Short-Term Prediction of the Attenuation in a Commercial Microwave Link Using LSTM-based RNN

Dror Jacoby*, Jonatan Ostrometzky†‡ *Senior Member, IEEE*, and Hagit Messer* *Life Fellow, IEEE*
†Department of Electrical Engineering, Columbia University in the City of New York
*School of Electrical Engineering, Tel Aviv University
‡Faculty of Engineering, Tel Aviv University

***Abstract*—The signals of microwave links used for wireless communications are prone to attenuation that can be significant due to rain. This attenuation may limit the capacity of the communication channel and cause irreversible damage. Accurate prediction of the attenuation opens the possibility to take appropriate actions to minimize such damage. In this paper, we present the use of the Long Short Time Memory (LSTM) machine learning method for short term prediction of the attenuation in commercial microwave links (CMLs), where only past measurements of the attenuation in a given link are used to predict future attenuation, with no side information. We demonstrate the operation of the proposed method on real-data signal level measurements of CMLs during rain events in Sweden. Moreover, this method is compared to a widely used statistical method for time series forecasting, the Auto-Regression Moving Average (ARIMA). The results show that learning patterns from previous attenuation values during rain events in a given CML are sufficient for generating accurate attenuation predictions.**

***Index Terms*—RNN, Machine Learning Applications, Rain Attenuation Prediction, Time Series, ARIMA**

I. Introduction

A microwave link is a channel of a communication system that uses radio waves in the microwave frequency range to transmit information. Such a system consists of two basic components: transmitter and receiver. The loss of power of the propagating electromagnetic wave is known as the path-loss, and is a result of various factors and depends on the communication channel. In cellular networks, CMLs are used for back-hauling and typically operate in a frequency range of 5–40 GHz, and recently extended to the E-band range (60–90 GHz). The signals at these frequencies are significantly affected by weather conditions, when fading due to rain has a significant impact on the signal's attenuation, mainly above 10 GHz [1].

The relation between the rain-induced attenuation and the rain rate can be modeled by the simplified the power-law [2].

$$A_r(t) = a\bar{r}(t)^b L \tag{1}$$

where $A_r(t)$ (in dB) is the channel path attenuation due to rain at time index t, $\bar{r}(t)$ (in mm/h) is the CML path-averaged rain rate at t, L (in km) is the CML path-length, and $\{a, b\}$ are coefficients that depend on the signal frequency, polarization, and the rain drop size distribution (DSD). It is worth noting, that while rainfall is commonly the most significant weather-related source of attenuation, it is not the only attenuation factor, and other factors have to be considered such as Wet Antenna effect [3], and other precipitations and atmospherics phenomena [4], [5]. Therefore, prediction of attenuation which is based on limited and simplified modeling might produce limited accuracy. CMLs are generally used in communication networks, which include a network management system (NMS). The NMS commonly contains systems designed to react for compensating for changes in the signal level in real-time such as by increasing the transmitted power level (ATPC - automatic transmit power control) or switching the transmission modulation scheme (AMC - adaptive modulation control). The motivation for forecasting future signal levels is clear: When predictions are available, the NMS can compensate for fading before it happens to minimize fluctuations in the signal levels and to improve the overall network Quality of Service (QoS). Moreover, the NMS's ability to rely on its available measurements with no requirement for side information is essential to guarantee the stability and the sustainability of the system.

Attenuation predictions algorithms for satellite-communication systems were suggested, based on probabilistic weather forecasts [6], and time series methods [7]. Preliminary uses of artificial neural networks for rain attenuation and rain-rate prediction were also recently suggested [8], [9]. IIn this paper, we explore the ability to learn from available real-data patterns of signal attenuation for the NMS to produce accurate forecasting of future values in a given single CML. Contrary to previous studies in this field, we focus on applying short-term forecasting with Recurrent Neural Network, based on historical time series of the attenuation, while producing high-resolution predictions of the expected values in a determined interval of time in the future.

We suggest a data-driven method, and for the first time for this application, we use an RNN-based LSTM network for time series forecasting that combines dynamic data of attenuation values and static information of link's characteristics. We compare our method to an Auto-regressive Integrated Moving Average (ARIMA) model, which is a widely used statistical method for time series forecasting [10], [11]. We analyze each method's performance and discuss the limitations of forecasting attenuation levels for short time intervals of 10 seconds to 2 minutes.

