



Alexandria University
Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com



Link quality prediction in wireless community networks using deep recurrent neural networks



Mohamed Abdel-Nasser^{a,c,*}, Karar Mahmoud^{b,c}, Osama A. Omer^c,
 Matti Lehtonen^b, Domenec Puig^a

^a *Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Tarragona 43007, Spain*

^b *Department of Electrical Engineering and Automation, Aalto University, FI-00076 Espoo, Finland*

^c *Department of Electrical Engineering, Aswan University, 81542 Aswan, Egypt*

Received 5 March 2020; revised 14 May 2020; accepted 26 May 2020

Available online 26 June 2020

KEYWORDS

Link quality prediction;
 Time-series analysis;
 Deep learning;
 RNN;
 LSTM;
 GRU

Abstract Wireless community networks (WCNs) are large, heterogeneous, dynamic, and decentralized networks. Such complex characteristics raise different challenges, such as the effect of wireless communications on the performance of networks and routing protocols. The prediction approaches of link quality (LQ) can improve the performance of routing algorithms of WCNs while avoiding weak links. The prediction of LQ in WCNs can be a complex task because of the fluctuated nature of LQ measurements due to the dynamic wireless environment. In this paper, a deep learning based approach is proposed to accurately predict LQ in WCNs. Specifically, we propose the use of two variants of deep recurrent neural network (RNN): long short-term memory recurrent neural networks (LSTM-RNN) and gated recurrent unit (GRU). The positive feature of the proposed variants is that they can handle the fluctuating nature of LQ due to their ability to learn and exploit the context in LQ time-series. The experimental results on data collected from a real-world WCN show that the proposed LSTM-RNN and GRU models accurately predict LQ in WCNs compared to related methods. The proposed approach could be a helpful tool for accurately predicting LQ, thereby improving the performance of routing protocols of WCNs.

© 2020 The Authors. Published by Elsevier B.V. on behalf of Faculty of Engineering, Alexandria University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Wireless community networks (WCNs) are owned and managed by community members [1]. FunkFeuer [2], Gifui.net [3], and Ninux [4] are typical examples of WCNs. These well-known networks enable members to access the Internet and

provide various free services. WCNs are decentralized infrastructures comprising many nodes and links and their size increases dynamically. Indeed, the unreliability and asymmetrical features of links in wireless networks affect the network performance and routing protocols, thereby degrading the delivery rate and traffic congestion. To cope with these issues, the routing protocols of these networks use various link quality (LQ) estimators to specify the best available paths considering higher-quality links. LQ estimators can measure the quality of links between nodes based on physical or logical

* Corresponding author.

E-mail address: egnaser@gmail.com (M. Abdel-Nasser).

Peer review under responsibility of Faculty of Engineering, Alexandria University.

<https://doi.org/10.1016/j.aej.2020.05.037>

1110-0168 © 2020 The Authors. Published by Elsevier B.V. on behalf of Faculty of Engineering, Alexandria University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

metrics, such as signal-to-noise ratio (SNR), received signal strength and packet success rate. However, they provide insufficient information about the quality of links in the future, i.e., they cannot give predictions for the upcoming LQ values. It is a fact that network protocols require an accurate LQ prediction method to predict LQ in advance besides modeling past LQ measurements.

The term LQ can be generally defined as link characterization that is related to throughput or reliability [5]. LQ can be also explained as the portion of positive probes received by a node from its neighboring node during a specified time duration. In turn, a neighbor LQ (NLQ) represents the portion of successful probes received by the neighboring node within a specified time duration. In this context, a probe can be defined as packets that each device sends automatically, from time to time, searching all-around to check if a network is nearby, and so they can connect. The values of LQ and NLQ are used to compute the expected transmission count (ETX) [6]. ETX is a widely used link metric for routing algorithms of WCNs which calculates the number of expected transmissions for a certain packet to be correctly transmitted from the sending node to the receiving node. For instance, in [7], a combination of different link metrics (ETX, minimum delay, and minimum loss) with link state routing (OLSR) was used to improve the reliability of wireless mesh networks in a smart grid.

To estimate LQ, the common procedure is to select the measurements that are periodically sampled from the links (LQ metrics) [8,9]. These metrics can be physical (hardware-based) or logical (software-based) [10]. The authors of [11] reported three general approaches for modeling LQ empirically: analytical models, probabilistic estimation models, and statistical prediction models. Machine learning algorithms are typically exploited in LQ estimation, where analytical models are anonymous, inexact, or difficult to be derived. Tao et al. [12] applied machine learning methods to predict the short-term LQ values for facilitating the use of middle links with repeated quality changes. They stated that by utilizing machine learning techniques, their adaptive link estimator can be adapted to the dynamics of the network better than the models that have been trained in a static manner. In [13], an approach, called XCoPred (cross-correlation to predict) was proposed for LQ prediction. This approach is based on pattern matching to estimate the future state of each link in terms of its quality variations. XCoPred does not consider potential emerging links (i.e. links that may be created in the future) since it makes predictions on the basis of the existing links of each node to its neighbors.

Besides, it does not need the use of extra hardware where it depends on SNR measurements. Miguel et al. [14] studied the performance of machine learning techniques for predicting LQ and proposed a hybrid on-line LQ prediction technique. Specifically, they used support vector machines (SVM), k-nearest neighbors (kNN), regression trees (RT) based on reduced error pruning, and Gaussian processes for regression (GPR). Millan et al. [15] presented LQ prediction methods for the FunkFeuer WCN based on time-series analysis techniques. They used KNN, RT, SVM and GPR algorithms, analyzed the error variability of each algorithm, and represented the results using boxplots. It is demonstrated in their study that the four utilized algorithms obtained a similar accuracy rate for most links. Furthermore, Millan et al. [16] demon-

strated that time-series analysis is a promising approach to accurately predict LQ values in WCNs and used SVM, kNN, RT, and rule-based regression (RBR) to build the LQ prediction models.

In addition, a conditional restricted Boltzmann machine (RBM) was used in [17] to predict LQ in an opportunistic sensor network. Additionally, a similarity index based on time parameters was also proposed to describe similarities between nodes. RBM represents the LQ time-series by adding connections from previous time steps. The efficacy of RBM was verified using INFOCOM and MIT datasets. A data-driven LQ estimation approach, called 4C, was proposed in [18] which comprises three steps: data collection, offline modeling, and online prediction. In the 4C approach, three well-known machine learning methods (Naive Bayes, logistic regression and artificial neural networks-ANN) were used for the off-line model training. In [19], the use of wavelet decomposition has been introduced for improving link quality prediction. Specifically, a wavelet-neural-network (WNN) was used to predict LQ in which SNR is used as an LQ metric. In this method, the SNR is decomposed into two parts: a time-varying part and a non-stationary random part and then they were fed into the WNN model. The authors of [20] utilized online learning methods to model different conditions of a wireless network without human interference. They mentioned that considerable LQ measurements should be collected for all links before building the prediction model. Weng et al. [21] introduced a nonparametric algorithm, called functional-coefficient autoregression, for time-series analysis to predict LQ online. The performance of this algorithm was assessed by various datasets: data collected from NS-2 simulation, data collected from the CRAWDAD wireless network, and the ANDES lab dataset. Chenhao et al. [22] proposed a cloud reasoning method that considers both short- and long-term time dimensions and current and historical cloud models to predict LQ in wireless sensor networks. They stated that their algorithm can capture the changes on links more precisely than the window mean exponentially weighted method.

