Time Series Analysis to Predict Link Quality of Wireless Community Networks

Pere Millan, Carlos Molina, Dept. of Computer Engineering, Universitat Rovira i Virgili, Tarragona, Spain
{pere.millan, carlos.molina}@urv.net

Esunly Medina, Davide Vega, Roc Meseguer, Dept. of Computer Architecture, Universitat Politècnica de Catalunya, Barcelona, Spain, {esunlyma, dvega, meseguer}@ac.upc.edu

Bart Braem, Chris Blondia, Dept. of Mathematics-Computer Sciences, University of Antwerp - iMinds, Antwerpen, Belgium, {bart.braem, chris.blondia}@uantwerpen.be

Abstract: *Community networks have emerged under the mottos "break the strings that are limiting you", "don't buy the network, be the network" or "a free net for everyone is possible". Such networks create a measurable social impact as they provide to the community the right and opportunity of communication. As any other network that mixes wired and wireless links, the routing protocol must face several challenges that arise from the unreliable nature of the wireless medium. Link quality tracking helps the routing layer to select links that maximize the delivery rate and minimize traffic congestion. Moreover, link quality prediction has proved to be a technique that surpasses link quality tracking by foreseeing which links are more likely to change its quality. In this work, we focus on link quality prediction by means of a time series analysis. We apply this prediction technique in the routing layer of large-scale, distributed, and decentralized networks. We demonstrate that it is possible to accurately predict the Link Quality in 98% of the instances, both in the short and the long terms. Particularly, we analyse the behaviour of the links globally to identify the best prediction algorithm and metric, the impact of lag windows in the results, the prediction accuracy some time steps ahead into the future, the degradation of prediction over time, and the correlation of prediction with topological features. Moreover, we also analyse the behaviour of links individually to identify the variability of link quality prediction between links, and the variability of link quality prediction over time. Finally, we also present an optimized prediction method that considers the knowledge about the expected link quality values.*

Keywords: Community Networks, Link Quality Tracking, Link Quality Prediction, Time Series Analysis.

I. Introduction

Community networks are distributed, large-scale and decentralized networking infrastructures composed of several nodes, links and services where the resources are made available to a group of people living in the same area. Networks of this kind are extremely diverse and dynamic because they are composed of decentralized nodes and mix wired and wireless technologies, several routing schemes and diverse services and applications [1]. The network is managed using an open peering agreement, which avoids barriers for the participation in the network. Governance, knowledge and ownership of the network are open. These networks are, therefore, not only decentralized but also owned and managed by community members. In addition, they grow dynamically with regards to the number of links, their capacity and services provided. Some relevant examples of community networks include guifi.net [2] or FunkFeuer [3].

These large, decentralized, dynamic and heterogeneous structures raise challenges that can be of interest to researchers, both as a source of inspiration and as a field to apply their research findings [4].

1

One of the most important challenges is the effect of the unreliability and asymmetrical characteristics of wireless communications on routing protocols and network performance. Many metric-based routing protocols for mesh networks that track link quality (LQ) and select higher-quality links have been proposed to maximize delivery rate and minimize traffic congestion [5], [6], [7]. Hence, LQ tracking is a key method to be applied when routing packets through an unreliable network. Moreover, it has been shown that routing algorithms should avoid weak links whenever possible [8], and as soon as possible [9].

LQ estimation (or prediction) [10], [11] is an approach that increases the improvements in routing performance achieved through LQ tracking. Typically, real-time metrics do not provide enough information to detect the degradation or activation of a link at the right moment. Therefore, prediction techniques are needed to foresee LQ changes in advance and take the appropriate measures.

In this work, we present a LQ analysis and prediction of the FunkFeuer wireless mesh community network [3]. To the best of our knowledge, no previous work explores LQ prediction in the routing layer of large-scale, distributed and decentralized systems. The main contributions of this work are the following:

- The use of time series analysis to estimate LQ in the routing layer for real-world wireless mesh community networks.
- Clear evidence that LQ values computed through time series algorithms can make accurate predictions in wireless mesh community networks.
- A detailed global analysis of links to identify the best prediction algorithm and metric, the impact of lag windows on it, its accuracy some time steps ahead into the future, its degradation over time and its correlation with topological features, showing the potential of time series to estimate LQ.
- A detailed analysis of individual links to identify the variability of LQ prediction between links and over time, showing the potential of time series to estimate LQ.
- A hybrid prediction approach that combines a number of prediction algorithms along with a value saturating technique.

This paper is structured as follows. Section 2 gives an overview of prediction in computer networks focusing on link quality. Section 3 presents the experimental methodology used in this work. Section 4 and Section 5 describe our proposal and analyse the results globally and locally, respectively. Finally, in Section 6 we provide some concluding remarks and future work.

II. Link Quality Prediction in Wireless Networks

LQ tracking has been previously applied in several scenarios in different ways [5], [6], [7], [8], [12] to select higher quality links that maximize delivery rate and minimize traffic congestion. LQ prediction/estimation is used in addition to LQ tracking to determine beforehand which links are more likely to change their behaviour. As a result, the routing layer can make better decisions at the appropriate moment. LQE (Link Quality Estimators) [10], [11] are in charge of measuring the quality of the links between nodes based on physical or logical metrics. Physical metrics focus on the received signal quality and logical metrics focus on the percentage of lost packets. LQE with metrics like LQI (Link Quality Indication) [13], SNR (Signal-to-Noise Ratio) [14] or RSSI (Received Signal Strength Indication) [15] fit in the former category, whereas metrics like RNP (Required Number of Packets) [16], ETX (Expected Transmission Count) [17], [18] or PSR (Packet Success Rate) [5] fit in the latter. It is also important to notice, as is shown in [10], that a slight variation in SNR may affect the link quality, changing it from good to bad in a bursty way. All these metrics can be used by LQE in isolation or as a combination of some of them [10], [11], [19] to select the more suitable neighbour nodes when making routing decisions.

The Four-Bit Wireless Link Estimation method [13] collects information of several layers (physical, link and network) to make LQ predictions that can be decoupled from particular layer implementations while remaining efficient and accurate. A Kalman Filter approach [15] estimates the received signal strength (RSS), then takes this value to evaluate the SNR and finally approximates the packet success rate (PSR) in a way that can adapt fast enough to the temporal dynamics of the links. There are some other studies [14] that conclude that including SNR information significantly improves wireless LQ prediction. The Holistic Packet Statistics (HoPS) [19] approach applies EWMA (Exponentially Weighted Moving Average) filters to calculate several metrics that help evaluate the short-term and long-term quality link, its variation and evolution over time. On the other hand, the WMEWMA estimator (Window Mean with Exponentially Weighted Moving Average) [5] predicts LQ of sensor networks based on the Packet Success Rate (PSR) in a simple, memory efficient and fast reacting way.

The analysis of the most important network properties for the design of efficient routing protocols leads to the conclusion that the required number of packets (RNP) that must be sent before a packet is received is a precise LQ estimator [16]. Based on this metric, distributed and centralized routing algorithms are proposed to improve efficiency of networks with low power links. ETX has also been proved to be a good metric for LQ estimation [17], particularly with larger networks with longer paths. For instance, when the DSDV and DSR routing algorithms are adapted to include the ETX metric, the overall performance of the network is significantly improved. The ETX_ANT algorithm [18] is a simple approach that predicts ETX values a few seconds in advance. This estimation assumes a linear regression method, which is applied to a variable number of last ETX measurements.

Special attention must be paid to the MetricMap mechanism [20]. MetricMap is fundamentally a routing protocol for wireless sensor networks that uses a learning-enabled method for LQ assessment. Based on the observation that high traffic rates make tracking link qualities more difficult, this protocol uses prediction methods to estimate them in advance. In a first stage, a machine-learning algorithm is applied to classify link qualities. Two types of classifiers are evaluated: a decision tree and a rule-based classifier. The data used to train both classifiers was preclassified offline based on a LQ indicator and other metrics that represent some features of the nodes. In a second stage, the MetricMap routing protocol estimates the LQ at runtime by replacing the current traffic information with the rules collected offline from the classifiers. Results show that MetricMap can achieve a significant improvement on the data delivery rate in high traffic rate applications. MetricMap is the most similar work to the one presented in this paper as it uses time series analysis to improve the routing protocol, however, there are some significant differences:

- They evaluate a small wireless sensor network whereas we evaluate a large wireless mesh community network.
- They give only a hint of the potential of time series analysis to predict LQ. In contrast, we perform a detailed and deep analysis of this potential.
- They apply a time series analysis to predict current LQ values while we use a time series to predict future LQ values.
- They use a cross-validation method, which uses a subset of the sample data to validate the LQ estimation. We, on the other hand, use new data to validate the LQ estimation.

III. Experimental Methodology

A. FunkFeuer Network and Open Data Set

FunkFeuer [3] is a free experimental Wireless Mesh Community Network deployed in several locations in Austria. This network is a non-commercial project maintained by computer enthusiasts that install Wi-Fi antennas across rooftops. Currently, there are around 2000 wired and wireless links and every week new nodes are added to the network. It uses the OLSR-NG routing protocol, which expands the capabilities of the OLSR protocol and makes it highly scalable. In fact, some members of the FunkFeuer network are actively involved in the olsr.org open source project as developers, testing the protocol in the network.