The rest of this paper is organized as follows: *Section II* provides a detailed description of our dataset and the necessary needed pre-processing for our experiments. *Section III* presents the methodology, including the problem description of time series forecasting and details the mathematical background of our considered models. *Section IV* describes the

conducted experiments and presents an empirical evaluation of the results. *Section V* provides the conclusion of this paper, as well as a discussion regarding the future potential of our findings.

II. Data Description

We created our data set based on CMLs measurements in Sweden, detailed in [12]. The data include static information, i.e., a meta-data table that characterizes the CMLs network, contains descriptions of the CMLs' physical features, such as length, location (longitude and latitude), frequency, and polarization, as well as dynamic datasets which contain records of the transmitted signal power level (TSL) and received signal power level (RSL) measurements, samples at intervals of $\Delta T = 10$ seconds, per CML. Hence, the total channel path-attenuation level at time t, $A_T(t)$ (in dB), can be derived directly from the measurement by:

$$A_T(t) = TSL(t) - RSL(t) \tag{2}$$

We use sequences of time series of attenuation values, derived from the NMS measurements as a dynamic input to the RNN. The RNN integrates the static information input to the network, which contains the CML physical features. According to (1), the CMLs's frequency and path-length are the two main features that affect the attenuation in the signal. We define the input to the RNN as follows:

$$X_i^{(d)} = [A(t_i), A(t_i+1), ..., A(t_i+N_i)] \tag{3}$$

$$X_n^{(s)} = [L_n, f_n] \tag{4}$$

Where i is an index for the selected day of measurements, and n indicates the link index. L_n, f_n are the link's length (in km), and frequency (in Hz), respectively. Time indices t_i and N_i are pre-determined, in order to set the interval of measurements from each day, in which a rain event is included in the interval. $A(t)$ is the attenuation value after reducing the relatively constant attenuation during dry periods, which commonly referred to as the baseline attenuation [13]. The data set contains 17 different CMLs in a length range of 1.5 km to 7 km and a frequency range of 14 GHz to 39 GHz, and includes measurements from 2015 to 2017 for a total of 1.4k hours of training.

III. Methodology

1) Time Series Forecasting

Formally, the approach for time series forecasting can be expressed as follows: Let y_t denote a variable of interest in a time step t. The target is to forecast the signal value at $t+h; h>0$, given the previous samples, that is:

$$\hat{y}_{t+h} = f_h(y_t, y_{t-1}, \ldots, y_1) \tag{5}$$

$f_h(\cdot)$ denotes a function of all available observations from the current time series, and $h \in (1, ..., H)$, where the value H sets the forecast limit of the prediction. Naturally, when h is larger, the prediction's uncertainty will be more significant. Several different approaches can handle Multi-step time series forecasting (i.e., $h > 1$). We demonstrate two of the approaches through two strategies of performing multi-step ahead forecasts for the next h signal values. First, we suggest the usage of RNN with LSTM cells as the data-driven method, with the implementation of a sequence-to-sequence prediction model. We compare our method to the ARIMA model, a well-known method for univariate time series prediction, as a reference for the comparative evaluation.

2) LSTM Model

Learning techniques and deep learning algorithms have introduced new approaches to time series problems. RNN using LSTM cells [14] are designed to improve gradient flow in deep networks, and allows the model to memorize long term dependencies in the sequential data. We propose an LSTM-based Encoder-Decoder Scheme for time-series forecasting combines dynamic data derived from the attenuation time series with static information, which relates to the physical features that affect the attenuation in different CMLs.

a) Pre-Processing

The input data were divided into sets of sub-sequences, for applying Back-propagation Through Time (BPTT) [15] efficiently on the unrolled RNN with a fixed length. The sliding window concept was adopted for preparing the data by creating input and output sets, fitting the forecasting process as a supervised learning task. For an input window size of T, the dynamic input to the RNN in each time step is constructed from the T previous attenuation values. The output of the LSTM in each time step is a forecast vector for the next H attenuation values. This frame of work is based on the Multi-Input Multi-Output (MIMO) strategy [16] for multi-step time series prediction, where the objective function is the mapping is learned as a single multiple-output model. Both the dynamic and the static data were scaled with basic min-max normalization:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{6}$$

3) Network Architecture

The Encoder-Decoder architecture with LSTM cells was implemented. This structure has been found as efficient in sequence to sequence problems [17] and was proven has efficient for time series forecasting tasks [18], [19].

