service (QoS) guarantees, scheduling, and reducing network energy consumption [10],

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2949954, IEEE Transactions on Vehicular Technology

[37]. We provide a few examples that rely on accurate wireless channel prediction.

- One of the most important applications of wireless channel quality prediction is managing the QoS of multiple different videos being streamed over a cellular network [3], [37]. In such scenarios, the ability to predict wireless channel variations on a per-client basis would enable the base station to effectively allocate communication slots, thereby maintaining QoS guarantees.
- Bit rate adaptation in WiFi networks has been widely used to improve application-level performance [4]. One of the key components necessary for performing anticipatory bit rate control is accurately predicting channel quality at the milliseconds' and seconds' timescale. For example, a block-based bit rate adaptation scheme [5] requires coarse timescale channel predictions to predict channel fluctuations from one block to the next (a block can take 1-2 seconds to be transmitted), while predictions at a finer time granularity help capture channel variations within a block.
- Predicting wireless channel quality could also help determine optimal communication paths and increase the rate of successful packet transmission in wireless sensor and mesh networks [7], [38]. Since IoT based networks have devices with limited battery life, reducing the number of retransmissions and controlling energy consumption would greatly benefit performance.
- Wireless networks in industrial settings are expected to maintain performance in environments harsher than commercial settings [39]. Maintaining QoS guarantees in these settings requires developing smart sensing applications that rely heavily on channel quality predictions [40], [41].

IV. DeepChannel: Sequence-to-Sequence Model

In this section, we describe DeepChannel, an encoder-decoder based sequence-to-sequence deep learning model for solving the wireless received signal strength prediction problem. Before diving into the details of DeepChannel, we first discuss the limitations of traditional model-based approaches and the applicability of deep sequence-to-sequence models for this prediction problem.

A. Why Sequence-to-Sequence Deep Learning Model?

Traditional model-based approaches (e.g., Markovian models) are parsimonious in nature, make simplifying assumptions, rely on few network parameters and require limited amount of previous history to make decisions. While simple model-based approaches are invaluable when computational power and data are a premium, they usually lack generality, and may not necessarily work well in diverse real-world settings. 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 are capable of providing superior prediction performance in diverse mobile wireless real-world networks.

To this end, we explore deep sequence-to-sequence models that are ideally suited for problems requiring mapping input sequences to output sequences. Deep sequence-to-sequence models have been extensively used for tasks such as video captioning [42] and natural language translation [43], while recent work [44] has also demonstrated their applicability for forecasting and prediction purposes where the objective is to predict the future based on past time series data.

Deep sequence-to-sequence models possess the ability to predict an entire sequence of data points based on past data, thus being able to predict further into the future. Additionally, deep models are best suited to scenarios where dependencies among data points are harder to discern exactly using model-based approaches, but can be learned automatically by the model by training on vast amounts of data. The deep architecture allows for elegantly learning non-linear dependencies as the encoded signal passes through the different hidden layers. As we will see, DeepChannel specifically employs a recurrent architecture that is best suited to time series data with LSTM/GRU cells that retain "memory" on dependencies that have a positive impact on the prediction. As the received signal strength over a wireless channel in a real-world setting varies randomly and has the property to be correlated for long time periods, this makes it ideally suited for designing deep models specifically tailored for this prediction task.

B. DeepChannel

Our model DeepChannel has two main components—an *encoder* and a *decoder*. Figure 1 provides an overview of DeepChannel. The encoder receives the past signal strength measurements X and produces a context vector C (i.e., the encoded state) that summarizes the input sequence X. The decoder receives this as an input and in turn produces $\hat{Y}$, 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., $n \neq k$) [45]. Both the encoder and decoder use RNN as the underlying neural network architecture that is particularly suited for sequence-to-sequence modeling. The encoder and decoder are thus organized as 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.

We next provide a brief overview of RNN. Figures 2(a) and 2(b) show the folded and unfolded versions of a single layer RNN, respectively, with the unfolded structure of the RNN showing the calculation done at each time step t. In these figures $X_t = [x_{t-2}, x_{t-1}, x_t]$ and $\hat{Y}_t = [\hat{y}_{t-2}, \hat{y}_{t-1}, \hat{y}_t]$ are the input and corresponding output vector respectively, h_t is the hidden layer, and W_{xh}, W_{hh}, and W_{hy} are the weight matrices. The hidden layer h_t serves as memory and is calculated using the previous hidden state h_{t-1} and the input x_t. At each time step t, the hidden state of the RNN is given by,

$$h_t = \phi(h_{t-1}, x_t) \quad (1)$$

where, ϕ is any non-linear activation function and $1 \leq t \leq n$. The weight matrices are used for transforming the input to the output via the hidden layers. We refer the reader to Goodfellow et al. [45] for additional details on updating h_t and the weight matrices.

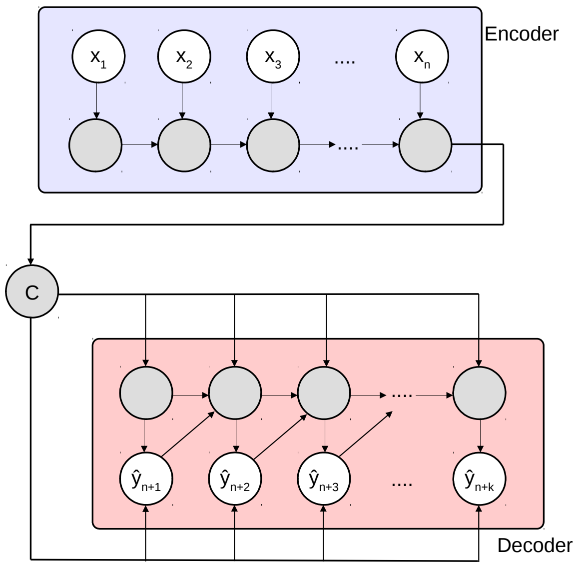
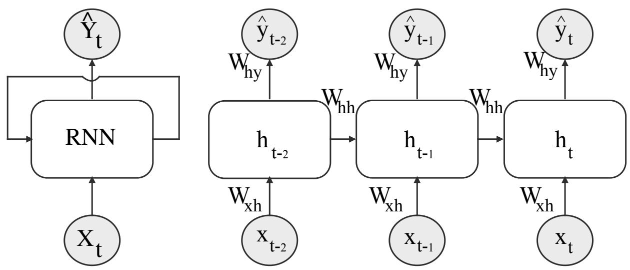


Fig. 1: Encoder-decoder based sequence-to-sequence architecture. In our model the "basic cell" is either LSTM or GRU.



(a) RNN folded (b) RNN unfolded

Fig. 2: Illustration of RNN architecture.

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 [46]. To circumvent this problem, LSTM and GRU cells were proposed; they create paths through time with derivatives that do not vanish or explode [45] by incorporating the ability to "forget". Therefore, we consider two versions of DeepChannel, where the basic building block can be either an LSTM or GRU cell.

The LSTM and GRU cells are both based on the same underlying idea and primarily differ in the number of gates and their interconnections. While, the LSTM cell consists of three gates namely, the input gate, the output gate, and the forget gate that lets it handle long-term dependencies, the GRU cell consists of two gates, a reset gate that combines the current input with previous memory and an update gate

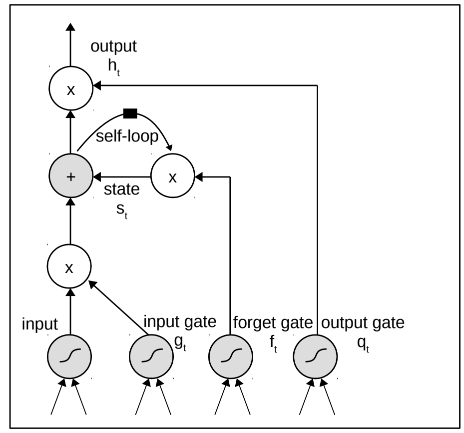


Fig. 3: Illustration of LSTM cell architecture. g_t, f_t and q_t are the input, forget, and output gates, respectively.

that determines the percentage of previous state to remember. Both LSTM and GRU based models have been shown to be effective in a number of prediction tasks and it is impossible to determine theoretically which one is likely to be more suited for a particular problem [45].