We used open data sets from the FunkFeuer network, available through the CONFINE Project open data platform [21]. The data set is composed of OLSR information such as routing tables and network topology data of 404 nodes with 2095 links, collected during 7 days in the period from April 28th to May 4th, 2014. The data set instances were sampled every 5 minutes. It is important to notice that the total number of nodes is large. Every node has about 3.5 neighbours on average (degree) and that the largest of the shortest paths in the network (diameter) is 16. This means that there are several paths where packets have to go through a relatively high number of hops in order to reach their destination. The routing protocol must, therefore, react quickly to any change in the network topology since this will be critical to achieve high performance.

In this work, we focus on link quality prediction for the FunkFeuer network by means of a time series analysis. A recent study [22] has analysed the FunkFeuer network focusing on link layer properties, topological patterns and routing performance.

B. Link Quality

ETX [17] is a link metric that measures the expected number of data transmissions required to send a packet over that link, and is widely used in several mesh network protocols. The ETX of a particular link is calculated as: ETX = 1 / (LQ × NLQ), where LQ and NLQ stand for the "Link Quality" and the "Neighbour Link Quality" of that link, respectively. The Optimized Link State Routing (OLSR) protocol uses the ETX to choose, for each device and packet, the next hop. The LQ assumed by OLSR is defined as the fraction of successful packets (HELLO) that were received by a node from a given neighbour within a certain time window, while the NLQ is the fraction of successful packets that were received by the neighbour within a time period. We focus on predicting the LQ, as the NLQ is directly derived from the neighbour-node LQ and the ETX can be easily calculated using both predictions.
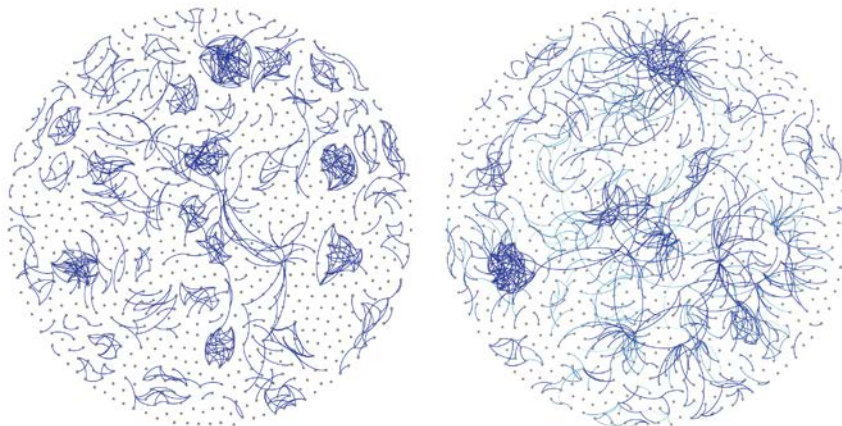

Fig. 1.   (left) Links with maximum link quality (LQ=1), and (right) Links with LQ<1.

Figure 1 illustrates the nodes and links from the open data set scenario used. On the left side, all the nodes are represented, but only links whose quality is always equal to one (maximum quality) are drawn. On the right side of Figure 1, all nodes and also those links that have experienced some variations in the link quality (LQ<1) at any instant over time are represented. For our simulations we only considered the latter links, which represent approximately half of the overall number of links (1068 of 2095). Moreover, we also discarded those links that did not have enough samples to perform the time series analysis. Therefore, our study only considered 1032 links. Notice that the prediction results will only be given for those links that present variations in the LQ (LQ<1). If we considered all links nodes in the network we would have obtained higher prediction accuracy; however, predicting the behaviour of links with unaltered LQ is trivial. For that reason, we have excluded these links from our study.

C. Time Series Analysis

A time series is a set of data collected over time with a natural temporal ordering. It differs from typical data mining or machine learning applications where the ordering of data points within a data set is not important. Time series analysis is the process of using statistical techniques to model and explain a time-dependent series of data points. Similarly, time series forecasting is a method that uses a model to generate predictions (forecasts) of future events based on known past events. In our case, we used more than one prediction algorithm so that we do not rely on a specific learning technique. We applied four of the best well-known approaches [23]: Support Vector Machines (SVM), k-Nearest Neighbours (kNN), Regression Trees (RT) and Gaussian Processes for Regression (GPR).

The Support Vector Machines (SVM) algorithm has recently become one of the most popular and widely used methods in machine learning. It performs a linear or nonlinear division of the input space and builds a prediction model that assigns target values into one or another category. The k-Nearest Neighbours (kNN) algorithm is one of the most simple machine learning algorithms as it makes no assumptions on the underlying data distribution. This algorithm takes the k data-points closest to the target value and picks the most common one. Regression Tree (RT) is a type of decision tree algorithm where the target value can have continuous values. This method recursively partitions the data space and runs a simple prediction model within each partition. The main advantages of tree algorithms are that (1) they produce fast results, and (2) they are resistant to irrelevant values. Finally, the Gaussian Processes for Regression (GPR) algorithm is a very flexible approach that can easily deal with complex data sets. In this model the output is a normal distribution denoted by the mean and variance terms. The target value is represented by this mean value and the variance can be interpreted as a measure of its confidence.

We applied training and test sets validation to evaluate the predictive accuracy of the models. After a model is processed using the training set, it is tested by making predictions against the test set. For this purpose, we used the Weka workbench system [24], a framework that incorporates a variety of learning algorithms and some tools for the evaluation and comparison of the results. Weka has a dedicated environment for time series analysis that allows forecasting models to be developed and evaluated. The Weka's time series framework takes a machine learning or data mining approach to model time series by transforming the data into a form that can be processed by standard propositional learning algorithms. To do so, it removes the temporal ordering of individual inputs by encoding the time dependency via additional input fields. These fields are sometimes referred to as "lagged" variables.

D. Metrics, Plots, and Topological Features

Usually, classification studies assess the predictive power of their model by using Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE), both widely used in related work. MAE is a common method

to evaluate the performance of prediction approaches that gives the same weight to all individual differences. This metric is calculated through the following formula: MAE = sum(abs(predicted - actual)) / N. RSME is another frequently used measure, but gives a relatively high weight to large errors. In this case, the formula is the following: RMSE = sqrt(sum((predicted - actual)^2) / N)

Boxplots are classic representations of a statistical distribution of values. A box is drawn around the region between the first and third quartile, and a horizontal line at the median value. Whiskers extend from the box to the lowest and highest value within 1.5 times the interquartile range of the lower and upper quartile, respectively. Data points that lie outside these limits are independently drawn.

The analysis of topological features of a network (degree, strength, distance, betweenness, etc.) may help to understand its behaviour, and therefore, to improve the efficiency of transferring information through nodes and links. In this work, we focus on node degree and edge betweenness (in terms of hopcount) to correlate them with the LQ. The former defines the number of links/edges connected to a particular node. The latter defines the number of shortest paths that go through one particular link or edge.

IV. Global Analysis of Links and Nodes

The first step of our analysis considers all links and nodes globally. This will allow us to get an overview of the behaviour of the nodes with LQ variations. In subsequent sections, we will analyse them individually, in order to identify different link classes according to their individual behaviour.

A. Comparison of learning algorithms based on time series

The main aim of this work is to explore whether time series analysis and prediction can be used to predict the next link-quality value. As stated before, we applied four well-known approaches: Support Vector Machines (SVM), k-Nearest Neighbours (kNN), Regression Trees (RT), and Gaussian Processes for Regression (GPR).
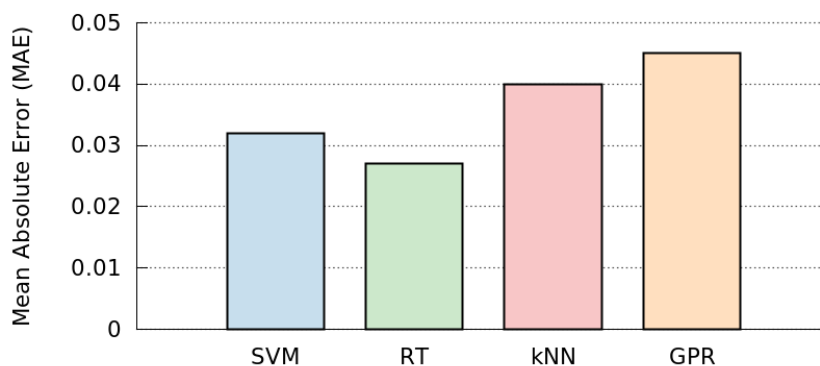

Fig. 2.   Average Mean Absolute Error (MAE) of the links.

Figure 2 shows the average Mean Absolute Error (MAE) per link using a training data set of 1728 instances (6 days, 1 instance every 5 minutes), a test data set of 288 instances (1 day) and a lag window composed of the last 12 instances. This test was performed to verify whether time series learning algorithms could predict consecutive LQ values. These results show that we achieved the best accuracy for the Regression Tree (RT) and the worst for Gaussian Processes for Regression (GPR). Notice that the maximum LQ value is 1 and therefore, the MAE per link is 2.7% for RT and 4.5% for GPR. We applied a T-test to mean values for independent samples (at 95% confidence level) in order to compare the

classification algorithms using the MAE. After this analysis, p-values smaller than 0.05 indicate that the means are significantly different, and therefore, we would reject the null hypothesis of no difference between the means. Consequently, we can claim that RT is a good candidate to make predictions.