The model composed of two sub-models: The encoder that reads the input sequences of values $\{x_1, x_2, ..., x_T\}$ sequen-

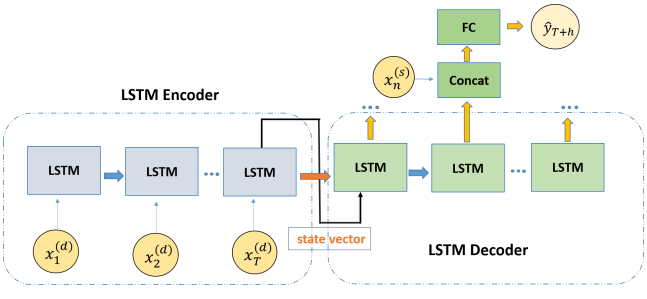


Fig. 1. LSTM Encoder-Decoder Architecture with external static features.

tially, and constructs a fixed-dimensional vector of an internal representation of the series.

The decoder interprets the output hidden state of the encoder to construct the following time steps, in order to generate the forecasting of the next time steps as the target sequence $\{x_{T+1}, ..., x_{T+H}\}$. The objective is to find representative and meaningful features from the input time series.

Then, the sequential output of the decoder layer is concatenated with $x_n^{(s)}$, the auxiliary information of the specific CML's data. The result of this process feeds into a fully connected (FC) layer, which generates a vector of predictions for the following H time steps. **Fig. 1** presents the network architecture of the sequence-to-sequence model for attenuation forecasting. The architecture is inspired by [20], which integrates the time series with auxiliary information through a fully-connected (FC) layer to create the forecasts for each time step. In our case, the goal is to give the network the ability to learn the impact of the link's physical features on the attenuation levels.

We use l_2 norm as the objective function which minimizes the absolute squared differences between the target observations values, $\mathbf{y}_t$, and the outputs, $\hat{\mathbf{y}}_t$.

$$\mathcal{L}(\theta)) = ||\mathbf{y}_t - \hat{\mathbf{y}}_t||_2^2 \tag{7}$$

Where $\mathbf{y}_t, \hat{\mathbf{y}}_t \in R^H$, and θ are the estimated parameters of the proposed model.

A. ARIMA models

1) Mathematical Background

Auto-Regressive Integrated Moving Average (ARIMA) model describes the evolution of the value of variables over the same sample period as a linear function of only their past observations.

The ARIMA model is a generalization of an Auto-Regressive (AR) and Moving Average (MA) models. The integration part of the process refers to the use of a difference-based transformation, which creates an approximate stationary time series. Formally, consider the value of time series in time step y_t:

- $AR(p)$ process is a regression model that uses the dependencies between an observation and a number of p lagged observations: $y_t = \sum_{i=1}^{p} \alpha_i y_{t-i} + \epsilon_t$;
- $MA(q)$ process model the time series as linear combination of the previous q error terms: $y_t = \sum_{i=1}^{q} \beta_i \epsilon_{t-i} + \epsilon_t$.

where ϵ_t are uncorrelated residuals with zero-mean and normally distributed, p is the lag orders (i.e., the number of lag observations included in the model), d is the degree of differencing, and q is the size of the moving average window. $\alpha \in R^p$ and $\beta \in R^q$ are the weights coefficients of the model. ARMA process assumes the data was created from a stationary process, which requires that the statistical properties do not change over time. In most real-world cases, we can not assume that the data do not contain non-stationary components such as trends. Therefore, we apply differencing operation as an invertible transformation in order to work in a stationary frame. The first order differencing transformation of y_t is the operation:

$$\nabla y_t = y_t - y_{t-1}$$

In most cases, the resulting time series will become stationary when the transformation stabilizes the mean and reduce the effect of time in the data. The general form of ARIMA(p,d,q) process is the combination of the models above:

$$\nabla^{(d)} y_t = \sum_{i=1}^{p} \alpha_i \nabla^{(d)} y_{t-i} + \sum_{i=1}^{q} \beta_i \epsilon_{t-i} + \epsilon_t \tag{8}$$

The process of forecasting using the ARIMA model is often generalized by the Box-Jenkins method [21], in order to best fit the model to previous values of the time series. The method is based on the following three-stage steps:

- Model Identification: Determine the model parameters, i.e., p, d, q. The order of d was determined by applying the Augmented Dickey-fuller test (ADF) [22] to verify that our data is stationary. The order of the AR and MA components, p and q, were determined by the values which minimize the Akaike Information Criterion (AIC).