We next describe the internal architecture of an LSTM cell. The working of a GRU cell is similar and we refer the reader to [45], [47], [48] for more details. Figure 3 shows the basic structure of a single LSTM cell. LSTM recurrent networks have an LSTM cell that has an internal recurrence (referred to as a self-loop in Figure 3). Note that this is in addition to the outer recurrence of the RNN. Each cell has the same inputs and outputs as a node in an ordinary recurrent network, but also has more parameters and a system of gating units that controls the flow of information. The most important component is the state unit s_t^i that captures the internal state of the i^{th} LSTM cell, which has a linear self-loop and a self-loop weight, which is given by,

$$s_t^i = f_t^i s_{t-1}^i + g_t^i \, \sigma\Big(b_i + \sum_j U^{(i,j)} x_t^j + \sum_j W^{(i,j)} h_{t-1}^j\Big) \tag{2}$$

where b_i, U, and W denote the bias, input weights, and recurrent weights, respectively. The self-loop weight is controlled by a forget gate unit f_t^i, which controls the dependence of the current state s_t^i on historical states s_{t-1}^i. f_t^i is set to a value between 0 and 1 via a sigmoid unit as shown below.

$$f_t^i = \sigma\Big(b_f^i + \sum_j U_f^{(i,j)} h_{t-1}^j + \sum_j W_f^{(i,j)} x_t^j\Big) \tag{3}$$

where i refers to the i^{th} LSTM cell, x_t is the current input vector and h_t is the current hidden layer vector. b_f, U_f, and W_f refer to the bias, input weights, and recurrent weights for the forget gate. j denotes the cells feeding into i and h_{t-1} corresponds to their output.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2949954, IEEE Transactions on Vehicular Technology

The external gate unit is similar to the forget gate and is given by,

$$g_t^i = \sigma\Big(b_g^i + \sum_j U_g^{(i,j)} x_t^j + \sum_j W^{(i,j)} h_{t-1}^j\Big) \tag{4}$$

Finally, the output of the LSTM cell h_t^i and the output gate q_t^i is given by,

$$\begin{aligned} h_t^i &= tanh(s_t^i) q_t^i \\ q_t^i &= \sigma\Big(b_o^i + \sum_j U_o^{i,j} x_t^j + \sum_j W_o^{i,j} h_{t-1}^j\Big) \end{aligned} \tag{5}$$

As mentioned earlier, in DeepChannel, both the *encoder* and the *decoder* operate as a deep RNN with either LSTM or GRU cells at each layer. While the optimal parameter configuration for DeepChannel varies with the dataset in consideration, we observe that a model consisting of 1 or 2 layers with 50 to 200 units in each layer provides the best performance, depending on the dataset. Our experiments demonstrate that superior performance for the wireless signal strength prediction problem can be achieved by experimenting with a limited number of parameter configurations. We present in-depth insight into the rationale behind choosing these parameters in Section VII-D.

V. Implementation Details

In this section, we discuss the implementation details of DeepChannel and the training methodology. We split the datasets into two parts—the first part consisting of 80% of the data is used for the training and the remaining 20% is used for testing. We train our models on a shared high performance computing cluster available at our University. Using this cluster, we are able to execute 10 to 15 experiments in parallel. Each experiment is allocated 4 cores and 4 GB of RAM. For the datasets considered in this work, for a particular configuration of parameters, training the deep models (i.e., a single experiments) can take in the order of 6 - 12 hours, which is typical of deep learning models. In comparison to training, the testing phase of the model takes only a few minutes for each experiment.

Due to the high computational requirement of deep models, we investigate the parameter space extensively over a period of three to four months before empirically deciding the 'best' parameters of the model. We note that determining the optimal parameter values theoretically for a particular dataset is still an open research question, and so we determine the parameters empirically. We experiment with different number of stacked layers, different numbers of hidden units in each layer as well as the lengths of the input and output sequences. We tune the model parameters for each dataset where we vary the number of stacked layers between 1 and 2, the number of hidden units between 50 and 200, and the learning rate from 0.01 to 0.0001. Our investigation shows that for predicting 10 time steps ($k = 10$) into the future using a historical data of 20 time steps ($n = 20$) provides the best prediction performance.

A. Training DeepChannel

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 DeepChannel, 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 parameters are trained iteratively using the backpropagation algorithm, which propagates the error in the output layer through the recurrent layers. We train DeepChannel for 1000 to 20000 epochs depending on the dataset. We select the number of epochs empirically by balancing the tradeoff between performance and training time.

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, 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 training scheme often suffers from poor generalizability at test time [45].

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.

We incorporate $L2$ regularization to reduce overfitting the model to training data. We will see in Section VII-D 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.

VI. Datasets and data preprocessing

To demonstrate the widespread applicability of the proposed model, we consider multiple received signal strength measurement datasets collected at the end hosts for five different

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2949954, IEEE Transactions on Vehicular Technology

networks—4G LTE, WiFi, WiMAX, an industrial network operating at 5.8 GHz channel gain within a factory environment and a wireless sensor network operating under 802.15.4 (Zigbee). The 4G LTE network measurements and majority of the WiFi measurements used in this paper are collected by us, while the other datasets are publicly available [49]–[51]. We note that for each type of network, we conduct experiments on multiple datasets under varying levels of user mobility and different sampling rates. We next describe the network settings, characteristics, and preprocessing steps undertaken for each dataset.

A. 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, respectively. RSRP measurements for both datasets are collected at the granularity of one second. Prior work [37] has demonstrated the utility of collecting signal strength measurements at the seconds' granularity for video streaming and channel modeling purposes.

B. WiFi Measurements

We use three WiFi traces collected in a mobile environment for different sampling rates. 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. These traces contain measurements for pedestrian mobility in both indoor and outdoor environments. The third dataset contains RSSI measurements collected over an 802.11n WiFi network using a mobile robot acting as an access point in a semi-outdoor environment at the Royal Institute of Technology (KTH), Stockholm, Sweden [51]. The measurements are collected at the granularity of 0.2 seconds for a duration of approximately 20 minutes. Prior work [5] has shown that both long term (order of seconds) and short term (order of milliseconds) predictions are beneficial for performing rate adaptation over a WiFi network.

C. WiMAX Measurements

We also consider RSSI measurements collected over a (802.16e) WiMAX network deployed in WINLAB at Rutgers University [12]. We consider three separate datasets, 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. As mentioned earlier, prior work [12], [37] has demonstrated the utility of seconds' timescale channel prediction over a cellular network.

D. 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. The datasets were collected by Block et al. [49] at the SmartFactoryOWL lab at the Institute Industrial IT (inIT), Ostwestfalen-Lippe University of Applied Sciences, Germany. 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. Because of the factory environment, the line of site between the antennas could have been obstructed due to industrial machinery and tools in addition to pedestrians. The antennas were aligned vertically and the obtained dataset was already normalized to remove errors from cables and adapters.

E. Zigbee Measurements

We consider signal strength measurements collected over a wireless sensor network operating under Zigbee by researchers at University of Duisburg-Essen, Germany and Norwegian University of Science and Technology, Norway [50]. The testbed consists of two sensor nodes communicating with each other over fixed distances ranging from 10m to 35m in an indoor environment. For each distance, the trace contains received signal strength data for successful transmissions at multiple transmission power levels between 3 and 31. These power levels correspond to a power range between -25 dBm to 0 dBm respectively. We only consider power level 31 because lower power levels have a larger number of missing RSSI values due to lost packets. As missing values indicate packet loss, we fill missing values at power level 31 with random signal strength values obtained between the smallest recorded RSSI and 10 units below that. We consider two datasets each containing around 2000 samples for distances 10m and 15m, respectively.