We analysed the error variability of each algorithm and represented the results using boxplots. The four algorithms achieved a similar performance for most of the links, as shown in Figure 3 (both MAE and RMSE). Although the median, first quartile, and third quartile values are similar for all of them, there are some outliers with large errors. These outliers increase the average values and change the overall evaluation of the algorithms. Comparing MAE and RMSE results, we can observe that the behaviour for the four algorithms is similar using MAE and RMSE, although RMSE values are higher. For brevity reasons, we select MAE as the comparative metric, unless in any particular study the RMSE results were very different.
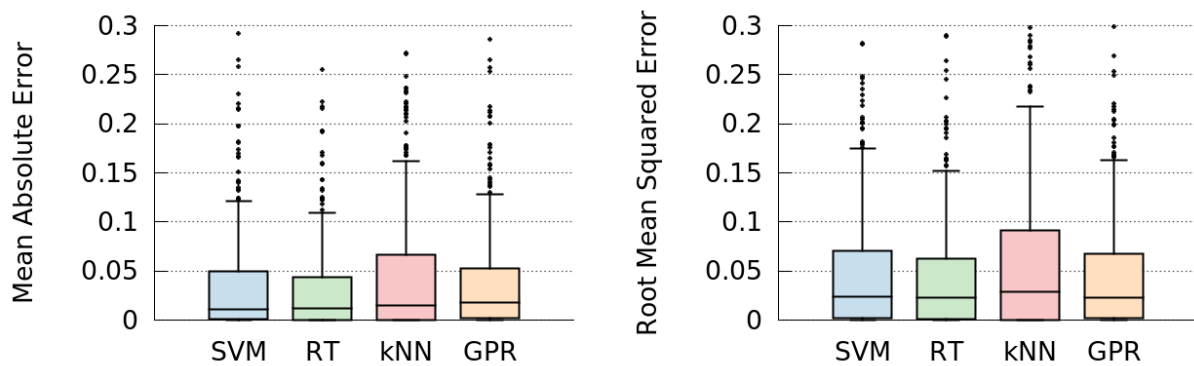

Fig. 3.   (left) Mean Absolute Error (MAE), and (right) Root Mean Squared Error (RMSE) of the LQ predictions, both as boxplots.

B.   Analysis of the Impact of the Lag Window Size

Lagged variables are the main mechanism by which we can capture relationships between past and current values of a series using propositional learning-algorithms. They create a "window" or "snapshot" over a time period. Basically, the number of lagged variables determines the window size (i.e. the number of previous samples used to make a new prediction, with samples every 5 minutes).
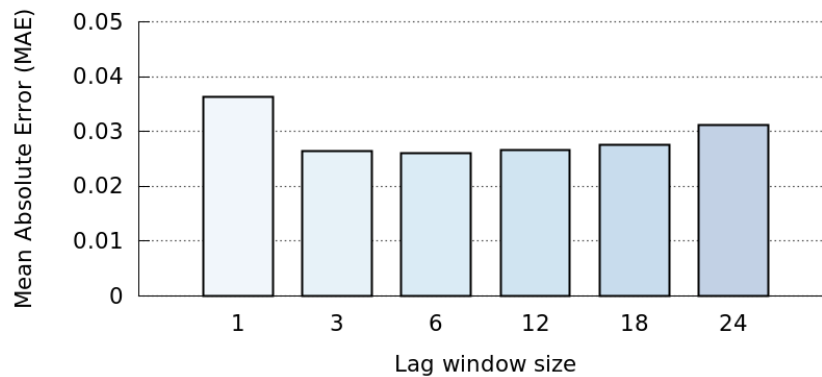

Fig. 4.   RT average Mean Absolute Error of the LQ predictions.

This analysis was performed to check the impact of the lag window in the prediction of the next link-quality value. Figure 4 shows the average MAE per link of the RT algorithm using the same experimental setup as in the previous test (1728 and 288 instances for training and testing, respectively) but in this experiment we used a lag window size ranging from 1 to 24 instances. We obtained good results using window sizes between 3 and 18 (Figure 4). The worst results were obtained for window sizes of 1 and 24. Nevertheless, these results are similar or even better than the results obtained by the other algorithms. Thus, we can sustain the claim that RT is the best candidate.

We also analysed the error variability for each window size and concluded that all window sizes achieved a similar performance for most of the links. Although the values for the median and the first quartile are similar for all window sizes, the values of the third quartile and the outliers slightly differ. These differences in the variability of errors lead to the differences in the average MAE. Finally, we tried to find the best lag window size for mean values using the T-test for independent samples (at 95% confidence level). After this analysis, we could not reject the null hypothesis at 95% of significance. Consequently, our results do not provide clear evidence about what is the best window size.

C. Prediction of Some Steps Ahead

This analysis was performed to explore if time series analysis and prediction can be used to predict the value of LQ several time steps ahead into the future.

Figure 5 shows the average MAE of links. It shows the results of the RT algorithm using the same setup that the baseline experiment (a lag window size of 12 instances, a training dataset of 1728 instances and a test dataset of 288 instances) and then predicting from 1 to 8 time steps into the future. The obtained results were good for all the tests. As we can observe, the average MAE grows very slowly. It seems possible to affirm that we could predict successfully the LQ several steps ahead in time. We also analysed the variability of errors for each number of steps ahead. Although the values for the median and the first quartile are similar for all steps ahead considered, the values of third quartile and outliers grow with the number of steps. These differences in the variability of errors lead to the differences in the average MAE.
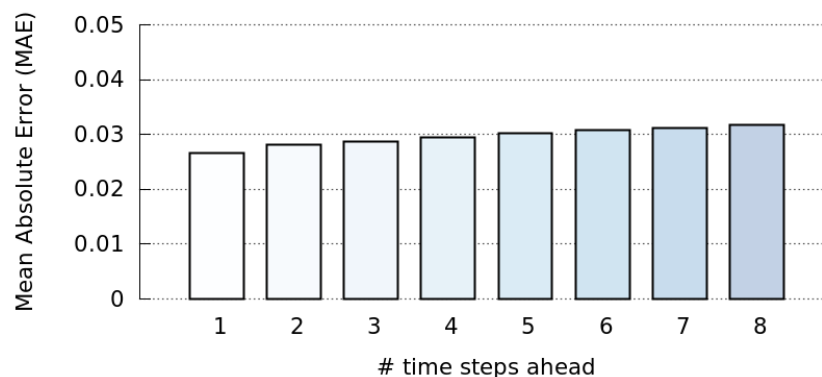

Fig. 5.   RT average Mean Absolute Error of the LQ predictions.

D. Degradation of the Regression-Tree Model over Time

This experiment was performed to evaluate the accuracy of the prediction models over time. Figure 6 shows the average MAE of the overall network and its approximation to a linear function. It shows the results of the RT algorithm using the same setup as the baseline experiment (a lag window size of 12 instances and a training dataset of 288 instances) but using a test dataset ranging from 144 (½ day) to 1728 (6 days) instances. We used linear regression to compute the parameters and estimate the goodness-of-fit, obtaining these parameters: slope = 0.0212 and b = 0.0132 (depicted by a line in Figure 6). Clearly, we can affirm that a linear function can be used to model the degradation of the RT over time.
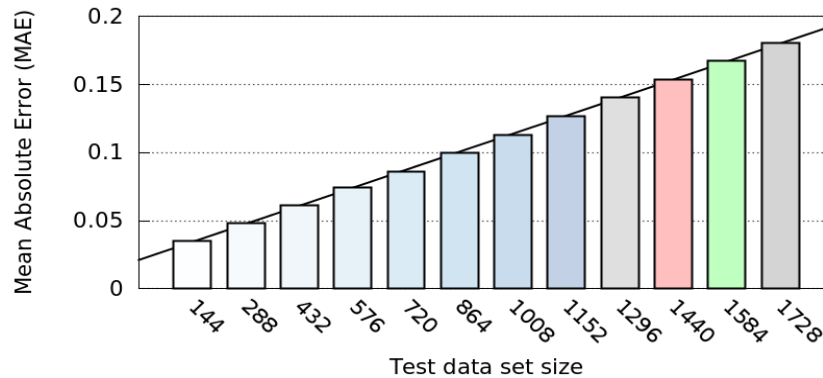

Fig. 6.   RT Mean Absolute Error of the whole network for several test dataset sizes.

We have also observed that the variability of errors increases linearly with the number of instances of the test dataset. For this reason, it is important to train the model again after a certain period. Due to the fact that both the MAE and the error variability follow a linear function, we could easily determine a trade-off between the resulting error and the frequency of updates to the model.