1.1. Motivation

Accurate LQ prediction of WCNs is a complex task because of the fluctuated nature of LQ measurements due to the noticeable dynamic and unpredictable wireless environment. Indeed, LQ changes randomly across time and space; and so LQ can be seen as a random time-series containing nonlinear and non-stationary features. This is the main reason why several methods in the above-mentioned literature review can produce inaccurate prediction results since most of them utilize traditional machine learning approaches. These shallow learning approaches exploited in machine learning based algorithms are insufficient for extracting features from LQ measurements of large-scale dynamic WCNs. In other words, they cannot fully represent the dynamic random features of wireless links. Notably, conventional ANN methods do not have memory units or recurrent mechanism, and thus they cannot handle the fluctuations in LQ datasets. In turn, deep learning approaches can have sufficient ability for learning relevant features of LQ measurements. Recently, deep learning has been successfully applied to different engineering problems, such as mobile advertising in vehicular networks [23], time-series

prediction [24], signal detection and channel estimation in orthogonal frequency-division multiplexing systems [25,26], phase fingerprinting for indoor localization [27], peak-to-average power ratio reduction scheme [28], and large-scale light curve time series prediction [29]. Recently, deep learning models, such as recurrent neural networks (RNNs), long short-term memory (LSTM) and gated recurrent units (GRU) have been used to handle the problems of time series prediction. They have been used in different tasks, for instance, a novel deep recurrent neural network technique was proposed in [30] which combines Savitzky-Golay filter with LSTM for predicting a task time series in cloud computing systems [31]. It has been demonstrated that the introduction of LSTM to time-series prediction results in significant benefits because it can cope with the difficulties of training RNNs and avoid the problem of long-term dependency in RNNs. Therefore, it is important to investigate/develop the application of LSTM and GRU to the LQ prediction in WCNs, which is the main scope of the paper. Notably, existing LQ prediction methods discussed in the above-mentioned literature use traditional machine learning techniques, such as KNN, RT, SVM and GPR algorithms. These traditional machine learning techniques may not handle the fluctuating nature of LQ measurements. Unlike the above-mentioned studies, we propose to use deep learning approaches in nonlinear time-series analysis to predict LQ based on historical information of the quality of all links.

1.2. Contribution and paper structure

Most existing LQ prediction methods in WCNs employ traditional machine learning techniques, which may not handle the fluctuating nature of link quality. In this paper, deep learning based approach is proposed to accurately predict LQ in WCNs. Specifically, we propose the use of deep LSTM-RNN and GRU for LQ prediction. To the best of our knowledge, this is the first attempt to propose the use of deep LSTM-RNN and GRU for LQ prediction in WCNs. The proposed approach models LQ in WCNs using historical LQ time-series data. The unique feature of the proposed approach is that it can address the fluctuating nature of LQ values, thanks to its ability to learn and exploit the context in time-series data.

Experiments are performed on a dataset collected from the Funkfeuer Wien WCN [1], which includes more than 500 nodes and around 2000 links, demonstrating that the proposed approach achieves very small LQ prediction errors. To demonstrate the efficacy of the proposed approach, it has been com-

pared with related methods. Based on the obtained results, the introduction of this deep learning approach greatly increases the accuracy rate of LQ prediction models and so it will have considerable positive impacts on the performance of WCNs in general.

The rest of this paper is structured as follows. Section 2 explains the proposed approach. Section 3 provides the experimental results and discusses them. Section 4 summarizes the paper and suggests some lines of future work.

2. Proposed approach

To demonstrate the LQ prediction problem, we provide a simple example of a WCN which comprises 6 nodes (Fig. 1). LQ measurements are collected at regular time steps from each link in the wireless network to construct the LQ model for the six nodes (node1, node2, node3, node4, node5, and node6). As shown, the LQ prediction algorithm uses the historical LQ data (past LQ measurements) to generate a model for node3. Then, the constructed model of node3 is used to predict the future values of LQ for that node. To build an accurate LQ model, we propose the use of deep LSTM-RNN and GRU methods.

It is important noting that the proposed approach is data-driven, and thus we do not deal with the system models or modeling dedicated for wireless transmissions. In the next subsections, we explain the basics of RNN, LSTM, and GRU, and we present the proposed approach in detail.

2.1. Recurrent neural networks (RNNs)

Deep RNNs have considered a variant of the traditional artificial neural networks and have achieved good results in the case of sequence data [32]. Indeed, traditional ANNs do not utilize the data computed at preceding past steps ($\dots, t-2, t-1$) of a specified sequence when calculating the value of the present step. In other words, traditional ANNs have no memory units. In turn, RNNs have feedbacks where they pass data of a current time step into the upcoming time steps. On the basis of the theoretical perspective, RNNs can utilize information in the case of long sequences, and they also look back a specified time steps.

Fig. 2 can simplify the understanding of the RNN model by unfolding or unrolling the RNN graph over the input sequence. This is a useful way for visualising RNNs while considering the updated graph formed via unfolding the network

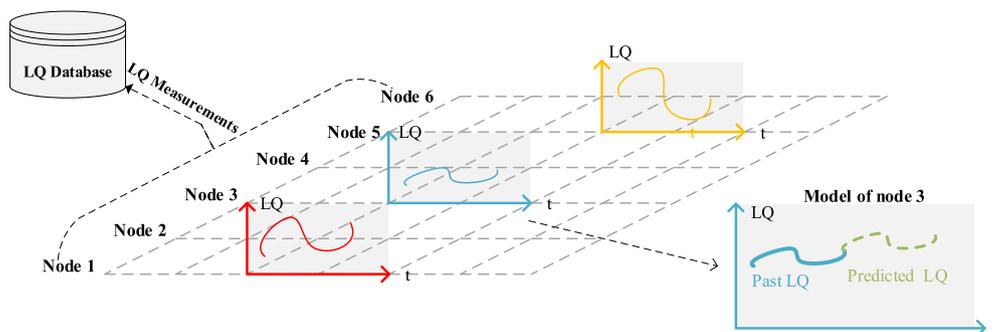


Fig. 1 Explanation of the LQ prediction problem.

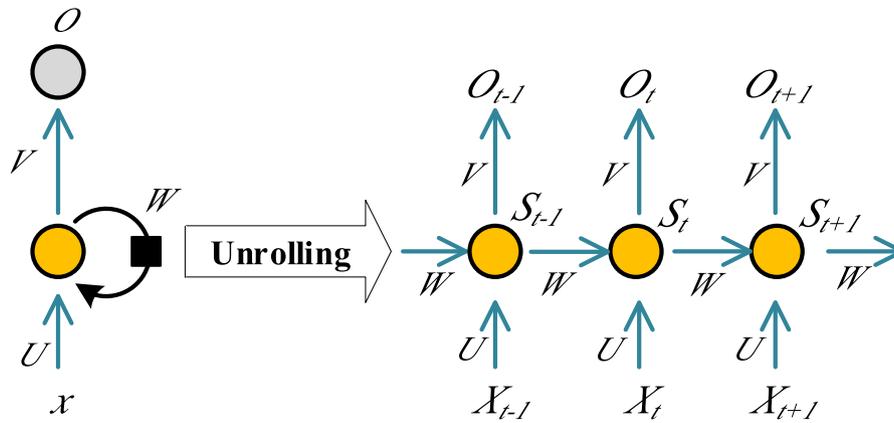


Fig. 2 The block diagram of RNN.

along the input sequence. The unfolding process can be considered as copying the network at each time step in the time-series data. Note that such unrolling of the RNN graph is common in the literature that is used to explain the mechanism of RNN. For example, in the case of considering time-series data which have 4 samples, RNN can be unfolded into a specified neural network which includes 4 different layers, where a single layer is assigned for each element in the time sequence data. In Fig. 2, x_t and s_t represent the input sample and the hidden state at time step t , respectively. RNN handles a specified sequence $x = (x_1, x_2, \dots, x_T)$ with a varied length, and the output $y = (o_1, o_2, \dots, o_n)$ can be of a non-consistent length. Note that during the time step t , the past hidden state and the current sample are employed to compute s_t by:

$$s_t = f(Ux_t + Ws_{t-1}) \quad (1)$$

In Eq. (1), f represents an activation function (e.g. tanh) which is non-linear. In order to calculate the initial hidden state, s_{-1} is required to be adjusted to zeros. The unfolding process of s_t can be mathematically expressed as follows:

$$s_t = f(s_{t-1}, x_t; \theta) = f(Ux_t + Ws_{t-1}) \quad (2)$$

$$\begin{aligned} s_t &= f(s_{t-1}, x_t; \theta) = f(f(s_{t-2}, x_{t-1}; \theta), x_t; \theta) \\ &= f(f(f(s_{t-3}, x_{t-2}; \theta), x_{t-1}; \theta), x_t; \theta) \end{aligned} \quad (3)$$

in which θ involves the parameters of RNN. In addition, the state s_t is related with the input samples $(x_t, x_{t-1}, x_{t-2}, \dots)$ as follows:

$$s_t = g_t(x_t, x_{t-1}, x_{t-2}, \dots) \quad (4)$$

It is important to note that the sample at time step t has a unique g_t function. However, all the samples have a similar f function. The output of RNN at time step t (o_t) can be computed by:

$$o_t = \text{softmax}(Vs_t) \quad (5)$$

It is a fact that s_t has the capability to detect information of the previous steps. Generally, the deep networks use diverse parameters for each layer, but RNNs share the parameters (U, V, W) of the different time steps, and so they duplicate the tasks. Therefore, RNNs minimize the required parameters for building their training models.

