Wireless Link Quality Prediction in IoT Networks

Miguel Landry Foko Sindjoung, Pascale Minet

▶ To cite this version:

HAL Id: hal-02432805

https://hal.science/hal-02432805v1

Submitted on 8 Jan 2020

Wireless Link Quality Prediction in IoT Networks

Miguel Landry Foko Sindjoung, Pascale Minet

Inria, 2 rue Simone Iff, CS 42112, 75589 Paris Cedex 12, France.

Email: miguel-landry.foko-sindjoung@inria.fr, pascale.minet@inria.fr

Abstract

The knowledge of link quality in IoT networks will allow a more accurate selection of wireless links to build the routes used by data gathering. Therefore, the number of retransmissions on these links is decreased, leading to a shorter end-to-end latency, a better end-to-end reliability and a larger network lifetime. In this paper, we propose to predict link quality by means of machine learning techniques applied on two metrics: RSSI and PDR. The accuracy obtained by Logistic Regression, Linear Support Vector Machine, Support Vector Machine and Random Forest classifier is obtained on the traces of a real IoT network deployed at Grenoble.

Index Terms

Machine learning, link quality estimation, wireless network, TSCH

I. Motivations

Internet of Things (IoT) is a network consisting of interconnected devices called objects, that are able to monitor their surrounding environment, produce data from the sensed information and communicate with each other without any external intervention. A specific node, called a sink, is usually in charge of gathering data and analyzing them in order to take decisions. All the IoT objects, except the sink, are subject to multiple constraints including reduced energy, limited processing capacity and reduced memory size. All these constraints make data gathering a challenge. IoT networks are used in many domains of our everyday life: health care, precision agriculture, security, process control, predictive maintenance, environment monitoring...

In this work, we are interested in studying the quality of wireless links in an IoT network. Using links of bad quality would incur re-transmissions wasting node energy which decreases network lifetime. In addition, re-transmissions increase delivery latency. Furthermore, re-transmissions may be unable to mask the unreliability of wireless links and message losses are observed. Therefore, the Quality of Experience (QoE) perceived by the user is reduced. Our objective is to use some link quality metrics to predict the link quality using machine learning methods. The expected benefit of using link quality prediction being to anticipate link breakages and route changes before loosing packets. It should result in a better QoE provided by the network.

Link quality estimation is more complex in IoT networks which may consist of a large number of deployed objects, and whose transmitters/receivers are designed to consume the least possible energy. Aguayo et al. [1] showed that such transmitters/receivers are more exposed to external interference and noises, which result in a quality degradation of wireless communication links. In that context, link quality estimation is used to enable sensors to choose the best path for delivering their data to the sink by anticipating link breakages. Efficient link quality estimation that provides a fine grain classification of links are usually based on several link quality metrics [2].

Multiple Machine Learning techniques exist. We can distinguish linear ones (e.g. Logistic regression, Linear Support Vector Machine) from non-linear ones (e.g. Support Vector Machine, Random Forest). The aim of this paper is to choose which one is the best one to predict wireless link quality.

II. State of the art

In this section, we will first present a brief overview of link quality metrics described in the literature, then we will focus on machine learning techniques that can be used to make predictions in a wireless network.

A. Link quality metrics in IoT

A good link quality metric should ensure stability and accuracy for the used learning technique. The existing metrics we know generally ensure either stability or accuracy, depending on their types, but rarely both.

To estimate the link quality in IoT, we distinguish two types of metrics: hardware-based metrics and software-based metrics. Hardware-based metrics obtain their data directly from the objects (without prior processing). The main hardware-based metrics are [2], [3]:

1) *Received Signal Strength Indicator (RSSI)*: most radio transmitters/receivers have an RSSI register, which provides information on the signal strength of the received packet. When there is no transmission, the register gives a background noise.

2) *Link Quality Indicator (LQI)*: This is a proposal of the IEEE 502.15 standard. Its evaluation is specific to the vendor. LQI can be considered as a measure based on the first eight symbols of the received signal

3) *Signal to Noise ratio (SNR)*: this is given by the difference between the actually received signal (without noise) and the background noise.

The accuracy provided by hardware-based metrics is not sufficient because of two main factors: only packets whose transmission succeed are considered and the evaluation does not consider the whole received packet, but only its first symbols.

The software-based metrics are obtained by computation and they can be classified into three categories [3]:

1) *Packet Delivery Ratio (PDR)*: also known as PRR (Packet Reception Ratio) or PSR (Packet Success Ratio), it is obtained by computing the ratio of the number of packets successfully received over the number of transmitted packets.

2) *Required Number of Packet Transmission (RNP)*: this is an emitter-based metric that counts the average number of transmission/re-transmissions required before successful packet reception. It can be computed as the number of packets transmitted and re-transmitted during an observation period, divided by the number of packets received successfully, minus one.

3) *Score*: some link quality metrics provide link estimations that do not refer to physical phenomena (such as receiving or re-transmitting packets), but instead provide a score or label that is set in a certain rank. There are several ones: Fuzzy Link Quality Estimator (F-LQE), MetricMap, CSI, ...