Figure 7 depicts the evolution of the absolute prediction error over time. In both figures, the RT model was trained at time 0. The left side shows 288 values predicted after training with 1728 instances. As we can observe, most of the errors are within the range from 0 to 1.5. On the other hand, the right side also shows 288 values predicted, but after training with only 288 instances. In this second case, the absolute error dispersion is higher and the range is also larger (from 0 to 3). The red lines in both figures represent the linear regression of the absolute error. The slope of this regression is $3.8 \cdot 10^{-4}$ and $4.8 \cdot 10^{-4}$ for the first and second figure respectively. It is important to notice that in the second case the value of the slope is a magnitude of order higher. Both figures show the impact of the size of the training dataset on the prediction error. We can observe that the larger the size of the training dataset, the smaller the error.
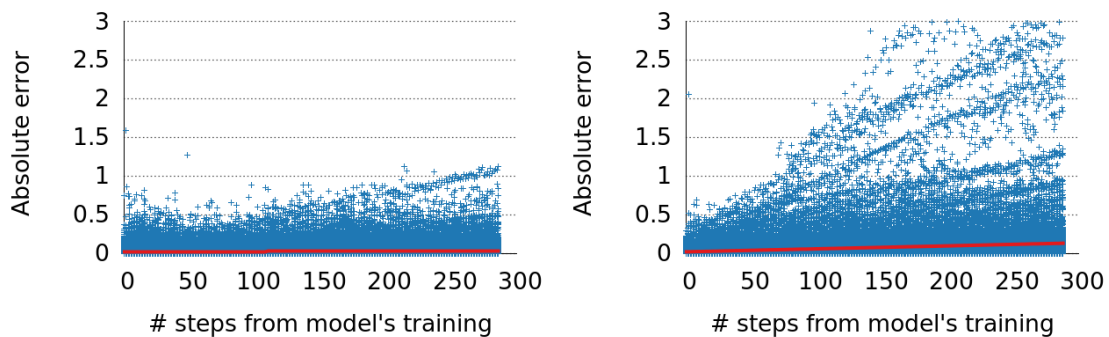

Fig. 7.   RT absolute error of LQ predictions with training data set of 1728 (left) and 228 (right) instances.

E. Analysis of Topological Features

In this section we present the results obtained after analysing whether topological features of links and nodes are related with the behaviour of the LQ. The first topological feature we analysed was the node degree. Figure 8 shows a histogram of the degrees of the network nodes. Notice that the majority of nodes (almost 90%) have degrees within the range from 1 to 5. We also have a significant number of nodes with degrees within the range from 6 to 12. Finally, we only have a small number of nodes with degrees higher than 12.
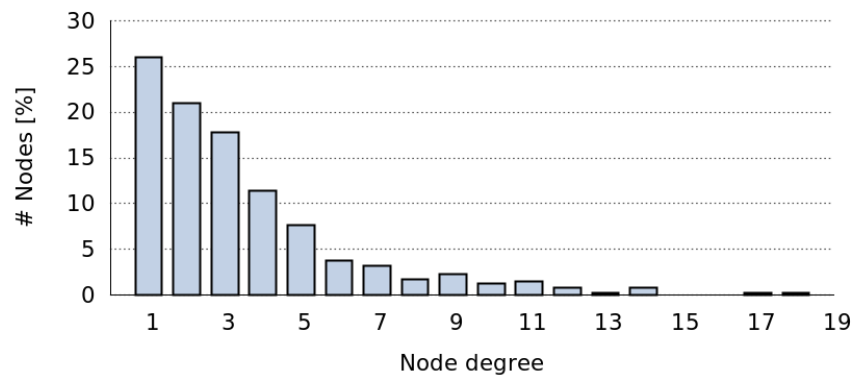
Fig. 8.   Histogram of node degree (for a total of 489 nodes).

Figure 9 depicts the average LQ of nodes (computed as the average of link-quality values on each edge of the node). The mean value for each node degree is depicted by the red line. We can observe that there are no significant differences of average LQ for different Node Degrees. The worst LQ value corresponds to Node Degree 3, with a mean LQ value of about 0.86. Node Degrees of 9 and above present slightly better values of LQ, but the number of nodes is very small to be statistically relevant. Hence, we do not observe a clear correlation between Node Degree and average LQ.
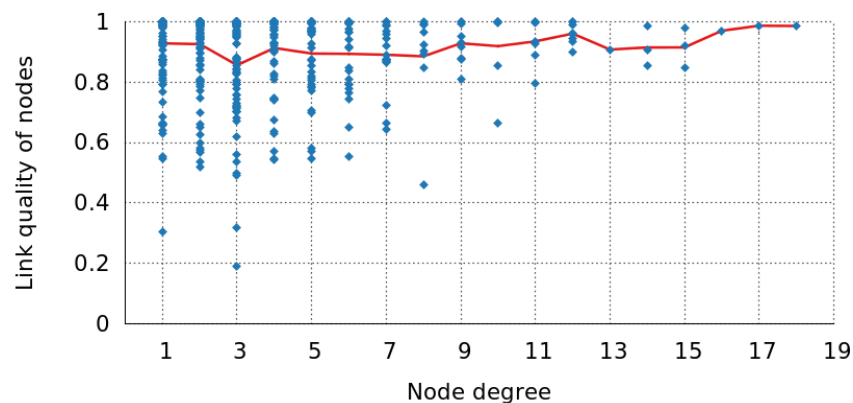
Fig. 9.   Node degree vs. average LQ (the red line corresponds to the mean value for each degree).

The next topological feature we analysed is the edge betweenness. This metric measures how often an edge appears on the shortest paths between nodes in the network. Figure 10 plots the LQ that corresponds to different edge betweenness values (represented in the figure as the amount of the shortest path length that traverses a node, normalized, with logarithmic scale). In this figure we identify six value intervals (from 10^{-6} to 0). Interval 1 (from 10^{-6} to 10^{-5}) presents a low dispersion of LQ values with a large concentration of them around 0.9. Interval 2 (from 10^{-5} to 10^{-4}) and interval 3 (from 10^{-4} to 10^{-3}) present a moderate dispersion of LQ values, the majority of them having an LQ=1 and a few of them an

LQ between 0.2 and 0.9. In interval 4 (from 10^{-3} to 10^{-2}) and interval 5 (from 10^{-2} to 10^{-1}) most nodes also have an LQ=1 but in this case there is a high dispersion and a significant number of values are between 0.1 and 0.9. Finally, Interval 6 is not relevant as it only deals with a few isolated values. With these results, we can conclude that there is a clear correlation between LQ and edge betweenness.
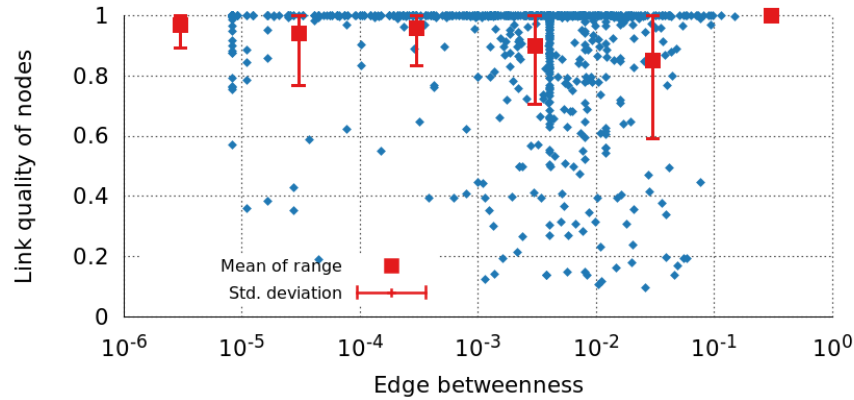

Fig. 10.   Edge betweenness vs. link quality.

In order to analyse the behaviour of our predictor with respect to the edge betweenness metric, we represented the RT prediction accuracy versus the Mean Average Error, as shown in Figure 11. We can observe that, in general, the behaviour is good for the whole range of edge betweenness values. However, for the two edge betweenness values with more concentration, the MAE is slightly worse (about 95% of prediction accuracy). When comparing the behaviour observed in figures 10 and 11, we can conclude that our predictor can successfully follow the behaviour pattern displayed by the edge betweenness metric.
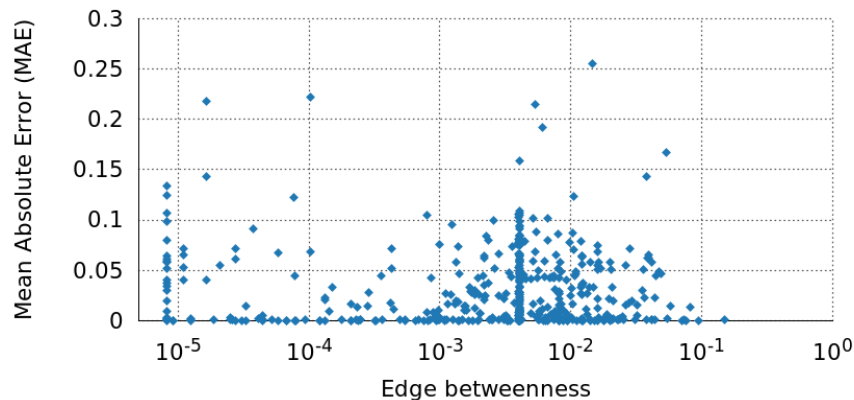

Fig. 11. Edge betweenness vs. RT prediction Mean Absolute Error.