Back-propagation through time is considered a traditional method which computes the sufficient gradients for training

a particular RNN model. Note that the gradients can be expressed as follows [33]:

$$\frac{\partial E}{\partial \theta} = \sum_{1 < t \leq T} \frac{\partial E_t}{\partial \theta} \quad (6)$$

in which E refers to a special cost function which measures the RNN operation during a specified task, where it can be transformed to independent cost functions for each step.

$$\frac{\partial E_t}{\partial \theta} = \sum_{1 < k \leq t} \frac{\partial E_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial^+ h_k}{\partial \theta} \quad (7)$$

Note that $\frac{\partial^+ h_k}{\partial \theta}$ represents the derivative of h_k with respect to θ . The gradient component represented by $\frac{\partial E_t}{\partial \theta}$ is considered as the total of temporal components represented by $(\frac{\partial E_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial^+ h_k}{\partial \theta})$. Each of these component can have long-term and/or short-term roles. Note that the long term refers to special components that satisfies $k \ll t$.

$$\frac{\partial h_t}{\partial h_k} = \prod_{t \geq i > k} \frac{\partial h_i}{\partial h_{i-1}} = \prod_{t \geq i > k} W_{rec}^T \text{diag}(\hat{\sigma}(h_{i-1})) \quad (8)$$

where diag converts a vector into a special diagonal matrix, and $\hat{\sigma}$ computes the first derivative of σ . The use of $\frac{\partial h_t}{\partial h_k}$ ensures the transportation of the error from the time step t to the step k .

Bengio et al. [34] stated that there are difficulties when training RNNs for modeling long term dependencies. The reason behind this argument is the vanishing/explosion problem of the gradients of the utilized cost function (as can be noticed in Eq. (7)). To cope with this issue, two different approaches were proposed in the literature review. In the first approach, improved learning methods, instead of the stochastic gradient descent methods, are used. In the second approach, modified activation functions, which include affine transformation and non-linear gating units, are utilized. LSTM and GRU are common examples of the gating units. In this paper, we propose the use of LSTM and RNN to predict LQ in WCNs. Below, we explain each method in details.

2.1.1. Long short-term memory recurrent neural networks (LSTM-RNN)

To cope with the difficulties of training RNNs, Hochreiter and Schmidhuber proposed a recurrent unit called LSTM that can

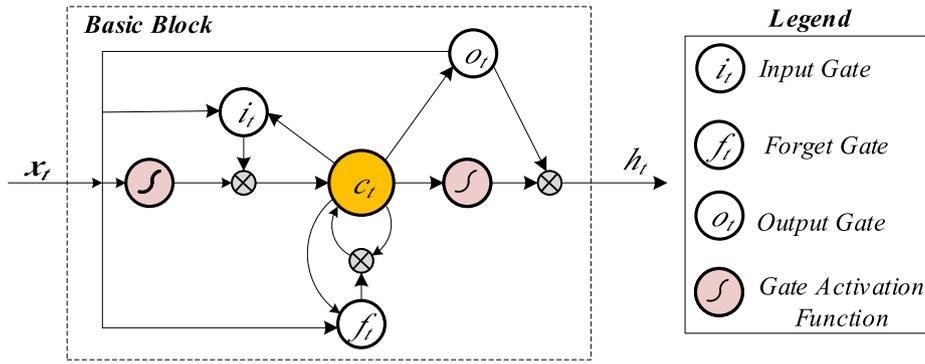


Fig. 3 The block diagram of LSTM.

learn short and long-term dependencies [35]. LSTM was proposed to avoid the problem of long-term dependency in RNNs, due to their memory blocks connected across layers. Each LSTM block has gates to estimate its state and the desired output. There are three gates in the LSTM block: forget gate, input gate, and the output gate.

Fig. 3 shows the block diagram of LSTM, where it receives an input time-series in which the activation units are used to trigger the gates. Each gate has weights that can be learned in the training stage of LSTM-RNN. The gates enable LSTM-RNN to memorize the recent steps. Each LSTM block has a cell with a state c_t at time step t . The input gate i_t , forget gate f_t , and output gate o_t are used to manage the reading or updating processes of this cell. At each time step t , LSTM reads inputs from two external sources at each of its four terminals (the forget gate, input gate, output gate, and the input). The first external source is the current sample x_t while the second one is the previous hidden states of LSTM blocks in the same layer h_{t-1} . The operation of LSTM can be mathematically formulated as follows:

$$i_t = \sigma(W_{xi}X_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (9)$$

$$f_t = \sigma(W_{xf}X_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (10)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}X_t + W_{hc}h_{t-1} + b_c) \quad (11)$$

$$o_t = \sigma(W_{xo}X_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (12)$$

$$h_t = o_t \tanh(c_t) \quad (13)$$

where W refers to the weight of the corresponding hidden neuron in the LSTM block.

2.1.2. Gated recurrent unit (GRU)

To cope with the limitations of RNN, Cho et al. proposed a variant of the recurrent units called GRU [36]. As demonstrated in Fig. 4, GRU has gating units representing the flow of information. Note that GRU does not have memory, and the activation function h_t^j of GRU at step t can be represented by the interpolation between the past activation h_{t-1}^j and the present activation \tilde{h}_t^j

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j\tilde{h}_t^j \quad (14)$$

where z_t^j controls the activation process, which is called the update gate, which can be calculated as follows:

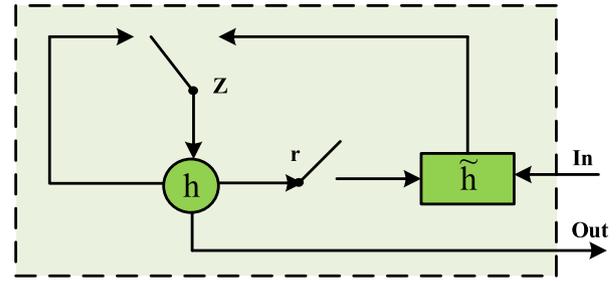


Fig. 4 Block diagram of GRU.

Table 1 The data structure of LQ measurements.