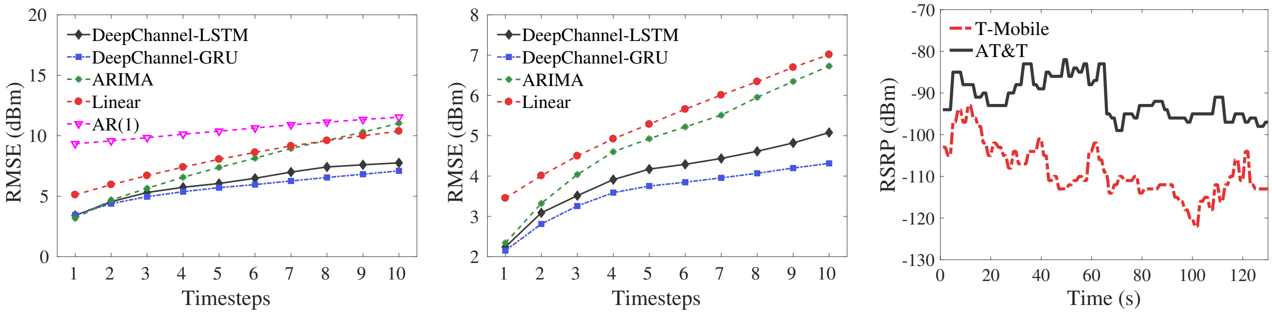
VII. Experimental Evaluation

In this section, we present experimental results that demonstrate the widespread applicability and robustness of DeepChannel. The main metrics used for evaluation are the root mean squared error (RMSE), mean absolute error (MAE) and the Relative Error (RE). RMSE and MAE capture the error in the absolute prediction, while RE captures the fraction of the error in the prediction with respect to actual channel variation. 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 m the number of test samples. The RMSE, MAE and RE are given by Equations 6, 7 and 8 respectively.

$$RMSE_j = \sqrt{\frac{\sum_{i=1}^{m} \left(\hat{y}_{ij} - y_{ij}\right)^2}{m}} \tag{6}$$

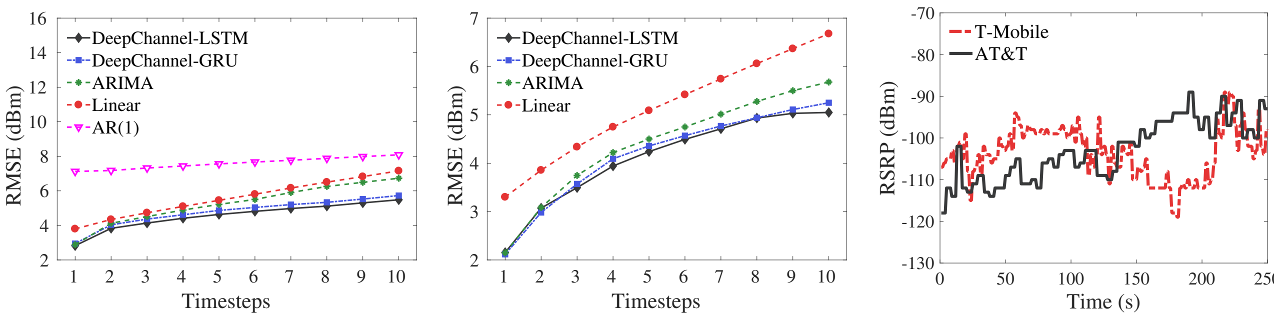
$$MAE_j = \frac{\sum_{i=1}^{m} |\hat{y}_{ij} - y_{ij}|}{m} \tag{7}$$

$$RE_j = \frac{\sum_{i=1}^{m} \frac{|\hat{y}_{ij} - y_{ij}|}{y_{ij}}}{m} \tag{8}$$



(a) T-Mobile (b) AT&T (c) Signal strength variations

Fig. 4: 4G LTE: Pedestrian mobility.



(a) T-Mobile (b) AT&T (c) Signal strength variations

Fig. 5: 4G LTE: Vehicular mobility.

We compare the performance of DeepChannel with respect to three baselines—linear regression, Auto Regression(1) and ARIMA(p, d, q). Similar to DeepChannel, the baselines also consider a history of previous 20 samples to predict 10 steps into the future.

1) Linear regression is a statistical model that fits the best line to the input data.
2) Auto Regression(1) or AR(1) is a simple model that only considers the previous value to predict the future. We consider the AR(1) baseline because prior work related to channel modeling has been mainly focused on designing Markov chains to capture the underlying channel correlation.
3) ARIMA(p, d, q) is a statistical model that has three components—an autoregressive term (AR), a differencing term (I) and a moving average term (MA), which are specified by p, d and q respectively. p represents the number of past values that are used for predicting the future, d represents the degree of differencing (i.e., the number of times the differencing operation is performed to make a series stationary), and q represents the number of error terms taken into consideration. We use the Auto-ARIMA toolkit[1] in python in our experiments. It selects the optimal combination for the input data after searching through a combination of the parameters p, d, and q. It is clear that ARIMA is a far more sophisticated baseline than AR(1) and is thus the main baseline for comparison purposes.

[1] https://pypi.org/project/pyramid-arima/

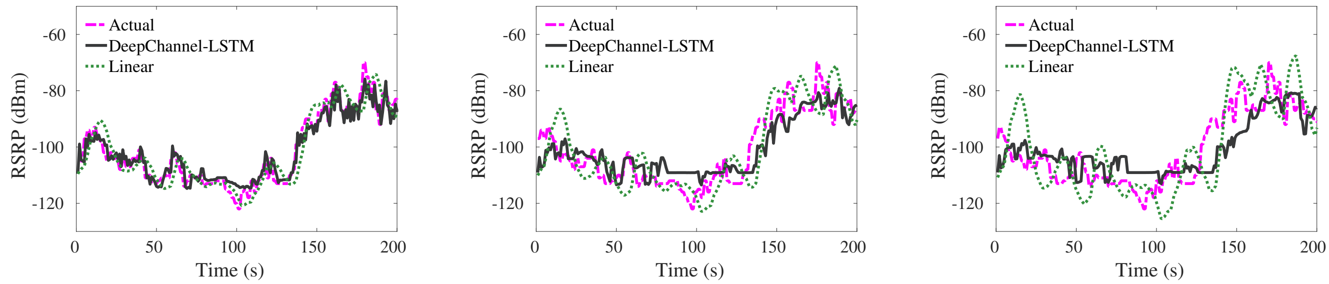
A. *RMSE Results for 4G LTE*

In this subsection, we discuss RMSE results for the 4G LTE network to demonstrate the superior performance of DeepChannel. As the performance results consist of multiple similar looking graphs, we first present results for the 4G LTE network and then discuss other networks.

Figures 4 and 5 show the performance of DeepChannel and the baseline approaches for 4G LTE networks (T-Mobile and AT&T) for pedestrian and vehicular mobility scenarios. We observe from the figures that the LSTM and GRU variants of DeepChannel significantly outperforms the linear regression, AR(1) and ARIMA models in both mobile settings. We observe that in comparison to linear regression and ARIMA, the RMSE values for DeepChannel increase slowly as the number of time steps increases. This means that DeepChannel 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 two versions of DeepChannel, with both variants outperforming one another depending on the network.