Whatever the metric used to estimate the link quality, some network properties must be preserved, since link quality estimation aims to improve the quality of service provided by the network. So it is important to keep in mind that good link quality estimators for IoT network should contribute to an increase in network throughput, a decrease in packet delivery latency and a higher route stability for data gathering. Therefore, they should improve the user QoE.

B. Machine learning techniques

Machine Learning (ML) is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed [4]. Multiple machine learning techniques exist. We distinguish two categories: linear models and non-linear models. In this paper, we will use the Linear Regression model (LR), the Linear Support Vertor Machine (Linear SVM) model, the SVM model and the Random Forest (RF) model. All are well-known models frequently used for classification tasks. LR and Linear SVM are linear models while SVM and RF are non-linear models.

III. PRESENTATION OF THE DATA SET

The data set used in this paper is collected from a Time-Slotted Channel Hopping (TSCH) network. TSCH is a multichannel time-slotted medium access scheme for lower-power and reliable wireless networking solutions in Low-Power Lossy Networks (LLNs) [5]. It has been designed to meet the requirements of industrial applications in terms of latency, reliability and network lifetime. To reduce latency, TSCH takes advantage on the one hand of a multichannel scheme to increase parallelism in transmissions, and on the other hand, of scheduled transmissions to avoid collisions. Furthermore,it implements a channel hopping scheme to mitigate noise and interference, and consequently to enable high reliability [6].

A. How traces have been collected?

Our data set is provided by Mercator [7], a combination of firmware and software specifically designed to collect connectivity data sets in testbeds. The same firmware runs on each node in the testbed; a software, running on a computer connected to the testbed, drives the experiment. The firmware allows the software to control the radio of the node, by sending commands to its serial port. The software can send a command to a node to either transmit a frame (specifying the frequency to transmit on), or switch the remote node to receive mode (on a particular frequency). In the receive mode, the node issues a notification to the software each time it receives a frame.

Mercator acts on the Grenoble testbed [1] consisting of 50 nodes operating on 16 channels. The test duration is about 48 hours. Collected data are timestamped. On the beginning of the experiment, the first node transmits a burst of 100 frames on the first channel, while all other nodes are in receive mode. Then, this node does the same on all the other channels. After transmitting on all channels, the next node takes over and does the same. This process takes place until all nodes have transmitted.

Measurements are done in transactions. A transaction starts from the transmission of the first frame on the first channel by the first node, to the transmission of the hundredth frame on the last channel by the last node. Each transaction is run in 2 hours 30 minutes and 58 seconds. The data set used has a total of 19 transactions.A frame sent by node A is counted as received by node B if and only if B sends an acknowledgment.

[1]https://www.iot-lab.info/deployment/grenoble/

B. Which features have been measured?

The collected data include 108659 measurements of the Packet Delivery Ratio (PDR) and the Average Received Strengh Indicator (RSSI).

Let A be the sender and B the receiver on a given channel for a transaction t.

$$PDR_t(A \to B) = \frac{Num.\ of\ acks\ sent\ by\ B}{Num.\ of\ frames\ sent\ by\ A}$$

Therefore, the PDR of a node for a transaction t, on a given channel is between 0 and 1 ($PDR_t(A_B) \in [0,1]$: 0 if no acknowledgment is sent by B and 1 if B sends 100 acknowledgments.

$$RSSI_{avg_t}(A \to B) = \frac{\sum_{i=1}^{Num.\ of\ frames\ sent\ by\ A} rssi_i}{Num.\ of\ frames\ sent\ by\ A}$$

Where $rssi_i$ represents the RSSI of the ith frame received by B on the channel considered. The RSSI values are less than 0 in all the cases. The data set is pre-processed as follows: if no frame is received by B, the corresponding RSSI will be set to the lowest value of collected RSSIs minus one.

Since the PDR values are expressed as a ratio, they have no unit, whereas the average RSSI values are given in dBm. **In the following, we write PDR to designate** $PDR_t(A \to B)$ **and RSSI to designate** $RSSI_{avg_t}(A \to B)$.

C. Statistical elements of the data set

The PDR and RSSI statistical elements are summarized in Table I. This table shows that we can use machine learning in our context, since the size of data (population) is important. We also note that there is a significant variation on both PDR and RSSI when we look at their min and max values. So, since ML models need data from different classes to make accurate predictions, we can expect to have fine predictions when using these metrics.

	PDR (in %)	RSSI (in dBm)
Population	108659	108659
Mean	0.874964	-74.551317
Standart Deviation (std)	0.248361	11.869037
Max	1	-26.39
Min	0.01	-91.00

TABLE I: PDR and RSSI statistical elements.

IV. How to predict

Machine learning models need to train themselves before being able to predict with a good accuracy. The data set used for predictions is split in two sets: a test set and a training set. The training set consists of labeled data used to train the ML model and increase prediction accuracy, whereas the test set contain data that the trained ML model may have never seen before, and which are used to measure how robust its predictions are. In our experiments, the split ratio is 75% for the training set and 25% for the validation set, as usual [8].

Since the wireless network considered operates on 16 channels and the quality of a wireless link depends on the channel considered as shown by Figures 1 and 5, we predict the quality of any link on each of the 16 channels, independently.