Link number	$LQ_{(t-1)}$	$LQ_{(t)}$
7	0.694	0.776
8	0.035	0.039
9	0.149	0.110
10	0.839	0.918
20	1	1

$$z_t^j = \sigma(W_z X_t + U_z h_{t-1}^j) \quad (15)$$

The GRU does not manage the condition of its state. However, it exposes the entire state at each step. The candidate activation \tilde{h}_t^j is expressed as follows:

$$\tilde{h}_t^j = \tanh(W X_t + U(r_t \odot h_{t-1}^j)) \quad (16)$$

in which r_t represents a group of reset gates, and \odot represents an element-wise reduplication. In the case of *turn off* state (r_t^j is approaches to 0), the reset gate enables the unit to read the data of the first sample, and so GRU removes the information of the previous states. The reset gate r_t^j is represented by:

$$r_t^j = \sigma(W_r X_t + U_r h_{t-1}^j) \quad (17)$$

2.2. LQ prediction using deep LSTM-RNN and GRU

In this paper, we propose the use of two deep RNN variants (LSTM and GRU) to model LQ. In Table 1, we present an example of the data structure of the proposed LQ prediction approach. As shown, the link 9 has an LQ value of 0.149 at

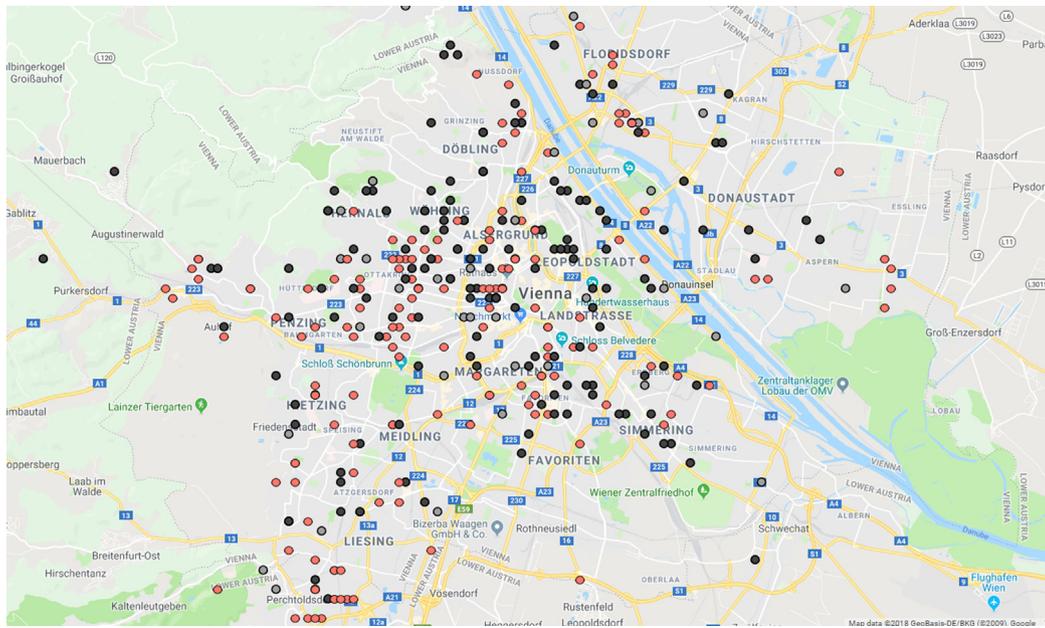


Fig. 5 Interactive map for Funkfeuer Wien WCN, which includes active and inactive nodes [37]. The red points refer to the active nodes while the black ones represent the inactive nodes.

Table 2 The performance of the proposed LSTM-RNN approach.

Proposed Approach	Average MSE \pm Std	Average RMSE \pm Std	Average MAE \pm Std
2 LSTM	0.0066795 \pm 0.00057553	0.0705911 \pm 0.00086189	0.0432364 \pm 0.00084861
5 LSTM	0.0065654 \pm 0.00071958	0.0809786 \pm 0.00051683	0.0432292 \pm 0.00066084
7 LSTM	0.0065765 \pm 0.00026346	0.0810479 \pm 0.00087782	0.0446647 \pm 0.00048420

the time step $(t - 1)$ and an LQ value of 0.11 at the next time step. The two proposed RNN variants are employed here to build a model that formulates the LQ prediction problem as a regression problem. Given the LQ value of the current time step t_i , the goal is to predict the LQ at the next time step t_{i+1} . The model receives the LQ values at the time steps LQ_{t-2} , LQ_{t-1} , and LQ_t as inputs, and then it predicts the LQ value at the time step LQ_{t+1} .

There are quite a few missing samples in CONFINE datasets. Note that a missing sample does not mean that its LQ value equals zero. This may mean that the link is off or the software capturing LQ measurements did not work. If there are missing samples, two scenarios can be used:

- The first scenario would be to treat them as $LQ = 0$.
- The second scenario would avoid offering prediction results for the missing samples. This approach is used in this study.

Indeed, the physical layer can provide immediate information on the quality of the decoding of a packet. In this study, we did not deal with the physical layer of the WCN or simulate a model for the network. The operator of the Funkfeuer network provides only LQ measurements for a period of time. Also, the operator of the network represents the link quality

values in three decimal digits (e.g. 0.951). Given that, our method is a data-driven approach, we deal only with the collected LQ dataset and the available LQ data representation.

3. Experimental results and discussion

3.1. Dataset

The dataset used in this paper is taken from Funkfeuer Wien, which is a real-world WCN with around 500 nodes and 2000 links. Fig. 5 provides an interactive map for all nodes of the Funkfeuer Wien network [37]. In this map, the red points represent the active nodes while the black points refer to the inactive ones. In this network, a routing protocol derived from an open link state routing is used to enhance the scalability of the network. The topology information of the network provided by the routing protocol were gathered from each node of the network every 5 min. The dataset is publicly available in the Confine project.

In the Funkfeuer Wien dataset, a total of 998 links have variation in their quality while the remaining links have constant LQ values and so we do not use them in the training phase of the proposed approach. Note that, there are many links with constant quality, of which deriving predictions is

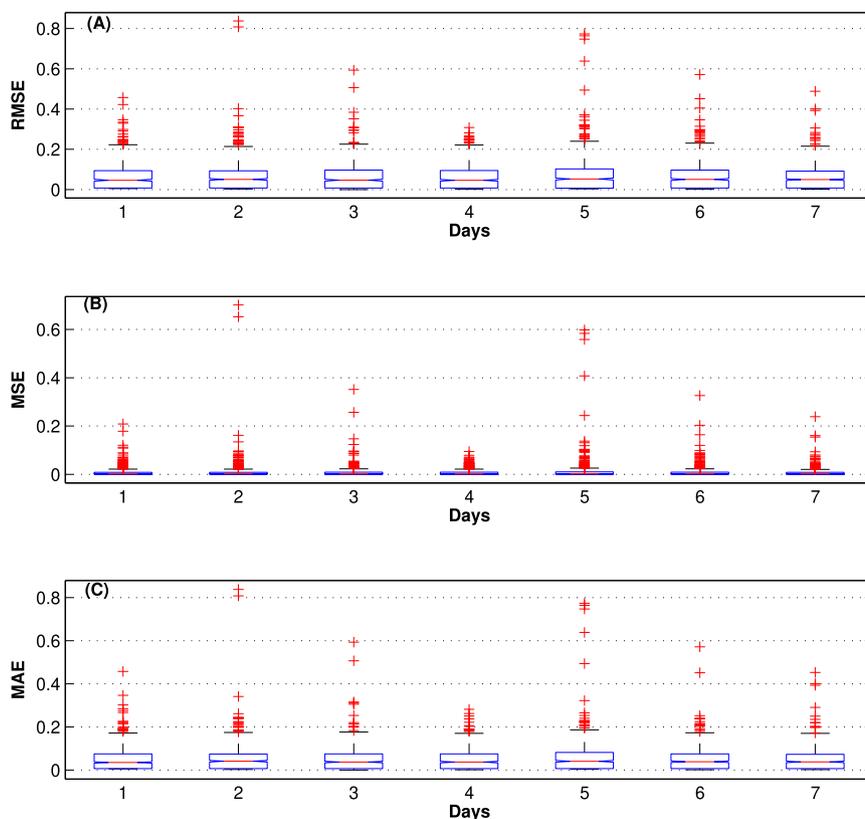


Fig. 6 Box plots of MSE, RMSE, and MAE of the proposed LSTM-RNN approach with the seven test days. Note that – refers to the median value of errors and + refers to an outlier.

trivial. Other publications that used this dataset also removed these links. Indeed, the LQ values of each link can be represented as a time-series because they were collected at regular time steps (5 min). The range of LQ values varies from 0 to 1, where 1 refers to a perfect link and 0 refers to a bad link.

Since there are related studies with different datasets derived from the CONFINE European project, we have uploaded the link quality dataset used in this study at <https://sites.google.com/site/lqdataset/1>.