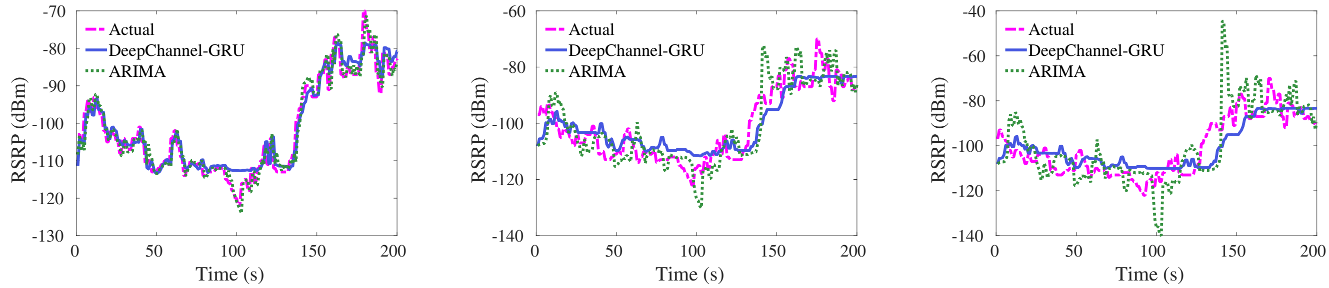
We observe from our experiments on different networks that AR(1) performs the worst, with its first step prediction being significantly worse in comparison to the other approaches. Figures 4(a) and 5(a) also show that the prediction performance of AR(1) gradually deteriorates (or almost remains constant) 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 AR(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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2949954, IEEE Transactions on Vehicular Technology



(a) LSTM Prediction - 1 step (b) LSTM Prediction - 5 step (c) LSTM Prediction - 10 step

Fig. 6: 4G LTE: Comparison of real and predicted values for pedestrian mobility (LSTM).



(a) GRU Prediction - 1 step (b) GRU Prediction - 5 step (c) GRU Prediction - 10 step

Fig. 7: 4G LTE: Comparison of real and predicted values for pedestrian mobility (GRU).

DeepChannel, linear regression and ARIMA.

We observe that the overall predictive performance gain is slightly greater for pedestrian mobility when compared to vehicular mobility. From Figures 4(a) and 4(b) we observe better performance for AT&T in comparison to T-Mobile for pedestrian mobility. However, the performance gains between T-Mobile and AT&T networks are comparable for vehicular mobility (Figures 5(a) and 5(b)). We attribute this change in performance to the actual signal strength variations shown in Figures 4(c) and 5(c). In Figure 4(c), we observe that the signal strength variation for AT&T is smoother in comparison to T-Mobile, which is the primary reason behind the better performance for AT&T. In contrast to this, both AT&T and T-Mobile show higher signal strength variation for the vehicular mobility scenario, leading to similar predictive performance.

1) Qualitative Results: We also present qualitative results to understand the predictive performance of DeepChannel. To this end, we present results for the 4G LTE T-Mobile network for the pedestrian mobility scenario. Figures 6 and 7 show a qualitative comparison between the actual and the prediction results for the LSTM model with respect to linear regression and the GRU model with respect to ARIMA, respectively. The figures illustrate a prediction timeframe of 200 seconds. Figures 6(a)-6(c) and 7(a)-7(c) show the actual values and predictions for time steps 1, 5, and 10, respectively.

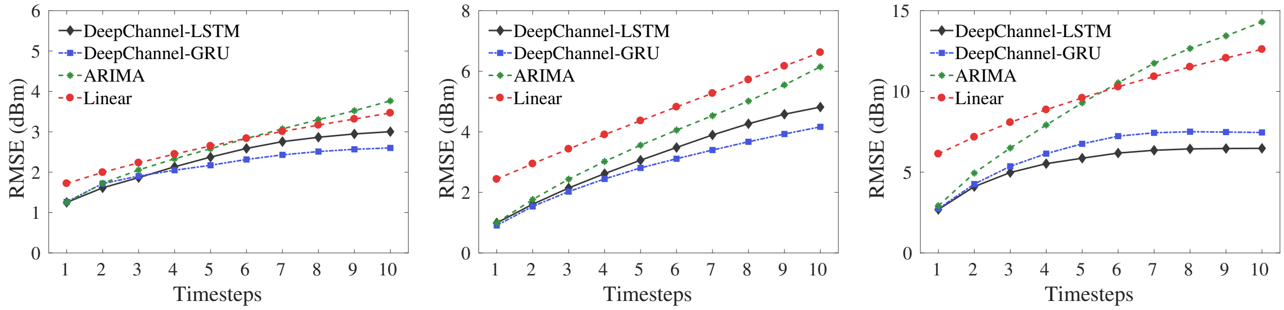
We observe from the figures that linear regression and ARIMA follow the past signal very closely, in particular for time step 1 while predicting the future values, thus resulting in poor predictive performance. This is because predictions for linear regression and ARIMA are dictated by the trend captured by the previous values. As the future may not follow this trend, these baselines provide poor performance. In comparison, DeepChannel captures the underlying correlations in the data, generates smoothened predictions, and thus provides lower prediction error and superior performance.

Additionally, we observe that the performance of the baselines deteriorate more with larger step sizes in comparison to the deep learning models. This correlates with RMSE variations shown in Figure 4(a). From these figures, we observe that though there are some differences (with GRU being smoother than LSTM), the actual predictions of both models are comparable, which also explains the closeness in the RMSE results. These variations in the actual predictions for the LSTM and GRU versions of DeepChannel can be attributed to the architectural differences between the LSTM and GRU cell types.

B. RMSE Results for Other Networks

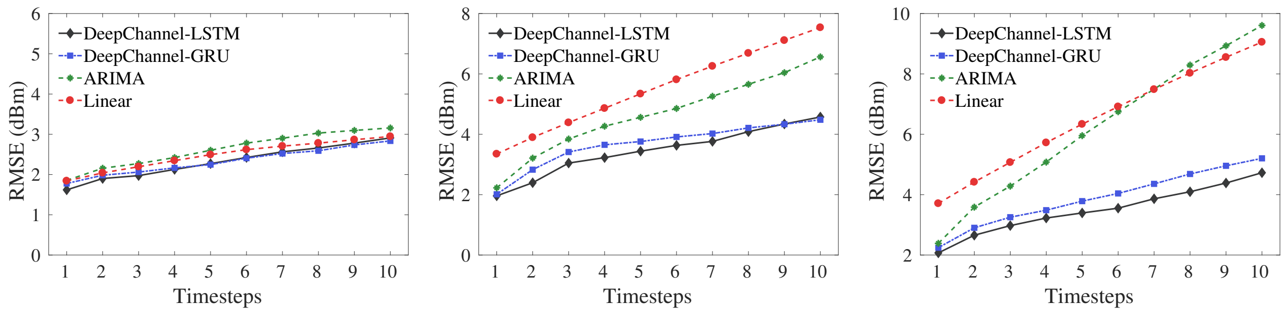
In this subsection, we present performance results comparing DeepChannel with the baselines for all other networks (i.e., WiFi, WIMAX, industrial network operating in the 5.8 GHz range, Zigbee) described in Section VI. Figures 8(a), 8(b), and 8(c) outline the prediction performance for different sampling rates (0.2s, 1s, and 2s) in a WiFi network for a pedestrian mobility scenario. We observe that the deep learning model outperforms the baselines in all cases. Interestingly, we note that the performance gap between the baselines and the deep model increases with the sampling rate. For lower sampling rates, we observe that multiple consecutive recorded signal strength values show little variation. As all measurements are undertaken in a pedestrian mobility scenario, there is little variation in physical position and mobility between consecutive samples at lower sampling rate. This makes it an easier prediction task, thus resulting in the baselines and DeepChannel having comparable performance (Figure 8(a)).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2949954, IEEE Transactions on Vehicular Technology