F. Lessons Learned

The global analysis of nodes allowed us to determine the general behaviour of the LQ by considering the entire network as a whole. However, some individual groups of nodes and links seem to present a divergent behaviour. Therefore, it would be interesting to further analyse the behaviour of such groups, in order to be able to characterize the observed behaviour heterogeneity. Nonetheless, from the global analysis of nodes performed in the previous sections, we are able to draw some interesting lessons and conclusions:

- RT is the prediction method that performs best, followed by SVM.
- Globally, the MAE and the RMSE metrics show an equivalent behaviour.
- Different lag window sizes produce slightly different results, but we could not determine an optimal size.
- The degradation of RT over time follows a linear function and the larger the size of the training dataset used, the smaller the resulting error will be.
- Regarding the topological features, the node degree does not present a correlation with the LQ, but there are indications that the edge betweenness affects the LQ results. However, our results are not conclusive enough to allow us to apply this correlation in the prediction process.

It is also important to notice that any LQ values should lie in the range 0 to 1. Therefore, the predicted values that are greater than 1 or lower than 0, should be considered to be exactly 1 or 0 respectively. Moreover, if the LQ is very poor, it makes no sense to make a prediction (it would be better to disable the predictor). These measures can be applied to improve the prediction results.

V. Analysis of Individual Links and Nodes

In this section, we perform an analysis of individual links and nodes. Our aim is to identify and analyse data individually, in order to classify the nodes and links according to their behaviour. Thus, we try to answer questions such as: "Do some nodes and links show a similar performance?", and "Is it possible to group nodes and links in clusters according to similarities in their behaviour?" The first part of this section analyses the variability of the LQ prediction among links. The last part, on the other hand, analyses the variability of the LQ prediction over time. The aim is to determine if this prediction remains constant with time or if it presents variations, and in the latter case, try to classify the behaviour of these variations.

A. Variability of Link Quality Prediction among Links

As an initial analysis, we classified the links that presented particular LQ values in different groups. Figure 12 shows a graph with the results considering both all links (blue line) and only those links that present variations in the LQ (red line).
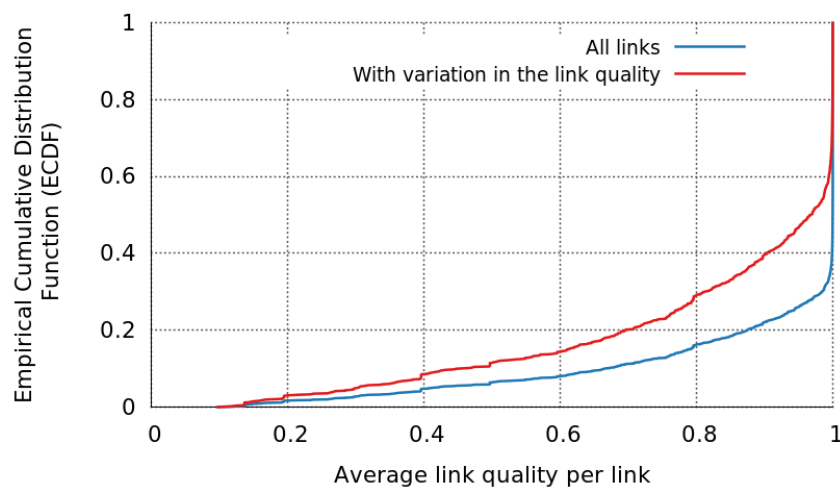

Fig. 12. Empirical Cumulative Distribution Function of the average LQ per link.

If we take into account the whole set of links, we can identify three different ranges of LQ values. The biggest group (80% of the links) contains link qualities higher than 90%. A second group of links (10%) contains link qualities between 70% and 90%. The remaining 10% of the links present link qualities below 70%. Similarly, if we consider only the links that present variations in the LQ, these percentages decrease slightly but they seem to follow the same trend. In this case, 60% of the links show link qualities of 90% or more, 20% of the links show link qualities between 70% and 90% and finally, the remaining 20% of the links present link qualities under 70%. Therefore, we have a whole range of LQ values but the majority of them are high.

The next analysis was intended to evaluate the behaviour of the four algorithms (SVM, RT, kNN, and GPR) for variations in the LQ in steps of 0.1, and therefore, to check if the behaviour changes for any particular range of LQ values (i.e., to verify if RT is not the best algorithm for a specific range). This study extends the global analysis presented in Figures 2 and 3 (left). Figure 13 shows the results obtained.
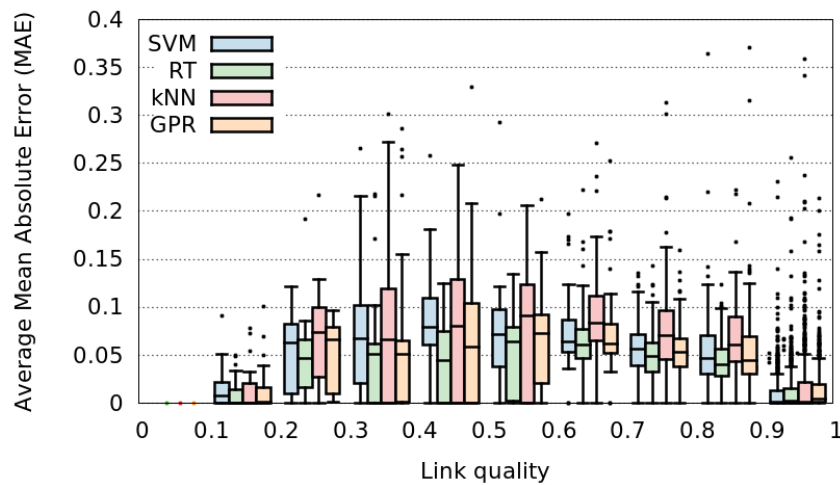

Fig. 13. Average Mean Absolute Error versus Link Quality in steps of 0.1.

As we can observe RT is the best solution for LQ values below 0.9, but in the range from 0.9 to 1, SVM shows a slightly better performance. Table 1 shows a more detailed analysis of the data for RT and SVM for LQ values in the 0.9-1 range.

Method	Min.	1st qu.	Median	Mean	3rd qu.	Max.	St. dev.
RT	0.000	0.0002	0.0020	0.0129	0.0154	0.5167	0.03105
SVM	0.000	0.0004	0.0015	0.0130	0.0125	1.4290	0.06172

Table 1: MAE in the 0.9-1 LQ range for RT and SVM algorithms.

These results show that SVM has better (lower values) median and 3rd quartile values, the mean values are almost the same for both methods, and SVM performs worse in the 1st quartile, maximum, and standard deviation results. This behaviour can be explained due to the fact that SVM presents some very high values that produce a significant increase in the mean and standard deviation values. Despite this, results in Figure 13 confirm that for 87.5% of the links SVM performs better than RT.

B. Variability of Link Quality Prediction over Time

The last LQ analysis performed was aimed at evaluating how the LQ values of each link vary over time. Figure 14 represents the evolution of the LQ values of a sample link over a period of 24 hours. We can observe that the LQ values on this link are not constant: most of the time the values are above 0.85, but at certain times, they drop below 0.7.
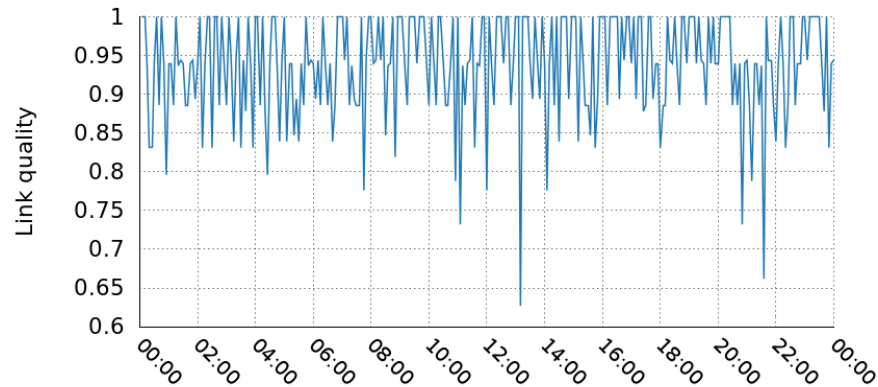

Fig. 14. Example of the variation of the LQ of a link over a period of 24 hours.

In order to include time components in our study, first we need to select the right mathematical tools. We therefore, need to analyse time series data (the evolution of LQ values over time) and, then, classify the links into groups (clustering) according to their behaviour.

The analysis and comparison of time series is a common topic in the literature [25] [26]. The classical Discrete Fourier Transform used to characterize time series, has been improved by the use of Wavelets, and more precisely, Haar wavelets [27]. The Haar wavelet captures the shape of time series better than the Fourier Transform (within the context of similarity search). The Haar transform decomposes a discrete signal into two subsignals of half its length. One subsignal is a moving average, or trend; the other subsignal is a moving difference, or fluctuation. The Haar wavelet has a major drawback: the basic functions for the Haar wavelet are not smooth (i.e., they are not continuously differentiable). On the contrary, the Haar wavelet approximates any signal by a ladder-like structure. In addition, the number of features we have to use is high (this property is referred to as "slow convergence").