3.2. Performance evaluation of LSTM-RNN and GRU

In our experiments, we have used the first day of the dataset to train the proposed LSTM-RNN and GRU models while one week (7 days) has been used to test the prediction accuracy. Note that we train one single model for all links. To demonstrate the efficacy of the proposed approach, we calculate the mean square error (MSE), root mean square error (RMSE), and mean absolute error (MAE) between the predicted LQ values and the actual ones. The formulae of these metrics can be expressed as follows:

$$MSE = \frac{1}{N} \sum_{i=1}^N (LQ_i^{Predicted} - LQ_i^{Actual})^2 \quad (18)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (|LQ_i^{Predicted}| - |LQ_i^{Actual}|)^2} \quad (19)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |LQ_i^{Predicted} - LQ_i^{Actual}| \quad (20)$$

where $LQ^{Predicted}$ is the predicted LQ values and LQ^{Actual} is the real LQ measurements. In general, there are two approaches to compute MAE over all the links and samples. The first approach is to compute the error for each link and then compute the average. The second approach would be to average the MAE for all samples. In this study, we use the second approach. To build the LSTM and GRU models, we exploit the sequential models of Keras library¹ (theano backend) and use the adaptive moment estimation (ADAM) optimizer [38]. The loss function of LSTM was the mean squared error. Both LSTM and GRU models are trained for a total of 100 epochs with a batch size of 1. The number of LSTM and GRU blocks, and the other super-parameters are experimentally tuned. To study the effect of the number of LSTM blocks on accuracy of LQ prediction, in Table 2 we show MSE, RMSE, and MAE per link for three configurations of the proposed LSTM-RNN approach (two, five, and seven LSTM blocks) for all links of all testing days. Note that we calculate MSE, RMSE and MAE for each day of the testing days individually and then average them for the seven days. The proposed approach with 5 LSTM blocks gives the smallest MSE (0.0065654) and MAE (0.0432292). The smallest RMSE is achieved with 2 LSTM blocks (0.0705911). Fig. 6 shows box plots of MSE, RMSE, and MAE per link with the seven test days (day 1 to day 7) for the best configuration of the LSTM-RNN approach (5 LSTM blocks). As shown, the LSTM-RNN approach gives small prediction errors with the seven test days.

¹ <https://keras.io/>.

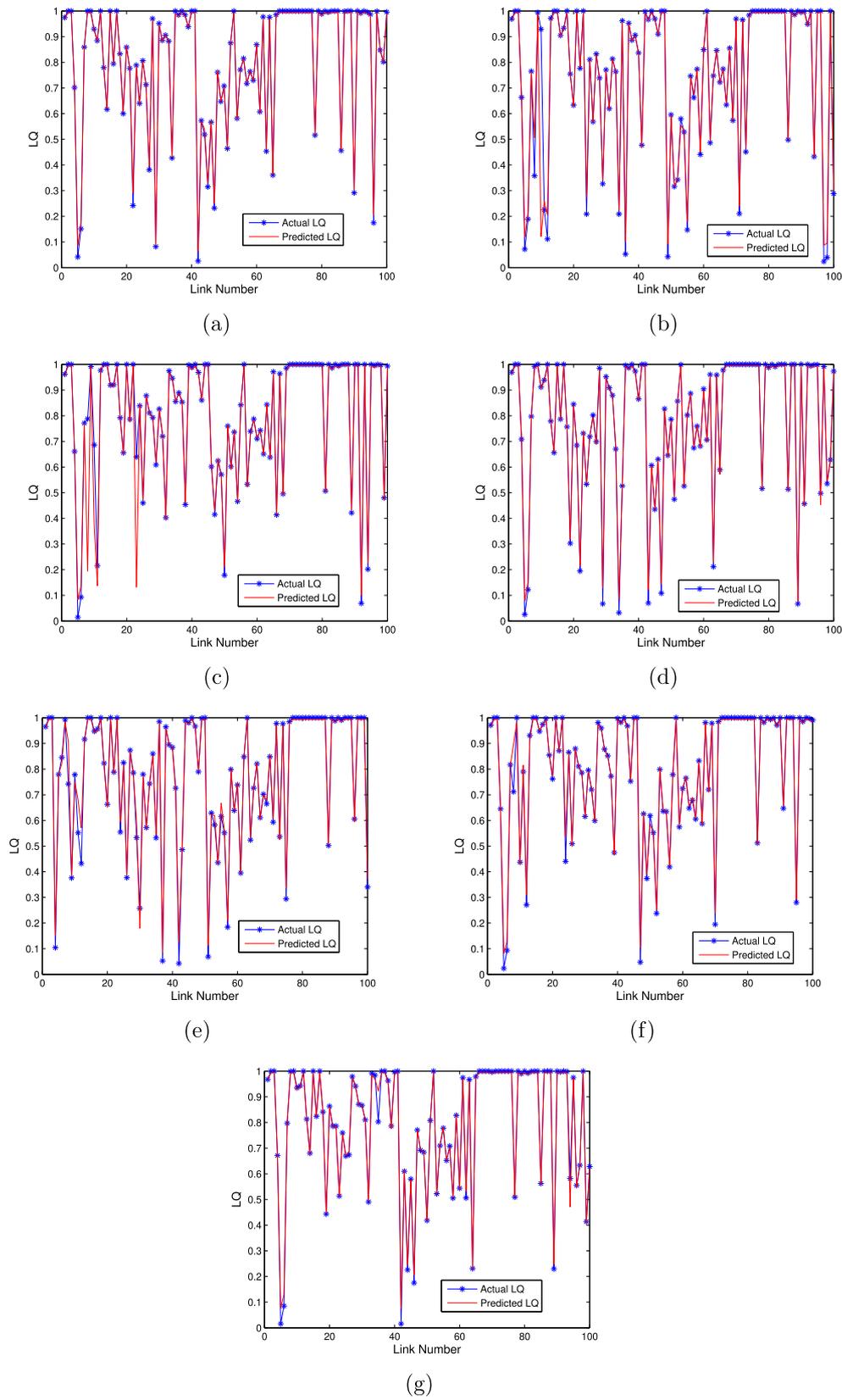


Fig. 7 The actual and predicted LQ values using the LSTM-RNN approach for (a) day1, (b) day2, (c) day3, (d) day4, (e) day5, (f) day6, and (g) day7.

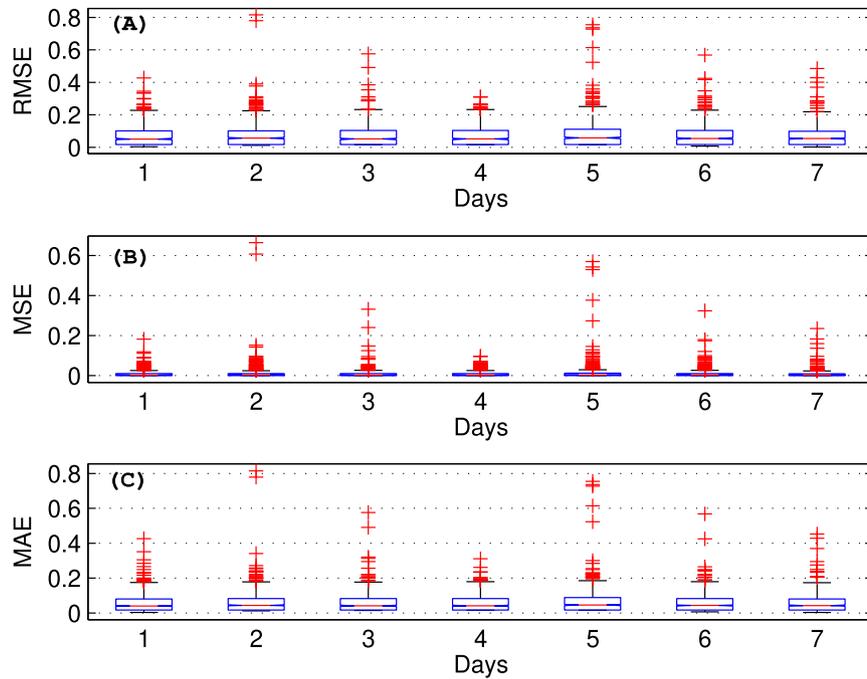


Fig. 8 Box plots of MSE, RMSE, and MAE of the GRU approach with the seven days. Note that – refers to the median value of errors and + refers to an outlier.