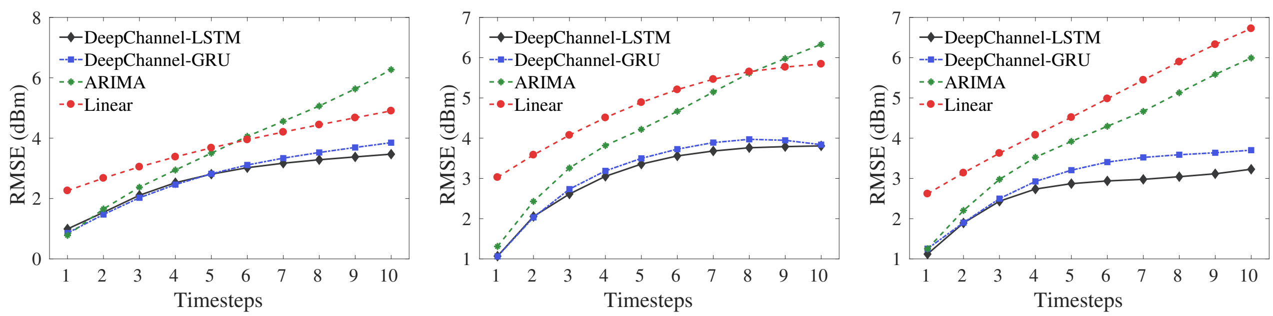
(a) Sampling Rate = 0.2 s (b) Sampling Rate = 1 s (c) Sampling Rate = 2 s

Fig. 8: WiFi experiments.



(a) Pedestrian mobility (outdoor) (b) Pedestrian mobility (indoor) (c) Vehicular mobility

Fig. 9: WiMAX experiments.



(a) Distance = 3.1 m (b) Distance = 10 m (c) Distance = 20.4 m

Fig. 10: Industrial network experiments.

Figures 9(a), 9(b), and 9(c) compare the prediction performance for a WiMAX network for outdoor and indoor pedestrian mobility scenarios, and for vehicular mobility scenario, respectively. Interestingly, we observe similar performance between the deep learning models and the baselines for outdoor pedestrian mobility (Figure 9(a)). To understand this better, let us consider Figure 12(a), where we plot the RSSI variation for the WiMAX outdoor pedestrian mobility trace. We hypothesize the seasonality in the RSSI variation to be the primary reason behind the reduced performance gap between deep learning models and the baselines. While it is clear that the DeepChannel outperforms the baselines, it is also evident that there is no clear winner between the LSTM or GRU versions. Overall 4G LTE, WiFi and WiMAX experiments show the applicability of our proposed model for power prediction in mobile settings.

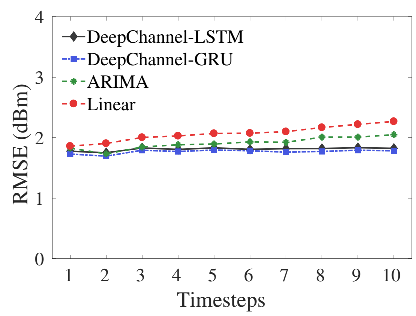
We next investigate the RMSE results obtained for industrial and Zigbee networks in stationary environments in Figures 10 and 11, respectively. Figures 10(a), 10(b), and 10(c) depict the RMSE results for an industrial network setting for antenna separations of 3.1m, 10m, and 20.4m, respectively. Though the proposed model significantly outperforms the baselines, the least performance gap is observed in Figure 10(a), the experiment conducted with the least separation and obstacles. For the Zigbee experiments, we again observe that the deep learning model outperforms the baselines for both separation distances of 10m and 15m (Figures 11(a) and 11(b)). We hypothesize the low overall fluctuations in the channel quality (Figure 12(b)) as the primary reason behind the reduced performance gap between the baselines and the proposed model for the Zigbee network.

C. RE and MAE Results

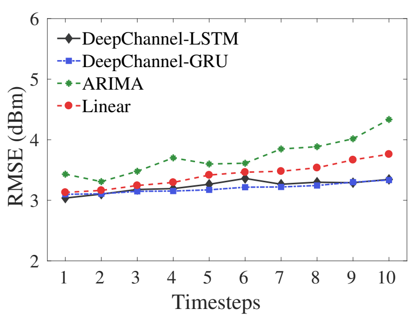
In this subsection, we compare the RE and MAE performance of DeepChannel with respect to the baselines. Table I shows the RE and MAE values as an average over 10 predictive steps for the LSTM and GRU variants of DeepChannel and the ARIMA and linear regression baselines for all network types. The RE results are presented in terms of percentage in the table. Similar to the RMSE results, the RE and MAE values

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2949954, IEEE Transactions on Vehicular Technology

show the superior performance of DeepChannel. We observe that both the LSTM and GRU variants predict channel variations within an average relative error margin of approximately 4%. The performance improvement of both DeepChannel over ARIMA and linear regression with respect to RE and MAE is around 15% and 25% respectively. Once again, we observe comparable performance for all models for the Zigbee datasets, which we attribute to the low overall channel variation.

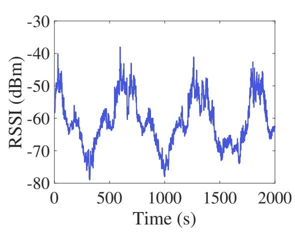


(a) Distance = 10 m

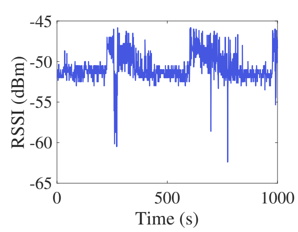


(b) Distance = 15 m

Fig. 11: Zigbee experiments (TxPower level = 31).



(a) WiMAX (outdoor pedestrian mobility)



(b) Zigbee (distance = 10 m)

Fig. 12: Signal strength variations.

D. Discussion on Design Decisions

As discussed in Section V, we determine the parameters of DeepChannel based on extensive exploration of the parameter space by studying the tradeoff between training time and performance. In this subsection, we briefly discuss hyper-parameter tuning and the rationale behind key design decisions in training. Here, we show all findings for the GRU variant of our model for the 4G LTE T-Mobile pedestrian mobility dataset. However, we note that these insights hold true for the LSTM version as well as for other networks

1) Hyper-parameter tuning: Figures 13(a) and 13(b) show the predictive performance of our model obtained by varying the number of stacked layers and the history window size, respectively. We experimentally validate that one or two stacked layers and a history window size of 20 in general provide the best performance. From Figure 13(a), we observe that models with extensive depth result in poor performance. Figure 13(b) 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 the deep model. Figure 13(c) 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. We observe that unguided training 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.

E. Applicability in Rate Prediction Scenarios

In this subsection, we conduct a preliminary study on the applicability of our model in other networking scenarios. Specifically, we consider two problems—i) the problem of predicting future bandwidth variations for video streaming applications over a cellular network, and ii) prediction of bit rates for bit rate adaptation, both of which are closely related to channel quality prediction.

1) Bandwidth Prediction: To this end, we use throughput logs collected for streaming sessions over Telenor's 3G/HSDPA network by researchers at the Simula Research Lab, University of Oslo, Norway [52]. The experiments were conducted in multiple vehicular mobility scenarios (i.e., bus and metro). For each trace, bandwidth measurements were recorded at the seconds' granularity for a time duration of approximately 20 minutes.