In our case, we used nine features to apply the Haar wavelet to our analysis of the LQ time series. However, due to the fact that the Haar window is only two elements wide, when a big change appears between an even and an odd value, the high-frequency coefficients will not reflect such a change. Therefore, the Haar wavelet was unable to capture high-frequency changes in our LQ time series, which are very common in our dataset (LQ values that change suddenly between 1 and 0 or vice versa). The Daubechies wavelet is similar to the Haar wavelet in the sense that it also computes the moving averages and differences via scalar products, but the scaling signals and wavelets are different. Daubechies wavelets use overlapping windows so that all high-frequency changes are included. For our analysis, we used Daubechies wavelets, with a 4-element window, and 9 features. On the other side, K-means is the most popular partitioning method and requires prior specification of the number of clusters to extract. We used this method to classify our LQ time series in different clusters with the main purpose of determining whether or not we can achieve better prediction accuracy for a specific cluster.

Our first task, in order to identify the different kinds of links (according to the temporal evolution of their LQ) was to determine the best number of clusters. To do so, we selected only those links that presented

changes in the LQ (we excluded links with a constant value of LQ=1). Figure 15 shows the "*within clusters sum of squares*" for different possible numbers of clusters (*K*), up to 20. It is important to notice that although small sum values are better because the elements within a cluster are more similar to each other, there is not a clear criterion to decide the "best" number of clusters. According to the figure, it would be appropriate to try *K* values of 6, 8, or 10. However, we have to take into consideration that we are not only interested in an analytical interpretation of the common features of the links within a cluster, but also in a human-readable interpretation, and when the number of clusters is large, the amount of elements belonging to some clusters are too small, making more difficult for humans to interpret the criteria used to classify the links that belong to each cluster. For this reason, we choose *K*=4.
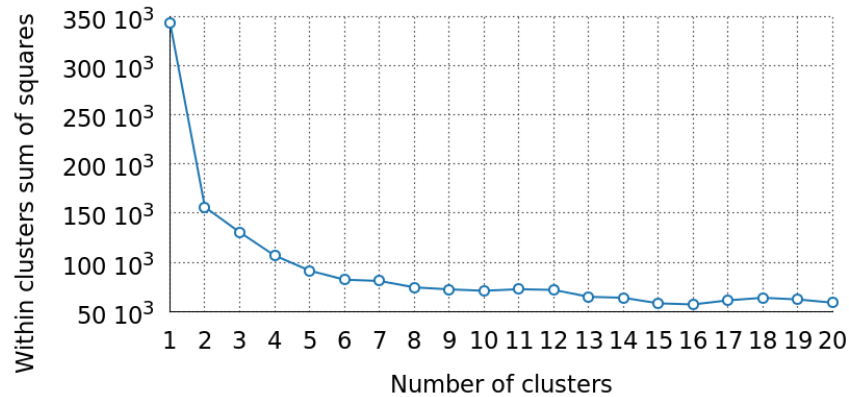

Fig. 15. Possible number of clusters for LQ time series classification.

The next step involved applying the K-means algorithm, with *K*=4 clusters. In Figure 16 we can observe a sample of the LQ time series for each one of the four clusters (note that the axes are similar to Figure 14, but in this case the x axis is the entire 7-day period). The main features of the links in each cluster are listed below:

- Cluster 1 (618 links): links with LQ very close to 1 and small oscillations.
- Cluster 2 (148 links): links with large oscillations in LQ.
- Cluster 3 (206 links): links with small LQ oscillations, but LQ values far from 1.
- Cluster 4 (55 links): links with ON/OFF behaviour (with small oscillations).

Once we identified the four clusters, we could analyse their behaviour. Figure 17 contains the boxplots of the average LQ for each one of the four cluster types. We can observe that clusters 1 and 4 achieve the best LQ values (closer to 1), cluster 3 has LQ values around 0.7, whereas cluster 2 is the worst (with average LQ values of only 0.4). This implies that the prediction would be better for clusters 1 and 4, but we will find more problems in clusters 2 and 3. When a link in cluster 4 is in the OFF state, the link does not exist and it does not have LQ values, thus the prediction is disabled.
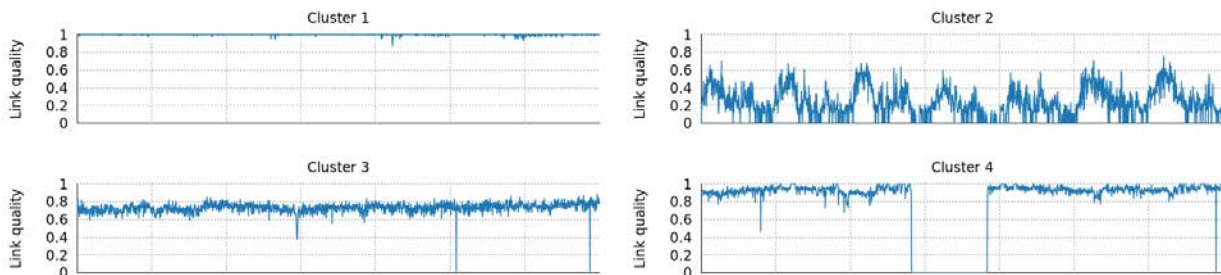

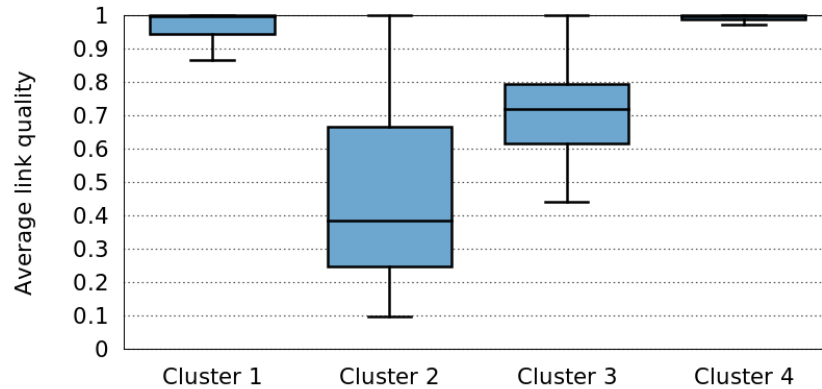Fig. 16. Sample of LQ time series for each of the 4 clusters from K-means algorithm.

15

Fig. 17. Average link quality of the links of each cluster, depicted as box plots.

We have also analysed how well each one of the four prediction methods (SVM, RT, kNN, and GPR) performs for each of the four cluster types. The results are available in Figure 18. Comparing these results with the ones depicted in Figures 2 and 3, we confirm that the global results obtained previously, where RT was the best prediction method, are now still valid for clusters 2 and 3. For cluster 1, nevertheless, both RT and SVM have a similar performance. A different result is obtained for cluster 4, where RT is the third best method, after kNN (the second best) and SVM (the best). Previously, in Figure 13, we had also observed a case with SVM performing better than RT, but now the difference is even more significant.
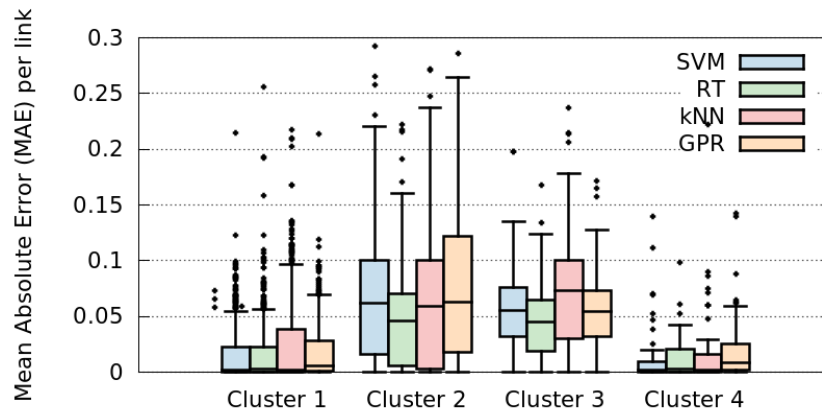

Fig. 18. Prediction error of each method for each cluster, depicted as box plot.

In order to further analyse the results presented in Figure 18, we considered each of the four clusters separately and evaluated each one of them for increasing LQ values in steps of 0.1 (as done previously for Figure 13). The results are presented in Figure 19. To compare the different prediction methods, we use the 3rd quartile (the top part of each box) because it represents the majority of the links (75% of the error values lie below the 3rd quartile). The results can be analysed for each cluster:

- Cluster 1: LQ values are in the range from 0.7 to 1. For LQ values below 0.9, RT achieves the best predictions. However, in the range from 0.9 to 1, SVM is again the best algorithm.

- Cluster 2: here we find a broader range of LQ values, from 0.1 to 1. In all ranges, RT performs best than the other prediction methods.
- Cluster 3: this is the cluster with the second wider range, with LQ values between 0.4 and 1. Similar to cluster 1, in this case it is also better to use RT for LQ values below 0.9, but switching to SVM for LQ values above 0.9.
- Cluster 4: this is the cluster with the narrowest LQ range, from 0.8 to 1. Also in this cluster, it is better to apply RT or SVM for values below or above 0.9 respectively.

We should note that although SVM is better than RT in some cases, it takes a more aggressive prediction approach, trying to follow very accurately any change, but generating more outliers and therefore, making more errors. On the contrary, RT is more conservative when predicting changes, and for that reason produces a lower number of errors than SVM.