As the testing datasets contain thousands of samples, we cannot visualize the predicted LQ values of all links. For the sake of keeping the visualization simple, in Fig. 7 we show the predicted LQ values by the proposed approach and the actual ones for only 100 randomly-selected links from each testing day. It is clear that the predicted LQ values of the links are very close to the actual ones. It worth noting that despite the strong variance between the quality of the 100 links, the proposed approach can accurately predict their quality.

We have also assessed the performance of the GRU approach with the seven test days. In this study, we set the number of GRU blocks to 5 (this value has been experimentally tuned). The MSE, RMSE and MAE of the GRU approach are 0.0071437, 0.0844780 and 0.0502189, respectively. Fig. 8 presents box plots of MSE, RMSE, and MAE per link with the seven test days for the GRU approach. As shown, this approach also gives small prediction errors.

For the sake of keeping the visualization of GRU results simple, in Fig. 9, we show the predicted LQ by the GRU approach and the actual values of LQ for 100 randomly-selected links from each day of the test days. As we can see, the predicted LQ values are very close to the actual values. It is important to note that we compute performance metrics over over all links not over 100 randomly selected links.

Indeed, the architectures of LSTM and GRU approaches are comparable in terms of complexity. For instance, the number of weights in the utilized LSTM-RNN approach (5 blocks) and the GRU approach (5 blocks) are 146 and 111, respectively. The training times of LSTM and GRU networks are 50 min and 45 min, respectively. In turn, their inference times are 0.1075 s and 0.0846 s, respectively. To demonstrate the accuracy of the prediction approach, we show in Fig. 10 the actual values and predicted values by LSTM of a link for a day where they follow the same trend.

3.3. Comparisons and statistical analysis

In this subsection, we compare the performance of LSTM-RNN and GRU LQ prediction approaches with the traditional ANN method. In our experiments, various configurations of ANN have been examined. Besides, MSE, RMSE, and MAE, we use two additional metrics in this comparison: mean absolute percentage error (MAPE) and normalized root mean square error (NRMSE):

$$MAPE = 100 \frac{1}{N} \sum_{i=1}^N (||LQ_i^{Predicted}| - |LQ_i^{Actual}||) / LQ_i^{Actual} \quad (21)$$

$$NRMSE = \frac{1}{R} \sqrt{\frac{1}{N} \sum_{i=1}^N (|LQ_i^{Predicted}| - |LQ_i^{Actual}|)^2} \quad (22)$$

where $R = \max(LQ^{Actual}) - \min(LQ^{Actual})$.

As shown in Table 3, the prediction (MSE, RMSE, MAE, MAPE, and NRMSE) of the GRU approach are greater than that of LSTM-RNN approach, while ANN gives prediction errors much greater than the ones of LSTM-RNN and GRU approaches.

It is interesting to examine the statistical significance of LSTM-RNN and GRU approaches in terms of the predicted LQ values for the seven test days. The term statistical significance refers to the likelihood that a relationship between two or more variables is not caused by chance. Statistical hypothesis testing is used to determine whether the result of a data set is statistically significant. This test provides a p-value, representing the probability that random chance could explain the result. In general, a p-value of 0.05 or lower is considered to be statistically significant.

We used Welch's t-test to determine the difference in predicted LQ values (significance level < 0.05). The normality

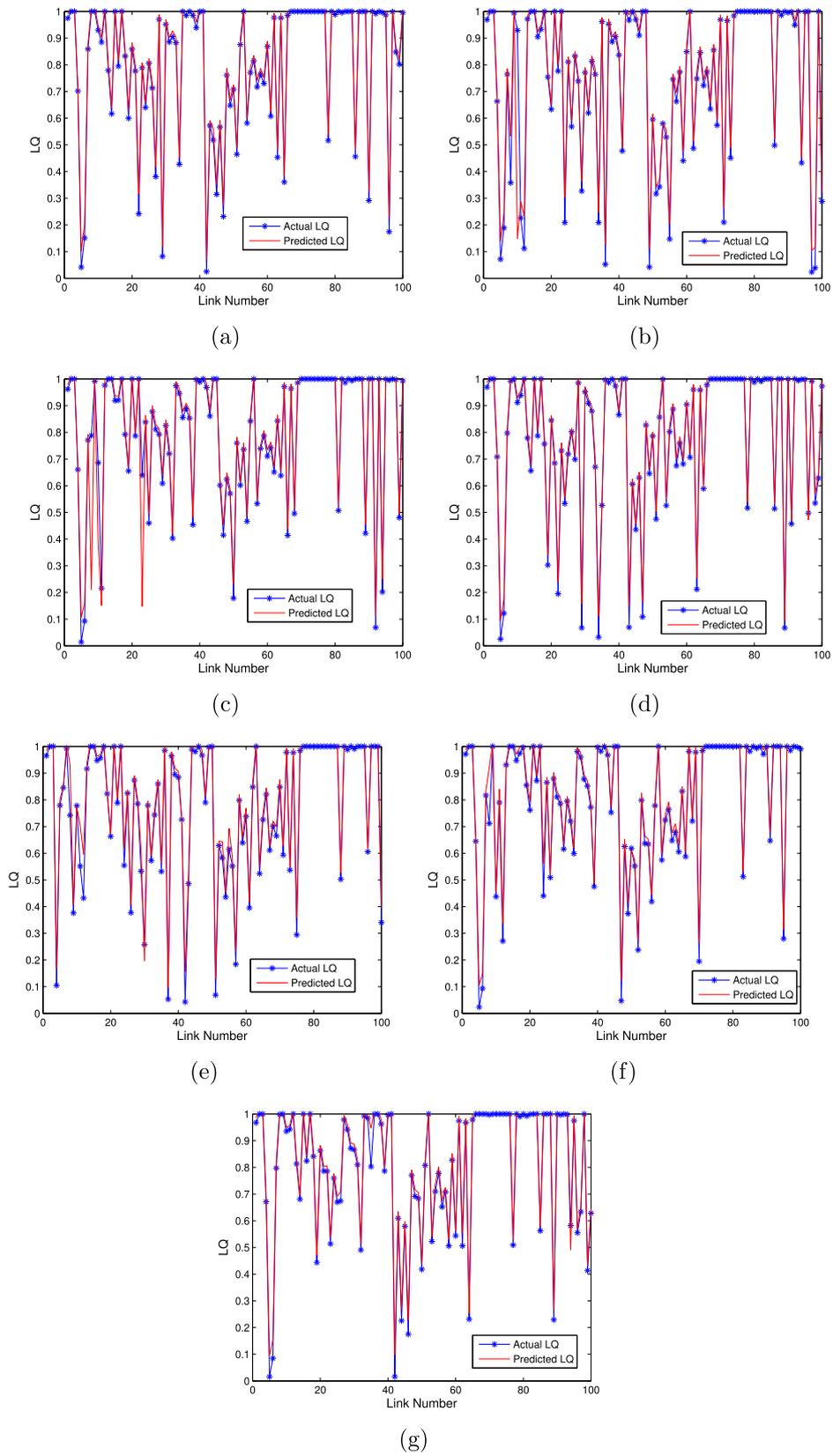


Fig. 9 The actual and predicted LQ values using the GRU approach for (a) day1, (b) day2, (c) day3, (d) day4, (e) day5, (f) day6, and (g) day7.