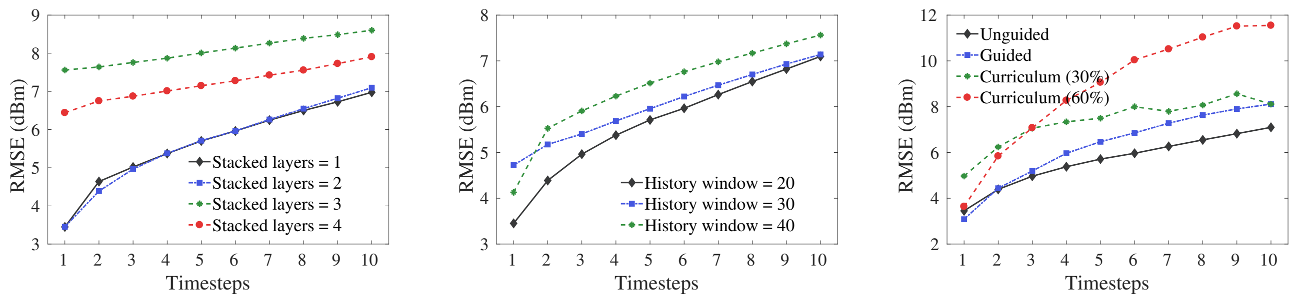
Figures 14(a) and 14(b) illustrate the predictive performance of our model for bandwidth variations for the vehicular mobility scenarios of bus and metro respectively. For these experiments, the deep models have the following parameters—number of layers = 1, number of hidden units = 200, and learning rate = 0.01. Similar to signal strength prediction, we observe that the deep learning models outperform the baselines. Additionally, we note a difference in prediction accuracy with respect to the mobility pattern, with the absolute prediction accuracy being higher for metro than for bus. We attribute this to the bandwidth variations shown in Figure 14(c), where the range of possible variations in bus is higher than in metro.

2) Bit Rate Prediction: For this preliminary study of bit rate prediction, we consider the 4G LTE datasets. We note that the 3GPP LTE enhanced video codec specifications [53] mentions seven possible rates. We assume that the channel variations present in the trace are capable of covering all the rates. We partition the full RSRP channel variations equally into seven ranges and form a one-to-one mapping between them and seven bit rate classes. We assume that these are the optimal bit rates that would be selected by a bit rate selection algorithm (similar to the one in [54]) based on the RSRP

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2949954, IEEE Transactions on Vehicular Technology

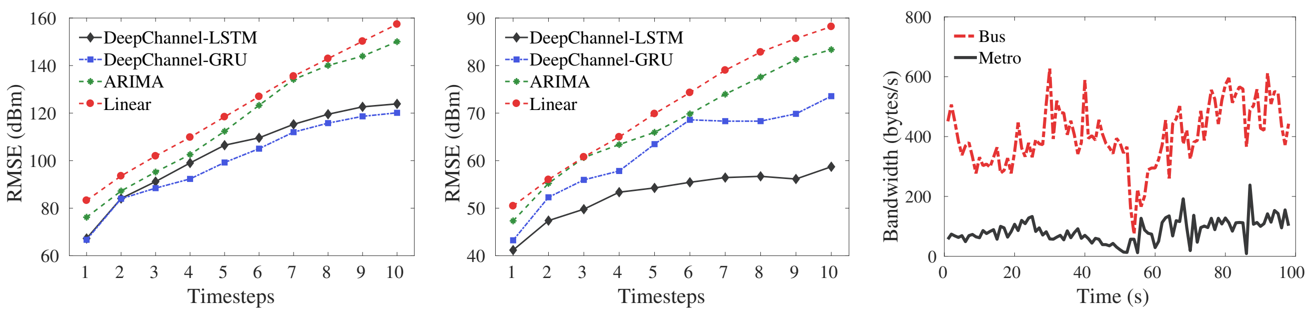
Network Type		Relative Error (%)				Mean Absolute Error			
		LSTM	GRU	ARIMA	Linear	LSTM	GRU	ARIMA	Linear
4G LTE	T-Mobile : Pedestrian mobility	5.11	4.73	6.04	6.75	4.97	4.6	5.9	6.57
	T-Mobile : Vehicular mobility	3.36	3.68	3.8	4.16	3.42	3.7	3.84	4.17
	AT&T : Pedestrian mobility	3.36	3.16	4.03	4.75	3.07	2.88	3.68	4.34
	AT&T : Vehicular mobility	3.22	3.21	3.21	3.89	3.2	3.2	3.25	3.95
WiFi	Sampling rate = 0.2 s	2.39	2.22	2.41	2.72	1.77	1.65	1.79	2.02
	Sampling rate = 1.0 s	3.29	2.89	3.52	4.54	2.27	2	2.44	3.13
	Sampling rate = 2.0 s	6.14	7.91	9.18	10.11	4.17	5.41	6.31	6.91
WiMAX	Pedestrian mobility (outdoor)	3.1	3.18	3.59	3.35	1.76	1.79	2.04	1.89
	Pedestrian mobility (indoor)	3.85	4.09	4.91	5.97	2.84	3.04	3.59	4.36
	Vehicular mobility	4.44	4.05	7.16	8.04	2.92	2.67	4.73	5.32
Industrial Network	Distance = 3 m	3.27	3.35	4.25	5.05	2	2.06	2.55	3.02
	Distance = 10 m	4.65	3.66	5.12	6.34	2.95	2.33	3.23	3.99
	Distance = 20 m	3.06	3.54	4.76	5.82	2.12	2.49	3.3	4.04
Zigbee	Distance = 10 m	2.45	2.4	2.41	2.71	1.23	1.21	1.22	1.37
	Distance = 15 m	3.9	3.76	4.21	3.92	2.16	2.06	2.32	2.16

TABLE I: RE and MAE results averaged over 10 predictive steps.



(a) Impact of Stacked Layers (b) Impact of History Window size (c) Impact of training method

Fig. 13: Impact on parametric changes on performance.



(a) Bus (b) Metro (c) Bandwidth Variations

Fig. 14: 3G/HSDPA bandwidth experiments.

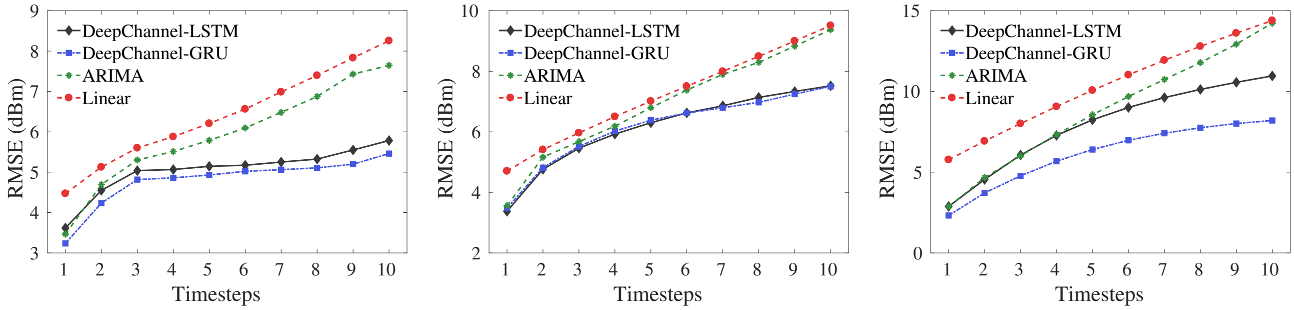
being in a particular range. Both variants of the DeepChannel model have the following parameters—number of layers = 1, number of hidden units = 10, and learning rate = 0.01. As the dataset is relatively simple with only seven possible values, we observe that a single layer with 10 hidden units is sufficient for prediction.

Our goal here is to test the effectiveness of the deep models for a bit rate prediction/control algorithm. Figure 15 illustrates the percentage of correctly predicted bit rates considering the 4G LTE T-Mobile pedestrian mobility dataset. The figure illustrates predictions for time steps of 1, 5, and 10 for the LSTM and GRU variants of the deep learning model and the baselines. Similarly, Table II shows the accuracy of the time step 5 bit rate prediction performance of all the models. Both the figure and the table also show the percentage of incorrect predictions for each model at each step based on the level of deviation between the optimal bit rate and the predicted bit rate. We observe that our deep model has overall higher accuracy than the baselines with the prediction performance for all models decreasing as they predict further into the future. In future, we plan to investigate more realistic bit rate prediction scenarios and the applicability of our model for other networking applications.