- Estimation: Use the historical data to train the coefficients of the selected ARIMA(p,d,q) model. We apply maximum likelihood estimation that best fits α and β to the time series.

- Diagnostic Checking: Evaluate the fitted model by checking if the estimated model corresponds to our stationary model assumptions, where the residuals should be independent and constant mean and variance over time.

2) Forecasting Method

In order to produce real-time predictions for the CML's attenuation using the ARIMA model, we use the rolling forecast method [23] on the tested events. Meaning, that for each time step, the model is re-trained. We can describe the method by repeating the following stages in each time step t:

- We determine the parameters and estimate the model coefficients according to the Box-Jenkins method from all the previous steps.

- The model predicts the next time step, $\hat{y}_{t+1}$. The prediction is the inverse transformation of the difference between the time-steps, which is calculated by the model equation (8).

- Predictions for $\hat{y}_{t+h}$,where $h \in (2, \ldots, H)$ are computed via iterations, using the previous forecasts as input to the model equation (which is also known as the recursive strategy in performing multi-step forecasting [24]). - The observation y_{t+1} is revealed and added to the training set.

We chose to apply a dynamic rolling forecast on the test data in order to demonstrate a scenario of rain event, in cases where producing real-time predictions from the available data, is required.

This approach is known as the walk-forward validation, where the model is updated in each time step, given that new data is received. This method has shown better results than the usage of past data, which was collected from previous rain events to train ARIMA parameters and to validate the test events. Using this approach, real-time trends in the attenuation patterns are better aligned, which allows for a periodic update of the current values.

IV. Experiments

In this section, we first describe the properties of the tested datasets that we used for the empirical study and detail the experimental setup. Then, we introduce the parameter settings of the RNN network, present the evaluation of our methods on the available datasets and perform an empirical comparison of the proposed methods.

A. Experimental Setup

We consider three rain events (detailed in Table II) to demonstrate our methods and evaluate the performance of predicting the future attenuation values for three different CMLs (detailed in Table I), which were not included in the training dataset and contain a total of 16 hours. The output window size in each time step is $H = 12$, setting the maximum prediction range to Δ = 120 seconds. The input window for training the LSTM network was set to 12 samples by conducting a grid search over the set $T \in \{1, 3, 6, 9, 12, 15, 18\}$.

For the hidden layer size for the encoder and the decoder, we conducted a grid search over the combination of the elements $\{64, 128, 256, 512\}$, and found the that the best performance occurs using a single hidden layer for the encoder and the decoder, each with 128 LSTM units.

For training the LSTM network, we used the Adam [25] optimization algorithm for the implementation of stochastic gradient descent and fit the model with a batch size of 256.

B. Results

We evaluate our method's accuracy using the Root Mean Square Error (RMSE) between the observations and the prediction values as a function of the prediction time.

$$RMSE_h = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(\hat{A}_{t+h} - A_{t+h}\right)^2} \qquad (9)$$

Where A_t is the actual attenuation measurements (in dB), and $\hat{A}_t$ (in dB) is the predicted value. The error metrics are calculated as a function of the prediction time $T_h = 10 \cdot h$ (in seconds). We demonstrate the ability to predict the attenuation values during a rain event in **Fig. 2**, where time series of the total path attenuation in CML during a rain event can be seen, versus the predicted values produced by the two forecasting methods. Each step in the horizontal axis represents 10 seconds interval, meaning the predicted values were calculated three step-ahead (i.e., h = 3) and compared to the observation.