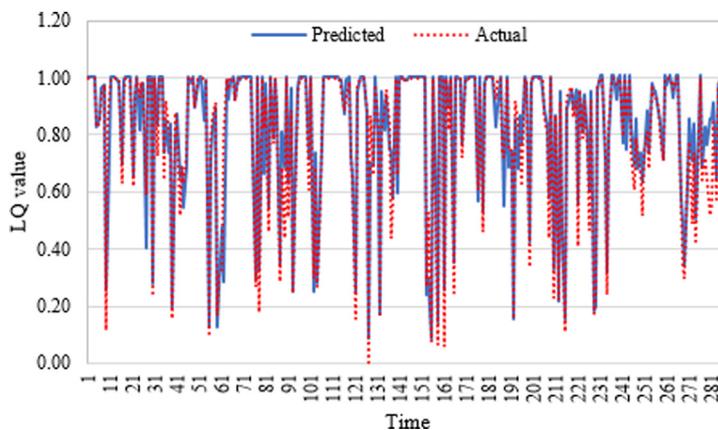


Fig. 10 The actual and predicted values by LSTM-RNN of a link for a day.

Table 3 Comparison between the proposed approach (LSTM-RNN and GRU) and traditional approach (ANN).

Proposed Approach	Average MSE \pm Std	Average RMSE \pm Std	Average MAE \pm Std	Average MAPE \pm Std	Average NRMSE \pm Std
ANN	0.10270 \pm 0.0644	0.32040 \pm 0.0017	0.18530 \pm 0.0577	18.8410 \pm 0.0357	0.3210 \pm 0.0279
GRU	0.0071437 \pm 0.00048053	0.0844780 \pm 0.00026415	0.0502189 \pm 0.0004164	9.6299 \pm 0.0457	0.0731 \pm 0.00566
LSTM-RNN	0.0065654 \pm 0.00071958	0.0809786 \pm 0.00051683	0.0432292 \pm 0.00066084	9.3692 \pm 0.0060	0.0642 \pm 0.0077

Table 4 The statistical significance of LSTM-RNN and GRU approaches (p -value) for the seven test days

Day	LSTM vs. GRU
1	0.328
2	0.217
3	0.690
4	0.427
5	0.473
6	0.421
7	0.115

of the distributions of the LQ values was assessed by means of bootstrapping and the Shapiro–Wilk test. We found that all the AUC values follow a normal distribution. Table 4 shows the statistical analysis of the predicted LQ values obtained by LSTM-RNN and GRU approaches. In this table, a p -value lower than 0.05 refers to statistical significance. As can be seen in Table 4, the predicted LQ values of LSTM-RNN and GRU are not statistically significant. In other words, there is a slight difference between the results of the LSTM and GRU models.

Since the performance of the LSTM-RNN approach superiors that of GRU and ANN approaches, in Fig. 11 we show the ranking of LQ values errors obtained by the LSTM-RNN approach with the first test day (day1, see Fig. 11 (a)) and the last test day (day7, see Fig. 11 (b)). In this figure, we reordered the predicted LQ values for the two days in an ascending order (from the lowest actual LQ value to the highest actual LQ value) to examine the prediction accuracy. As shown, the curves of the predicted values and the actual ones are very

close (except some outliers). In addition, the proposed LSTM-RNN approach yields higher prediction accuracy in the case of higher LQ values, and so the proposed approach is very helpful for accurately predicting LQ which can improve the performance of routing protocols of WCNs. Deep learning techniques offer better representation and prediction results on a multitude of time-series data compared with shallow approaches (e.g., auto-regressive networks and ANN) [39,40]. Thus, in this study, we focus only on RNNs and compared it with the traditional ANN.

It is worth noting that any network with less than the required Nyquist memory will not be optimal. In this work, we have not dealt with the acquisition step of LQ values since we used a publicly available LQ dataset (i.e. Funkfeuer dataset). Thus, we do not consider this point in our study. In the literature, the LSTM and GRU models have been applied to different time-series prediction problems with big dataset [24,30,36], and thus it is expected that the proposed approach will still be effective as the scale of data increases.

4. Conclusion and future work

In this paper, we have proposed a deep learning approach for predicting LQ in WCNs. The problem has been formulated as a time-series prediction problem, and two variants of deep RNN (LSTM-RNN and GRU) have been proposed. The proposed approach addresses the fluctuating nature of LQ values in WCNs because LSTM-RNN and GRU models can learn and exploit the context in LQ time-series. In our experiments, we have used a real dataset derived from the CONFINE European project. Indeed, the use of such publicly available dataset is a good choice concerning reproducibility. According to the experimental results, we can conclude the following key points:

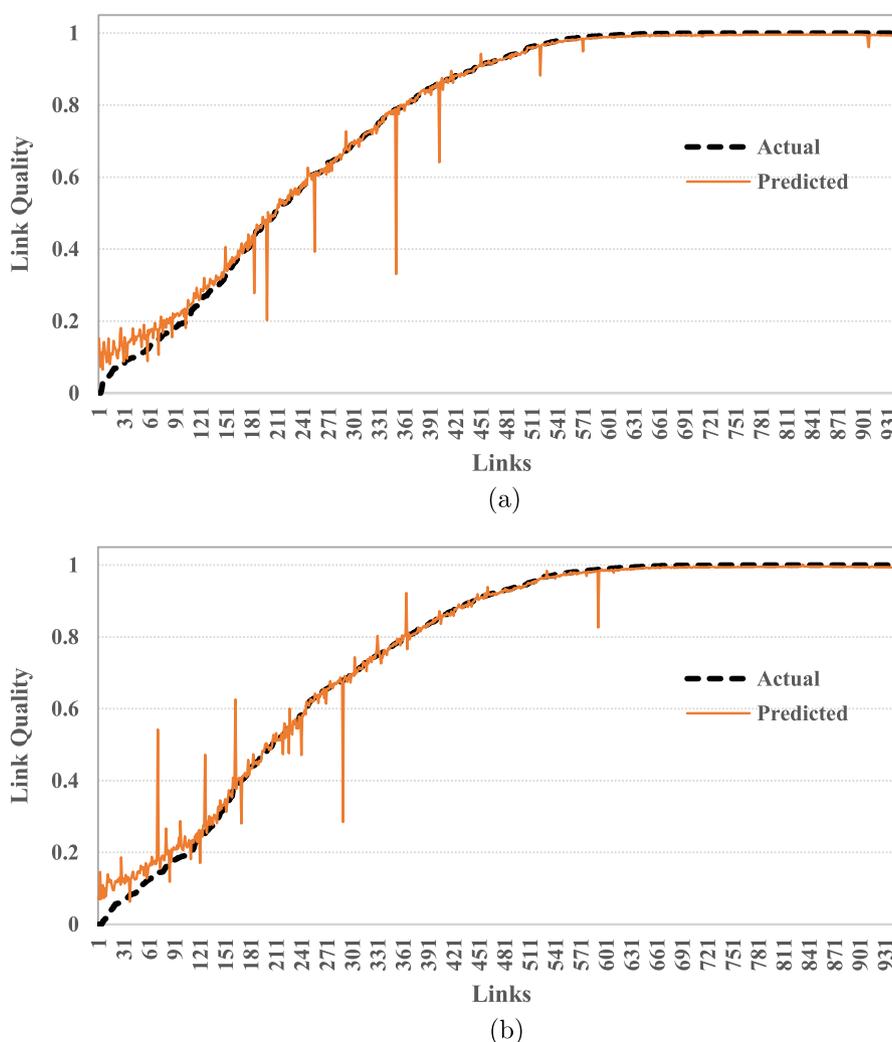


Fig. 11 The rank of actual and predicted LQ values for (a) day1, and (b) day7 based on LSTM-RNN.

- The proposed deep learning approaches achieve smaller prediction errors than that of the traditional ANN.
- The proposed LSTM-RNN approach gives RMSE, MSE and MAE values smaller than the ones of GRU for the test seven days.
- The statistical analysis has demonstrated that the predicted LQ values of LSTM-RNN are not statistically significant than the ones of GRU.

Therefore, the proposed deep LSTM-RNN and GRU models could be a very promising approach for LQ prediction. In general, the process of predicting LQ could raise the complexity of routing protocols because it may required adding hardware and/or software. However, if the LQ prediction approach is accurate, the overall performance of the routing protocols can be greatly enhanced. Since we have utilized deep learning models to predict LQ, their computational cost of training phase is a bit high as it includes forward and backward passes. Note that our models have been trained off-line while the LQ prediction can be performed on-line in a very short time (the time of forward pass only).