VIII. Discussion and Future Work

In this section, we discuss the predictive performance of our trained model on previously unseen (i.e., new) data as well as future research directions. Training deep learning models are computationally expensive involving significant amount of time and computational power. Therefore, it would be helpful if a trained deep learning model can be applied to many application scenarios without re-training. To this end, we conduct a preliminary study to test the robustness of the deep learning model for the channel quality prediction problem

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2949954, IEEE Transactions on Vehicular Technology



(a) 4G LTE - pedestrian mobility

(b) 4G LTE - vehicular mobility

(c) WiFi (sampling rate = 1s)

Fig. 16: Predictive robustness with time (train-test separation = 1 week).

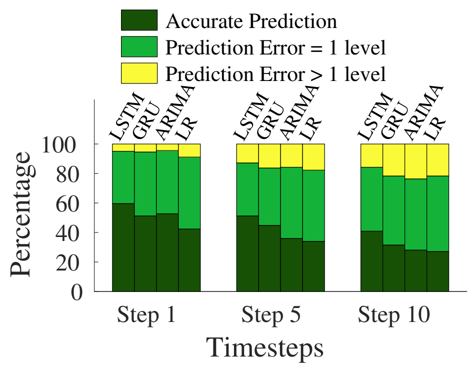


Fig. 15: Bit rate prediction percentage for 4G LTE T-Mobile pedestrian mobility. For each predictive step (1, 5, 10) performance of seq-to-seq LSTM, seq-to-seq GRU, ARIMA and linear regression are labeled as “LSTM”, “GRU”, “ARIMA” and “LR” respectively.

Network Type	LSTM	GRU	ARIMA	Linear
T-Mobile : Vehicular				
Accurate	51.71	52.61	49.02	46.86
Error = 1 level	42.2	40.94	40.94	43.81
Error > 1 level	6.11	6.47	10.06	9.34
AT&T : Pedestrian				
Accurate	53.8	34.82	25.95	21.52
Error = 1 level	9.5	29.75	41.78	36.71
Error > 1 level	36.71	35.45	32.28	41.78
AT&T : Vehicular				
Accurate	44.49	44.86	54.78	41.92
Error = 1 level	47.06	45.23	35.67	40.81
Error > 1 level	8.46	9.93	9.56	17.28

TABLE II: 4G LTE: Time step 5 bit rate prediction percentage (%) for accurate predictions, error = 1 level and error > 1 level.

by investigating if a model trained on one dataset can provide good performance on another dataset.

To conduct this investigation, we collect new datasets for the T-Mobile 4G LTE pedestrian and vehicular mobility scenarios, and WiFi networks in addition to the ones mentioned in Section VI. We then train the deep learning models on the first datasets and test them on the new datasets. Figures 16(a), 16(b), and 16(c) show the predictive performance considering a train-test separation of datasets for 4G LTE pedestrian mobility, 4G LTE vehicular mobility and WiFi networks, respectively. We observe that the deep learning models still outperform the baselines. We also conduct experiments using models trained on the pedestrian mobility dataset and tested on the vehicular mobility dataset and vice versa. Overall, we observe that the performance of DeepChannel deteriorates as it encounters previously unseen data.

A possible reason behind the lower overall performance of the deep learning model is that it is trained on a particular range of signal strength variations and is required to predict a different range at test time. From a machine learning perspective, this means that the sequences that the model sees at train time are different from the sequences it sees at test time, thus resulting in lower performance. An important question that arises is—how to create a reasonably sized training dataset that will enable the deep learning model to observe a wide variety of signal strength sequences such that it can provide superior performance in different network settings? We plan to address this question as part of our future work.

In addition to incurring significantly long training times, one of the other issues with deep learning models is their lack of interpretability. Therefore, in future, we plan to approach the wireless channel quality prediction problem from a graphical modeling perspective, in particular by designing models based on Gaussian Markov Random Fields and Gaussian Conditional Random Fields [55], [56]. In comparison to deep learning models, the results obtained by these graphical models can be easily interpreted and also require less training time. It will also be interesting to analyze how the graphical and deep learning models fare against each another.

IX. Conclusion

In this paper, we investigated the received signal strength prediction problem in wireless networks. We developed DeepChannel, 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 DeepChannel with the ARIMA and linear regression baselines and observed that our model significantly outperforms these baseline models for multiple technologies—4G LTE, WiFi, WiMAX, and Zigbee under varying levels of user mobility and in commercial and industrial environments. The superior performance of our model across different network types signals its practical applicability.

REFERENCES

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

[2] Xuan Kelvin Zou, Jeffrey Erman, Vijay Gopalakrishnan, Emir Halepovic, Rittwik Jana, Xin Jin, Jennifer Rexford, and Rakesh K Sinha. Can accurate predictions improve video streaming in cellular networks? In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pages 57–62. ACM, 2015.

[3] Athula Balachandran, Vyas Sekar, Aditya Akella, Srinivasan Seshan, Ion Stoica, and Hui Zhang. Developing a predictive model of quality of experience for internet video. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 339–350. ACM, 2013.

[4] Mathieu Lacage, Mohammad Hossein Manshaei, and Thierry Turletti. Ieee 802.11 rate adaptation: a practical approach. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 126–134. ACM, 2004.

[5] Xiaozheng Tie, Anand Seetharam, Arun Venkataramani, Deepak Ganesan, and Dennis L Goeckel. Anticipatory wireless bitrate control for blocks. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, page 9. ACM, 2011.

[6] Tijs Van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180. ACM, 2003.

[7] Tiansi Hu and Yunsi Fei. Qelar: A machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks. *IEEE Transactions on Mobile Computing*, 9(6):796–809, 2010.

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

[9] Nicola Bui, Matteo Cesana, S Amir Hosseini, Qi Liao, Ilaria Malanchini, and Joerg Widmer. A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques. *IEEE Communications Surveys & Tutorials*, 19(3):1790–1821, 2017.

[10] Chunxiao Jiang, Haijun Zhang, Yong Ren, Zhu Han, Kwang-Cheng Chen, and Lajos Hanzo. Machine learning paradigms for next-generation wireless networks. *IEEE Wireless Communications*, 24(2):98–105, 2017.

[11] Qian Mao, Fei Hu, and Qi Hao. Deep learning for intelligent wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2018.

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

[13] Mustapha Amara, Afef Feld, and Stefan Valentin. Channel quality prediction in lte: How far can we look ahead under realistic assumptions? In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on*, pages 1–6. IEEE, 2017.

[14] Joseph M Bruno, Yariv Ephraim, Brian L Mark, and Zhi Tian. Spectrum sensing using markovian models. *Handbook of Cognitive Radio*, pages 1–30, 2017.

[15] Peppino Fazio, Mauro Tropea, Cesare Sottile, and Andrea 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.

[16] Hong Shen Wang and Pao-Chi Chang. On verifying the first-order markovian assumption for a rayleigh fading channel model. *IEEE Transactions on Vehicular Technology*, 45(2):353–357, 1996.

[17] Yong Wang, Margaret Martonosi, and Li-Shiuan Peh. Predicting link quality using supervised learning in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(3):71–83, 2007.

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

[19] Shiva Navabi, Chenwei Wang, Ozgun Y Bursalioglu, and Haralabos Papadopoulos. Predicting wireless channel features using neural networks. *arXiv preprint arXiv:1802.00107*, 2018.

[20] Changqing Luo, Jinlong Ji, Qianlong Wang, Xuhui Chen, and Pan Li. Channel state information prediction for 5g wireless communications: A deep learning approach. *IEEE Transactions on Network Science and Engineering*, 2018.

[21] Lu Liu, Bo Yin, Shuai Zhang, Xianghui Cao, and Yu Cheng. Deep learning meets wireless network optimization: Identify critical links. *IEEE Transactions on Network Science and Engineering*, 2018.