Comparing Figure 13 and Figure 19, we can observe:

- The boxplots in Figure 13 with LQ values between 0 and 0.4 correspond to cluster 2 in Figure 19.
- The boxplots in Figure 13 with LQ values in the range from 0.6 to 0.8 correspond to cluster 3 in Figure 19.
- The boxplots in Figure 13 with LQ values above 0.8 correspond mainly to cluster 1 (because cluster 1 contains 618 links, whereas cluster 4 only contains 55 links).
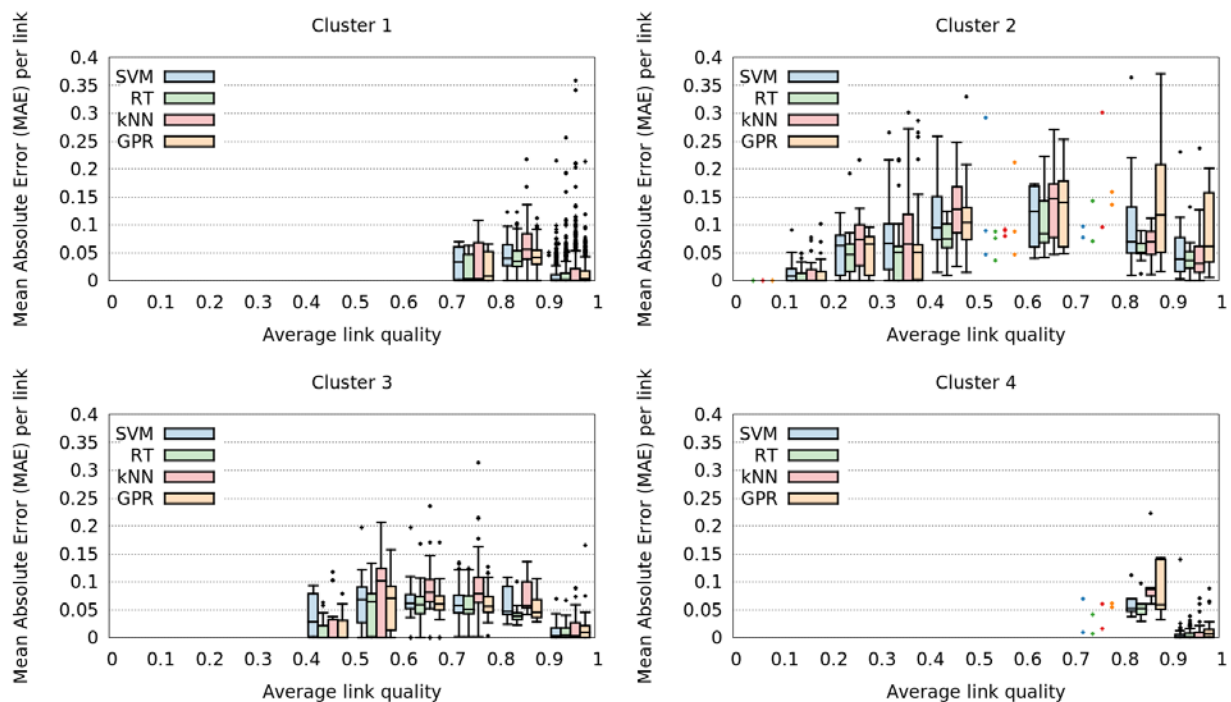

Fig. 19. Average Mean Absolute Error versus link quality (in steps of 0.1), for each prediction method and cluster, depicted as box plots.

C. Lessons Learned

The LQ prediction analysis we performed individually on the links provided us with some interesting results. This helped us reach the following conclusions:

- In general, the RT predictor achieves the best results.
- The majority of the LQ values lie within the range from 0.9 to 1.
- For clusters 1, 3, and 4, the SVM predictor performs better than the RT for LQ values within the range from 0.9 to 1.
- For cluster 2, RT always produces the best results for the whole range of LQ values.
- To include topological features in the predictor does not improve the results.
- The predictor generates LQ values out of the range from 0.0 to 1.0, which offers an opportunity to improve the predictor.

VI. Enhanced Predictor

Based on the observations made in the previous section, we propose two orthogonal improvements to the prediction process: (1) saturation and (2) RT/SVM. On the one hand, we have seen that all four predictors analysed in this paper make predictions of LQ values by applying methods and computations that produce unbounded values. Therefore, we want to study how much the prediction process improves when we limit the predicted values to the "logical" range from 0.0 to 1.0. Thus, if the prediction produces values above 1 or below 0, we "saturate" the predicted value (i.e., all predictions above 1 would be 1, and all predictions below 0 would be 0). Obviously, this simple improvement will produce smaller errors, and result in a better overall performance of the prediction process. On the other hand, we have also seen that sometimes the SVM predictor performs better than the RT predictor (LQ values within the range from 0.9 to 1 for clusters 1, 3 and 4) but sometimes the RT predictor performs better than SVM (the whole range of LQ values for cluster 2). Therefore, we propose to select the best of these two prediction approaches depending of the current cluster and range of values.

In order to test the various improvements, starting from the "*Base*" method (RT without saturation), we add the saturation process (RT Sat.), then a combination of RT and SVM (RT/SVM) and, finally, optimizations to both algorithms (RT/SVM Sat.). The analysis was performed with the following parameters:

- Maximum lag window size: 12
- Training dataset size: 1728
- Test dataset size: 288
- Metric: Absolute Error of LQ overall predictions

	RT (base)	RT Sat %Improv.	RT/SVM %Improv.	RT/SVM Sat. %Improv.
2nd qu.	0.0040	25%	62.5%	75.0%
3rd qu.	0.0298	2.0%	3.0%	4.0%
87.5% per.	0.0645	1.7%	0.2%	1.1%
Max.	8.2130	87.8%	0.0%	87.8%

Table 2: Results of the Base (RT) and *Optimized* methods (Sat.: with saturation, RT/SVM: combination of the RT and the SVM algorithms, for clusters and ranges where SVM performs better than RT). %Improv.: percentage of improvement over the *Base* method. The results correspond to the maximum and representative quartiles.

A comparison of the results obtained for the *Base* and the three *Optimized* methods, together with the improvement percentage (computed as (*Base-Optimized*)/*Base*) are presented in Table 2 and Figure 20.
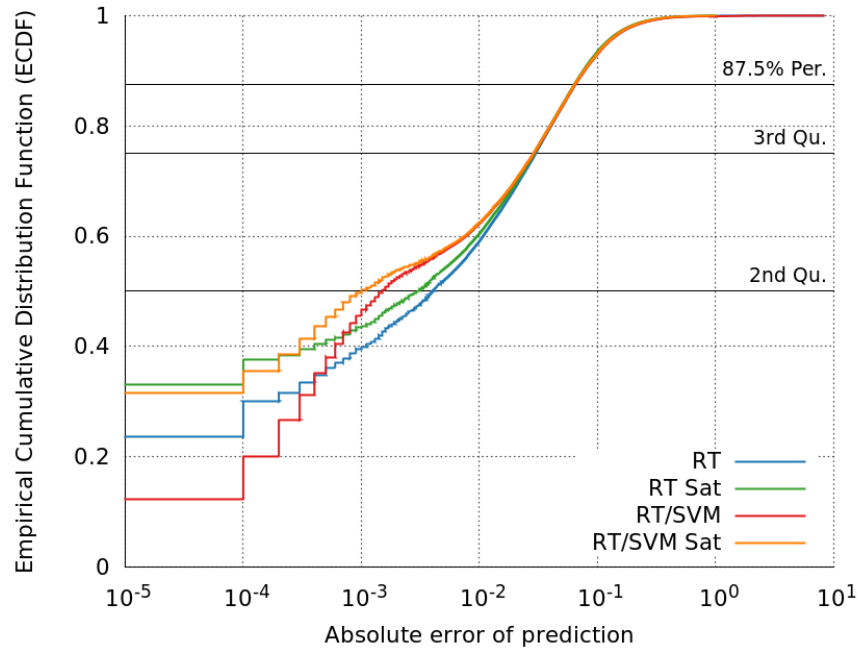

Figure 20. Empirical Cumulative Distribution Function of the absolute error of the overall link quality predictions for the Base (RT) and the three enhanced methods.

If we analyse the results presented in Figure 20 and Table 2, we can conclude the following:

- For the 50% of the links that had good prediction results (2nd quartile), all improvements have a significant impact on the results, especially when combining the RT and the SVM algorithms.
- For links that displayed the worst behaviour (with an absolute error close to the maximum value) only the saturation strategy produces some improvements.
- For the rest of the links, which achieved a moderate prediction performance (those with values from the 3rd quartile to almost the maximum), any of the strategies proposed showed significant improvements in the results.
- According to Figure 20, by applying the saturation strategy we can observe an increment in the ECDF for absolute errors smaller than 10^{-5} (in practice, this means an absolute error equal to 0). Therefore, we can claim that the optimization strategies based on the correction of the predicted value produce a significant improvement in the results.

VII. Conclusions

This study demonstrates that time series analysis is a promising approach to accurately predict LQ values in community networks. This technique can be used to improve the performance of the routing protocol by

providing information to make appropriate and timely decisions to maximize the delivery rate and minimize traffic congestion.