Future work includes different extensions of the current study such as: (1) the performance of the proposed approach

will be assessed using additional real-world datasets, (2) the wavelet decomposition will be used to further improve the prediction accuracy of LQ, (3) Bayesian approaches will be used to give robust uncertainty quantification in LQ predictions [41,42], and (4) we will investigate the improvement in the performance of various routing algorithms when integrated with the proposed approach.

Declaration of Competing Interest

The authors declare that they have no conflict of interest.

References

- [1] B. Braem, C. Blondia, C. Barz, H. Rogge, F. Freitag, L. Navarro, J. Bonicioli, S. Papathanasiou, P. Esrich, R. Baig Vinas, et al, A case for research with and on community networks, *ACM SIGCOMM Comput. Commun. Rev.* 43 (3) (2013) 68–73.
- [2] Funkfeuer, www.funkfeuer.at.
- [3] R. Baig, R. Roca, F. Freitag, L. Navarro, Guifi. net, a crowdsourced network infrastructure held in common, *Comput. Netw.* 90 (2015) 150–165.

- [4] L. Maccari, An analysis of the Ninux wireless community network, in: IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2013, pp. 1–7.
- [5] N. Baccour, A. Koubaa, L. Mottola, M.A. Zúñiga, H. Youssef, C.A. Boano, M. Alves, Radio link quality estimation in wireless sensor networks: A survey, *ACM Trans. Sensor Netw. (TOSN)* 8 (4) (2012) 34.
- [6] D.S. De Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, *Wireless Netw.* 11 (4) (2005) 419–434.
- [7] Y. Tsado, K. Gamage, B. Adebisi, D. Lund, K. Rabie, A. Ikpehai, Improving the reliability of optimised link state routing in a smart grid neighbour area network based wireless mesh network using multiple metrics, *Energies* 10 (3) (2017) 287.
- [8] C. Renner, S. Ernst, C. Weyer, V. Turau, Prediction accuracy of link-quality estimators, in: European Conference on Wireless Sensor Networks, Springer, 2011, pp. 1–16.
- [9] V. Kolar, S. Razak, P. Mähönen, N.B. Abu-Ghazaleh, Link quality analysis and measurement in wireless mesh networks, *Ad Hoc Netw.* 9 (8) (2011) 1430–1447.
- [10] J. Zhou, M. Jacobsson, E. Onur, I. Niemegeers, An investigation of link quality assessment for mobile multi-hop and multi-rate wireless networks, *Wireless Personal Commun.* 65 (2) (2012) 405–423.
- [11] C.J. Lowrance, A.P. Lauf, Link quality estimation in ad hoc and mesh networks: a survey and future directions, *Wireless Pers. Commun.* 96 (1) (2017) 475–508.
- [12] T. Liu, A.E. Cerpa, TALENT: temporal adaptive link estimator with no training, in: Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, ACM, 2012, pp. 253–266.
- [13] K. Farkas, T. Hossmann, F. Legendre, B. Plattner, S.K. Das, Link quality prediction in mesh networks, *Comput. Commun.* 31 (8) (2008) 1497–1512.
- [14] M.L. Bote-Lorenzo, E. Gómez-Sánchez, C. Mediavilla-Pastor, J.I. Asensio-Pérez, Online machine learning algorithms to predict link quality in community wireless mesh networks, *Comput. Netw.* 132 (2018) 68–80.
- [15] P. Millan, C. Molina, E. Medina, D. Vega, R. Meseguer, B. Braem, C. Blondia, Time series analysis to predict link quality of wireless community networks, *Comput. Netw.* 93 (2015) 342–358.
- [16] P. Millan, C. Aliagas, C. Molina, E. Dimogerontakis, R. Meseguer, Time series analysis to predict end-to-end quality of wireless community networks, *Electronics* 8 (5) (2019) 578.
- [17] J. Shu, Q. Chen, L. Liu, L. Xu, A link prediction approach based on deep learning for opportunistic sensor network, *Int. J. Distrib. Sens. Netw.* 13 (4) (2017), 1550147717700642.
- [18] T. Liu, A.E. Cerpa, Foresee (4C): Wireless link prediction using link features, in: Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, IEEE, 2011, pp. 294–305.
- [19] W. Sun, W. Lu, Q. Li, L. Chen, D. Mu, X. Yuan, WNN-LQE: Wavelet-neural-network-based link quality estimation for smart grid WSNs, *IEEE Access* 5 (2017) 12788–12797.
- [20] T. Liu, A.E. Cerpa, Temporal adaptive link quality prediction with online learning, *ACM Trans. Sensor Netw. (TOSN)* 10 (3) (2014) 46.
- [21] L. Weng, P. Zhang, Z. Feng, H. Cheng, H. Lian, B. Fu, Short-term link quality prediction using nonparametric time series analysis, *Sci. China Inf. Sci.* 58 (8) (2015) 1–15.
- [22] C. Jia, L. Liu, X. Gu, M. Liu, A novel link quality prediction algorithm for wireless sensor networks, *Comput. Sci. Inf. Syst.* 14 (3) (2017).
- [23] X. Li, L. Yang, Y. Li, C. Zhou, Deep trajectory: a deep learning approach for mobile advertising in vehicular networks, *Neural Comput. Appl.* (2019) 1–13.
- [24] M. Abdel-Nasser, K. Mahmoud, Accurate photovoltaic power forecasting models using deep LSTM-RNN, *Neural Comput. Appl.* (2017) 1–14.
- [25] H. Ye, G.Y. Li, B.-H. Juang, Power of deep learning for channel estimation and signal detection in OFDM systems, *IEEE Wireless Commun. Lett.* 7 (1) (2018) 114–117.
- [26] K. Mahmoud, M. Abdel-Nasser, Fast-yet-accurate energy loss assessment approach for analyzing/sizing PV in distribution systems using machine learning, *IEEE Trans. Sustain. Energy.*
- [27] X. Wang, L. Gao, S. Mao, PhaseFi: Phase fingerprinting for indoor localization with a deep learning approach, in: Global Communications Conference (GLOBECOM), 2015 IEEE, IEEE, 2015, pp. 1–6.
- [28] M. Kim, W. Lee, D.-H. Cho, A novel PAPR reduction scheme for OFDM system based on deep learning, *IEEE Commun. Lett.* 22 (3) (2018) 510–513.
- [29] C. Lu, L. Peng, J. Bi, H. Yuan, An improved dro-based recurrent neural networks for large-scale light curve time series prediction, in: 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), IEEE, 2018, pp. 117–121.
- [30] J. Bi, S. Li, H. Yuan, Z. Zhao, H. Liu, Deep neural networks for predicting task time series in cloud computing systems, in: 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC), IEEE, 2019, pp. 86–91.
- [31] J. Bi, H. Yuan, M. Zhou, Temporal prediction of multiapplication consolidated workloads in distributed clouds, *IEEE Trans. Autom. Sci. Eng.* 16 (4) (2019) 1763–1773.
- [32] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112.
- [33] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in: International Conference on Machine Learning, 2013, pp. 1310–1318.
- [34] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, *IEEE Trans. Neural Netw.* 5 (2) (1994) 157–166.
- [35] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780.
- [36] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555.
- [37] Funkfeuer interactive map, <https://map.funkfeuer.at/wien/> accessed: 2018-03-21.
- [38] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.
- [39] M. Längkvist, L. Karlsson, A. Loutfi, A review of unsupervised feature learning and deep learning for time-series modeling, *Pattern Recogn. Lett.* 42 (2014) 11–24.
- [40] R. Fu, Z. Zhang, L. Li, Using LSTM and GRU neural network methods for traffic flow prediction, in: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), IEEE, 2016, pp. 324–328.
- [41] R. Chandra, K. Jain, R.V. Deo, S. Cripps, Langevin-gradient parallel tempering for Bayesian neural learning, *Neurocomputing* 359 (2019) 315–326.
- [42] R. Chandra, Y.-S. Ong, C.-K. Goh, Co-evolutionary multi-task learning for dynamic time series prediction, *Appl. Soft Comput.* 70 (2018) 576–589.