[22] Jing Wang, Jian Tang, Zhiyuan Xu, Yanzhi Wang, Guoliang Xue, Xing Zhang, and Dejun 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.

[23] Sami Mekki, Mustapha Amara, Afef Feki, and Stefan Valentin. Channel gain prediction for wireless links with kalman filters and expectation-maximization. In *Wireless Communications and Networking Conference (WCNC), 2016 IEEE*, pages 1–7. IEEE, 2016.

[24] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. Csi-based fingerprinting for indoor localization: A deep learning approach. *IEEE Transactions on Vehicular Technology*, 66(1):763–776, 2017.

[25] Le Thanh Tan and Rose Qingyang Hu. Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning. *IEEE Transactions on Vehicular Technology*, 67(11):10190–10203, 2018.

[26] Hongji Huang, Jie Yang, Hao Huang, Yiwei Song, and Guan Gui. Deep learning for super-resolution channel estimation and doa estimation based massive mimo system. *IEEE Transactions on Vehicular Technology*, 67(9):8549–8560, 2018.

[27] Guan Gui, Hongji Huang, Yiwei Song, and Hikmet Sari. Deep learning for an effective nonorthogonal multiple access scheme. *IEEE Transactions on Vehicular Technology*, 67(9):8440–8450, 2018.

[28] Yibo Zhou, Zubair Md Fadlullah, Bomin Mao, and Nei Kato. A deep-learning-based radio resource assignment technique for 5g ultra dense networks. *IEEE Network*, 32(6):28–34, 2018.

[29] Yu Wang, Miao Liu, Jie Yang, and Guan Gui. Data-driven deep learning for automatic modulation recognition in cognitive radios. *IEEE Transactions on Vehicular Technology*, 2019.

[30] Hongji Huang, Yiwei Song, Jie Yang, Guan Gui, and Fumiyuki Adachi. Deep-learning-based millimeter-wave massive mimo for hybrid precoding. *IEEE Transactions on Vehicular Technology*, 2019.

[31] Hao Huang, Wenchao Xia, Jian Xiong, Jie Yang, Gan Zheng, and Xiaomei Zhu. Unsupervised learning-based fast beamforming design for downlink mimo. *IEEE Access*, 7:7599–7605, 2019.

[32] Ying He, Chengchao Liang, F Richard Yu, Nan Zhao, and Hongxi Yin. Optimization of cache-enabled opportunistic interference alignment wireless networks: A big data deep reinforcement learning approach. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

[33] Jie Wang, Xiao Zhang, Qinhua Gao, Hao Yue, and Hongyu Wang. Device-free wireless localization and activity recognition: A deep learning approach. *IEEE Transactions on Vehicular Technology*, 66(7):6258–6267, 2017.

[34] Yiding Yu, Taotao Wang, and Soung Chang Liew. Deep-reinforcement learning multiple access for heterogeneous wireless networks. *arXiv preprint arXiv:1712.00162*, 2017.

[35] Zubair Md Fadlullah, Fengxiao Tang, Bomin Mao, Nei Kato, Osamu Akashi, Takeru Inoue, and Kimihiro Mizutani. State-of-the-art deep learning: Evolving machine intelligence toward tomorrows intelligent network traffic control systems. *IEEE Communications Surveys & Tutorials*, 19(4):2432–2455.

[36] Nei Kato, Zubair Md Fadlullah, Bomin Mao, Fengxiao Tang, Osamu Akashi, Takeru Inoue, and Kimihiro Mizutani. The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective. *IEEE wireless communications*, 24(3):146–153, 2017.

[37] Partha Dutta, Anand Seetharam, Vijay Arya, Malolan Chetlur, Shivkumar Kalyanaraman, and Jim Kurose. On managing quality of experience of multiple video streams in wireless networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 1242–1250. IEEE, 2012.

[38] Anna Forster. Machine learning techniques applied to wireless ad-hoc networks: Guide and survey. In *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pages 365–370. IEEE, 2007.

[39] Vehbi C Gungor, Gerhard P Hancke, et al. Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Trans. Industrial Electronics*, 56(10):4258–4265, 2009.

[40] Harish Ramamurthy, BS Prabhu, Rajit Gadh, and Asad M Madni. Wireless industrial monitoring and control using a smart sensor platform. *IEEE sensors journal*, 7(5):611–618, 2007.

[41] Lei Tang, Kuang-Ching Wang, Yong Huang, and Fangming Gu. Channel characterization and link quality assessment of ieee 802.15. 4-compliant radio for factory environments. *IEEE Transactions on industrial informatics*, 3(2):99–110, 2007.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2019.2949954, IEEE Transactions on Vehicular Technology

[42] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015.

[43] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[44] Martin Längkvist, Lars Karlsson, and Amy Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24, 2014.

[45] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[46] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

[47] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[48] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[49] Dimitri Block, Niels Hendrik Fliedner, and Uwe 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.

[50] Songwei Fu and Yan Zhang. CRAWDAD dataset due/packet-delivery (v. 2015-04-01). Downloaded from https://crawdad.org/due/packet-delivery/20150401, April 2015.

[51] Ramviyas Parasuraman, Sergio Caccamo, Fredrik Baberg, and Petter Ogren. CRAWDAD dataset kth/rss (v. 2016-01-05). Downloaded from https://crawdad.org/kth/rss/20160105/outdoor, January 2016. traceset: outdoor.

[52] Haakon Riiser, Paul Vigmostad, Carsten Griwodz, and Pål Halvorsen. Commute path bandwidth traces from 3g networks: analysis and applications. In *Proceedings of the 4th ACM Multimedia Systems Conference*, pages 114–118. ACM, 2013.

[53] Kari Jarvinen Stefan Bruhn. Enhanced voice services codec for lte, 2018.

[54] John Charles Bicket. *Bit-rate selection in wireless networks*. PhD thesis, Massachusetts Institute of Technology, 2005.

[55] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[56] Havard Rue and Leonhard Held. *Gaussian Markov random fields: theory and applications*. CRC press, 2005.

Adita Kulkarni is a graduate student in the Computer Science department at SUNY Binghamton. She received her Bachelor of Engineering degree in Computer Engineering from Savitribai Phule Pune University, India. Her research interests include information-centric networks, mobile computing and wireless systems.

Anand Seetharam is an assistant professor in the computer science department at State University of New York Binghamton. He obtained his PhD. from University of Massachusetts Amherst in 2014. He is broadly interested in the field of computer networking. His research encompasses internet-of-things, information-centric networks, wireless networks, and security. He has published numerous papers in peer-reviewed journals and conferences. He co-organized the IEEE INFOCOM 2016 MuSIC workshop and the IEEE MASS 2015 CCN workshop. He has served on the TPC of multiple conferences including IEEE ICC, IEEE ICCCN and IEEE WoWMoM and as reviewer for multiple journals including IEEE TMC, IEEE TNET and Computer Networks.

Arti Ramesh is an assistant professor in Department of Computer Science at Binghamton University. She received her PhD in Computer Science from University of Maryland, College Park. Her primary research interests are in the field of machine learning, data mining, and natural language processing, particularly probabilistic graphical models. Her research focuses on building scalable models for reasoning about interconnectedness, structure, and heterogeneity in socio-behavioral networks. She has published papers in peer-reviewed conferences such as AAAI and ACL. She has served on the TPC/reviewer for notable conferences such as NIPS, SDM, and EDM. She has won multiple awards during her graduate study including the Ann G. Wylie Dissertation Fellowship, outstanding graduate student Dean's fellowship 2016, Dean's graduate fellowship (2012-2014), and yahoo scholarship for grace hopper.

J. Dinal Herath is a graduate student in the computer science department at State University of New York Binghamton. He received his Bachelor of Science degree from Department of Physics, University of Colombo, Srilanka. His research interests include computer networks, wireless networks and cache networks.