We analysed results from four learning algorithms (Support Vector Machine, k-Nearest Neighbours, Regression Tree and Gaussian Processes for Regression) that model time series. All algorithms achieved percentages of success between 95% and 98% when predicting the next future value of the LQ, with the Regression Tree being the best one. Moreover, these results were obtained only considering those links that experienced variations. Therefore, the prediction accuracy could have been even better including all the network links. In addition, we showed that the prediction of values that are more than one step ahead in time (and not just the next value) also achieves high success ratios, between 97% and 98%. We also observed that the size of the training data set is a key factor to achieve high accuracy in the predictions. The bigger the size of the data set, the smaller the degradation of the error over time.

The global analysis of the LQ behaviour we performed in this work, allowed us to identify the best prediction algorithm (Regression Tree) and metric (Mean Absolute Error), and to understand the impact of lag windows in the prediction (although we were not able to determine the best lag window size). It also helped us to evaluate the accuracy of prediction some time steps ahead into the future (it seems possible to predict the LQ some steps ahead), the degradation of prediction over time (the degradation follows a linear function: the larger the size of the training dataset, the smaller the error), and the correlation of prediction with some topological features (node degree does not present correlation with LQ, but there are indications to the fact that edge betweenness produces changes in the LQ; however, it is not straightforward to apply this correlation to the prediction process). We also analysed the behaviour of links individually to identify the variability of the LQ prediction between links (for LQ values within range 0.9-1, SVM performs better than RT in 87.5% of links) and over time (we identified four kinds of links according to the results of a time series clustering; and SVM continued performing better than RT in three of the four clusters). Finally, we also enhanced the prediction method taking into consideration our previous knowledge about the expected LQ values. To realize this, the optimized predictor applies an approximation (saturation) of the predicted LQ values to keep them within the range from 0 to 1. Combining RT, SVM and saturation, we achieved improvements of up to 75% for the second quartile, but for links that displayed the worst behaviour, only the saturation strategy produces some improvements. Therefore, we can claim that the optimization strategies based on the correction of the predicted value lead to a significant improvement in the results.

As future work, we plan to identify which links contribute most to the error in the LQ prediction and to understand what factors make it more difficult to predict the behaviour of these links. We also want to analyse if there is a subset of links that provides real trends in LQ. Moreover, we believe that social analysis and user profiles may help to understand link quality behaviour and its prediction. On the other hand, we plan to improve the prediction process discarding those links whose relation between LQ and prediction accuracy is above a certain threshold. This discarding can be done at the very beginning (static discarding) or at any given time (dynamic discarding). Furthermore, we are currently implementing our proposal, that will allow us to determine its cost, how to be used by routing protocols, and if the addition of other sources of information (i.e. NIC parameters) could improve the predictions. Finally, we want to extend the analysis presented in this research work to other community networks, such as guifi.net [2] to evaluate if the observed behaviour could be generalized.

Acknowledgments

References

[1] J. Avonts, B. Braem and C. Blondia. "A Questionnaire Based Examination of Community Networks". International Workshop on Community Networks and Bottom-up-Broadband, 2013 DOI:10.1109/WiMOB.2013.6673333

[2] R. Baig, R. Roca, L. Navarro and F. Freitag, "guifi.net: a Network Infrastructure Commons", In Proceedings of the Seventh International Conference on Information and Communication Technologies and Development, 2015 DOI:10.1145/2737856.2737900

[3] FunkFeuer, a free, experimental network in Vienna. http://www.funkfeuer.at/

[4] B. Braem, C. Blondia, C. Barz, H. Rogge, F. Freitag, L. Navarro, J. Bonicioli, S. Papathanasiou, P. Escrich, R. Baig Viñas, A. L. Kaplan, A. Neumann, I. Vilata i Balaguer, B. Tatum, M. Matson. "A Case for Research with and on Community Networks". Computer Communication Review, vol. 43 (3), 2013 DOI:10.1145/2500098.2500108

[5] A. Woo, et al. "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks". International Conference on Embedded Networked Sensor Systems, 2003. DOI:10.1145/958491.958494

[6] R. Draves, J. Padhye and B. Zill. "Comparison of Routing Metrics for Static Multi-Hop Wireless Networks". The ACM Sigcomm, vol. 34 (4), 2004. DOI:10.1145/1030194.1015483

[7] C. E. Koksal and H. Balakrishnan. "Quality-Aware Routing Metrics for Time-Varying Wireless Mesh Networks". Journal on Selected Areas in Communications, vol. 24 (11), 2006. DOI:10.1109/JSAC.2006.881637

[8] S. Vural, Dali Wei, and K. Moessner. "Survey of Experimental Evaluation Studies for Wireless Mesh Network Deployments in Urban Areas Towards Ubiquitous Internet". Communications Surveys Tutorials, vol. 15 (1) 2013. DOI:10.1109/SURV.2012.021312.00018

[9] J. F. Rodríguez-Covili, S. F. Ochoa, J. A. Pino, R. Meseguer, E. Medina and D. Royo. "A Communication Infrastructure to Ease the Development of Mobile Collaborative Applications". Journal of Network and Computer Applications, vol. 34 (6), 2011. DOI:10.1016/j.jnca.2010.12.014

[10] N. Baccour, et al. "Radio Link Quality Estimation in Wireless Sensor Networks: a Survey", ACM Transactions on Sensor Networks, vol. 8 (4), 2012. DOI:10.1145/2240116.2240123

[11] L. Lu and T. Zhou. "Link Prediction in Complex Networks: A Survey", Physica A: Statistical Mechanics and its Applications, vol. 390 (6), 2011. DOI:10.1016/j.physa.2010.11.027

[12] T. Rault, A. Bouabdallah and Y. Challal. "Energy Efficiency in Wireless Sensor Networks: A Top-Down Survey". Journal of Computer Networks, vol 67, 2014 DOI:10.1016/j.comnet.2014.03.027

[13] R. Fonseca, O. Gnawali, K. Jamieson and P. Lewis. "Four Bit Wireless Link Estimation". International Workshop on Hot Topics in Networks, 2007.

[14] M. Yunquian. "Improving Wireless Link Delivery Ratio Classification with Packet SNR". International Conference on Electro Information Technology, 2005. DOI:10.1109/EIT.2005.1626960

[15] M. Senel, et al. "A Kalman Filter Based Link Quality Estimation Scheme for Wireless Sensor Networks". Global Communications Conference, 2007. DOI:10.1109/GLOCOM.2007.169

[16] A. Cerpa, J. L. Wong, M. Potkonjak and D. Estrin. "Temporal Properties of Low Power Wireless Links: Modeling and Implications on Multi-Hop Routing". International Symposium on Mobile Ad Hoc Networking and Computing, 2005. DOI:10.1145/1062689.1062741

[17] D. De Couto, D. Aguayo, J. Bicket and R. Morris, "A High Throughput Path Metric for Multi-Hop Wireless Routing". Mobicom Conference, 2003. DOI:10.1145/938985.939000

[18] S. Naimi, A. Busson, V. Vèque, L. B. H. Slama, R. Bouallegue. "Anticipation of ETX Metric to Manage Mobility in Ad Hoc Wireless Networks". International Conference ADHOC-NOW, 2014. DOI:10.1145/381677.381685

[19] C. Renner, S. Ernst, C. Weyer and V. Turau. "Prediction Accuracy of Link-Quality Estimators". European Conference of Wireless Sensor Networks, 2011. DOI: 10.1007/978-3-642-19186-2_1

[20] Y. Wang, et al. "A New Scheme on Link Quality Prediction and its Applications to Metric-Based Routing". International Conference on Embedded Networked Sensor Systems, 2005. DOI:10.1145/1098918.1098958

[21] FunkFeuer-CONFINE Open Data. http://opendata.confine-project.eu/group/funkfeuer

[22] L. Maccari, R. L. Cigno, "A week in the life of three large wireless community networks", Ad Hoc Networks vol. 24, Part B, 2015 DOI:10.1016/j.adhoc.2014.07.016

[23] M. A. Alsheikh, S. Lin, D. Niyato and H-P Tan. "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications". The IEEE Communications Surveys & Tutorials, vol. 16 (4), 2014. DOI:10.1109/COMST.2014.2320099

[24] M. Hall, et al. "The WEKA Data Mining Software: An Update". SIGKDD Explorations, vol. 11 (1), 2009. DOI:10.1145/1656274.1656278

[25] P. Chaovalit, A. Gangopadhyay, G. Karabatis and Z Chen. "Discrete Wavelet Transform-Based Time Series Analysis and Mining". ACM Computer Surveys, vol. 43 (2), 2011. DOI:10.1145/1883612.1883613

[26] T. W. Liao. "Clustering of Time Series Data: a Survey". Pattern Recognition, vol. 38 (11), 2005. DOI:10.1016/j.patcog.2005.01.025

[27] F.F. Chan, A, W. Fu and C. Yu. "Haar Wavelets for Efficient Similarity Search of Time-Series: With and Without Time Warping". IEEE Transactions on Knowledge and Data Engineering, vol. 15 (3), 2003. DOI:10.1109/TKDE.2003.1198399