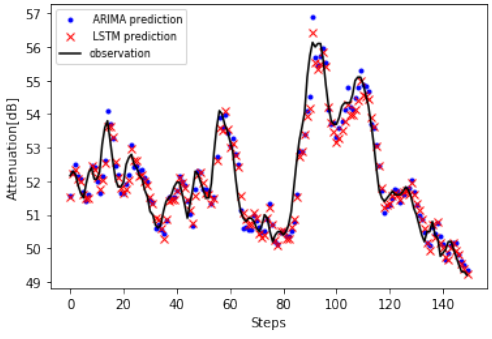


Fig. 2. Actual measurements versus the predicted values by the RNN-LSTM and by the ARIMA models, for a prediction time of 30 seconds

TABLE I
Characteristics of the 3 tested links

CMLs	Length (km)	Frequency (GHz)
A	1.83	37.24
B	2.87	37.21
C	3.64	28.20

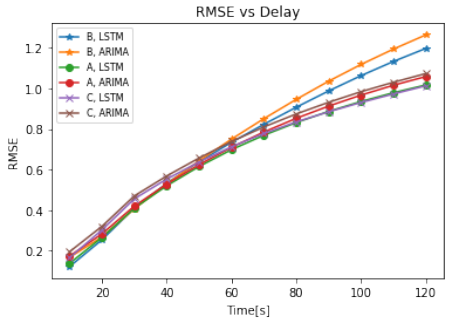
TABLE II
Characteristics of the 3 tested rain events

Date	Duration	Av. Attenuation (dB)
I. 2017-01-11	4h	A: 52.9 , B: 65.7 , C: 63.2
II. 2015-06-02	5h	A: 51.2 , B: 57.9 , C: 61.4
III. 2015-05-05	7h	A: 53.1 , B: 57.3 , C: 60.4

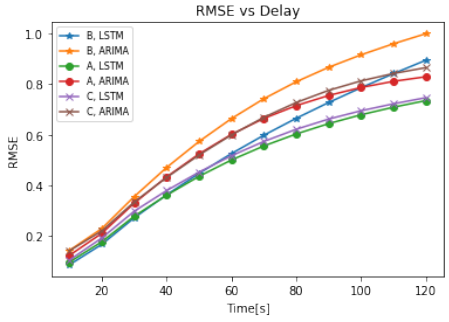
Fig. 3 depicts the $RMSE_h$ of (9) as a function of T_h for both the LSTM and the ARIMA prediction. The results were derived for 3 different CMLs and for 3 different rain events (described in Table II). The results suggest that for all cases, the LSTM-based algorithm performs better than the ARIMA. Nonetheless, the difference in the RMSE is negligible in some cases and more significant in others, which depends on the specific CML and the specific rain event. Both methods achieve high accuracy in shorter-term prediction, and as can be expected, the RMSE increases as the prediction lag, h, increases. The accuracy of the LSTM over the ARIMA increases as the prediction time grows. The outcome results fit the ARIMA model's limitation for generating multi-step forecasting with recursive strategy, which suffers from the growth of errors as long as the forecasting horizon increases. On the other hand, it is worth noting that whereas the LSTM model was pre-trained on previous events, the ARIMA model attains comparable results without any additional information.

V. Conclusion

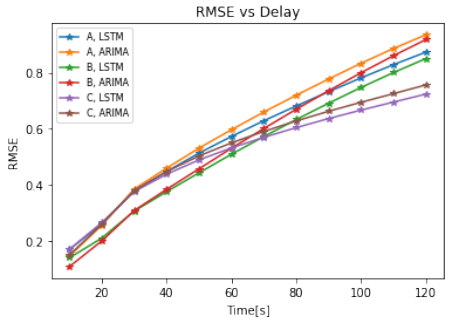
In this paper, we demonstrated two methods for forecasting the attenuation of CMLs signals, based on real data measurements. The presented methods enabled us to generate short-term predictions for the CMLs future attenuation values, based on the historical data of the signal, with high accuracy, and without any additional information. We focused on predicting the attenuation during rainfall events, which causes increased fading in the signal by comparing model and data-driven approaches. Using LSTM-based RNN encoder-decoder architecture, we showed that historical measurements could be used to generate high accuracy predictions of attenuation future values. The training data was collected from several different CMLs (in addition to the CMLs used for testing our model's accuracy, which we did not use as part of the training). Therefore, we used an RNN architecture that combines both static and dynamic information for the adjustment of the



(a) Event I



(b) Event II



(c) Event III

Fig. 3. RMSE values as a function of the prediction time lag, for all values between 10 and 120 seconds. Each panel contains results from a different rain event (see Table II). The RMSE values are derived for each of the 3 CMLs (see Table I) for both the ARIMA and the LSTM algorithms.

predictions for different CMLs. We compared our results to the ARIMA model using a rolling forecast method on the test data. Based on the empirical demonstration we performed, we show that both methods achieved high accuracy, where the LSTM-based method performs better, with respect to the prediction accuracy, as we increase the prediction time. For future research, we will extend the time interval of the predictions and consider advanced time series forecasting methods, which is a fast-growing area of research, in order to improve the forecast accuracy further. Furthermore, we will use our predictions for the attenuation values to develop algorithms for predictive weather-aware wireless network management to compensate for the expected attenuation in the signals in real-time.

REFERENCES

[1] Robert K Crane. *Electromagnetic wave propagation through rain*. Wiley-Interscience, 1996.

[2] R. Olsen, D. V. Rogers, and D. Hodge. The ar b relation in the calculation of rain attenuation. *IEEE Transactions on antennas and propagation*, 26(2):318–329, 1978.

[3] Jonatan Ostrometzky, Roi Raich, Lei Bao, Jonas Hansryd, and Hagit Messer. The wet-antenna effectâa factor to be considered in future communication networks. *IEEE Transactions on Antennas and Propagation*, 66(1):315–322, 2017.

[4] Robert K Crane. *Propagation handbook for wireless communication system design*. CRC press, 2003.

[5] Farukh Nadeem, Stefano Chessa, Erich Leitgeb, and Safdar Zaman. The effects of weather on the life time of wireless sensor networks using fso/rf communication. *Radioengineering*, 19(2), 2010.

[6] Isabelle Dahman, Nicolas Jeannin, Philippe Arbogast, and Bouchra Benammar. Attenuation forecasts model exploiting short range probabilistic weather forecasts. 2016.

[7] Dalia Das and Animesh Maitra. Time series predictor of ku band rain attenuation over an earth–space path at a tropical location. *Int. Jour. of Satellite Communications and Networking*, 30(1):19–28, 2012.

[8] MN Ahuna, TJ Afullo, and AA Alonge. Rain attenuation prediction using artificial neural network for dynamic rain fade mitigation. *SAIEE Africa Research Journal*, 110(1):11–18, 2019.

[9] Ibukun Daniel Olatunde, Kazeem Oladele Babatunde, and David Oluwarotimi Afolabi. Rain attenuation prediction in nigeria using artificial neural network (ann). *International Journal of Electrical and Electronic Science*, 6(1):1–7, 2019.

[10] Sasan Barak and S Saeedeh Sadegh. Forecasting energy consumption using ensemble arima–anfis hybrid algorithm. *International Journal of Electrical Power & Energy Systems*, 82:92–104, 2016.

[11] Sui-Lau Ho, Min Xie, and Thong Ngee Goh. A comparative study of neural network and box-jenkins arima modeling in time series prediction. *Computers & Industrial Engineering*, 42(2-4):371–375, 2002.

[12] L Bao, C Larsson, M Mustafa, J Selin, JCM Andersson, J Hansryd, M Riedel, and H Andersson. A brief description on measurement data from an operational microwave network in gothenburg, sweden. In *15th International Conference on Environmental Science and Technology, Rhodes, Greece*, volume 31.

[13] Jonatan Ostrometzky and Hagit Messer. Dynamic determination of the baseline level in microwave links for rain monitoring from minimum attenuation values. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(1):24–33, 2017.

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

[15] Michael C Mozer. A focused backpropagation algorithm for temporal. *Backpropagation: Theory, architectures, and applications*, 137, 1995.

[16] Nguyen Hoang An and Duong Tuan Anh. Comparison of strategies for multi-step-ahead prediction of time series using neural network. In *2015 International Conference on Advanced Computing and Applications (ACOMP)*, pages 142–149. IEEE, 2015.

[17] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[18] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*, 2017.

[19] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

[20] Lingxue Zhu and Nikolay Laptev. Deep and confident prediction for time series at uber. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 103–110. IEEE, 2017.

[21] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[22] J Humberto Lopez. The power of the adf test. *Economics Letters*, 57(1):5–10, 1997.

[23] RJ Hyndman. Variations on rolling forecasts, 2014.

[24] Souhaib Ben Taieb, Rob J Hyndman, et al. *Recursive and direct multi-step forecasting: the best of both worlds*, volume 19. Citeseer, 2012.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.