For each link and each channel, we split the data set into a training set and a test set; we then train the model over the training set and predict the link quality on the channel considered for the samples in the test set. By comparing the predicted set with the real set, the confusion matrix is computed by evaluating the true-positive, the true-negative, the false-positive and the false-negative for the link and channel considered. The confusion matrix for the link is obtained by summing the confusion matrices obtained for that link on all the 16 channels. As usual, the accuracy of the link prediction is equal to the sum of the number of true-positive and the number of true-negative, divided by the total number of samples considered for this link.

We then compute the confusion matrix for all links by summing the confusion matrices obtained for each link and evaluate the accuracy of the ML technique as previously. Programs written for our experiments are available online[2].

V. Comparison of ML techniques using RSSI

RSSI is a simple hardware metric able to provide a quick and accurate estimate of whether a link is of good quality or not, according to the power of signal received by a node [3]. Fig. 1 depicts the evolution of RSSI values on a network link ($Link_1$). There is no signal in channel 22, while the majority of channels have constant RSSI values. Moreover, the signal on channel 21 is of short duration compared to the simulation duration. We also notice that $Link_1$ has very low RSSI values, close to the minimum.
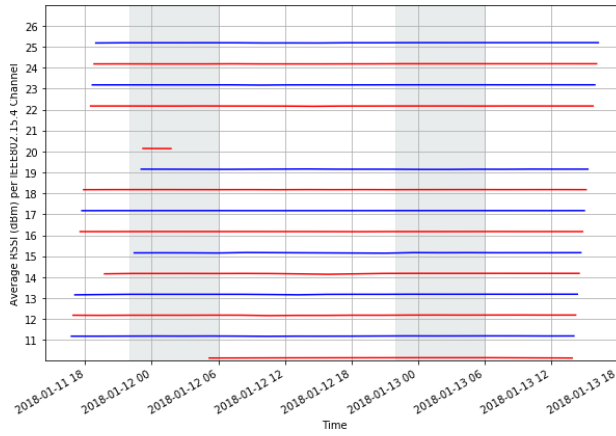
---

[2]https://github.com/miguelfoko/LinkQuality

Fig. 1: RSSI evolution over time for $Link_1$.

A. Definition of classes for the prediction using RSSI

Since the PDR value of a link has a direct impact on link performance (e.g. number of retransmissions), we draw the waterfall plot [7] in Fig. 2, which depicts the pairs (PDR, RSSI) for all the samples in the data set. We compute the median RSSI value associated with the PDR values of 0.75 and 0.3 considered as good and bad values, respectively to deduce the RSSI thresholds associated with each class, leading to the RSSI values of -85 dBm and -87 dBm, respectively.
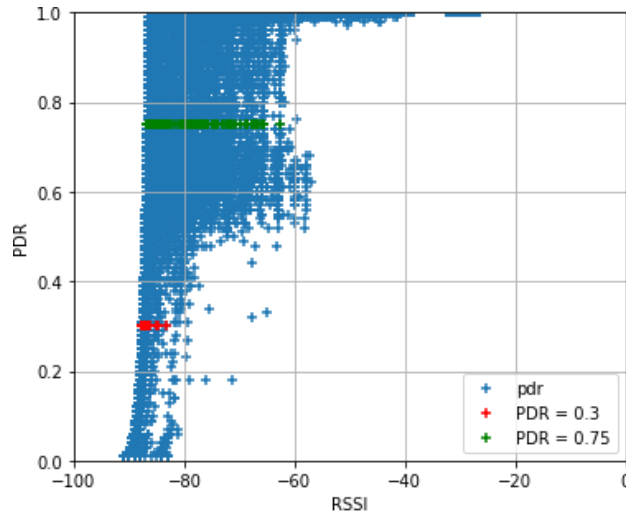

Fig. 2: Values of RSSI and PDR for the data set considered.

Hence, we define three classes for prediction using the RSSI feature:

- A link is considered as *Good* if its RSSI value is greater than or equal to -85 dBm;
- If the RSSI is less than or equal to -87 dBm, we consider the link as *Bad*;
- Otherwise ($-87 < RSSI < -85$ dBm), we consider the link as *Intermediate*.

B. Results of predictions based on RSSI over $Link_1$

Fig. 3 depicts the confusion matrices applied to the RSSI feature and obtained with LR, Linear SVM, SVM and RF, respectively. These confusion matrices correspond to the prediction on $Link_1$ (Fig. 1). For this link, the Random Forest (RF) classifier has the highest prediction accuracy, followed by SVM classifier, Logistic Regression and Linear SVM, respectively. Notice that SVM is too optimistic by classifying 24% of intermediate links as good ones.

C. Prediction based on RSSI over all the links

To obtain a more general result for the RSSI feature, we make prediction over all the network links according to the principle given in Section IV. The corresponding confusion matrices depicted in Fig 4 show that RF has the highest accuracy (99.5%), followed by SVM (98.4%), Logistic Regression (96.4%) and Linear SVM (95.5%) respectively.

(a) Logistic regression

(b) Linear SVM Classifier
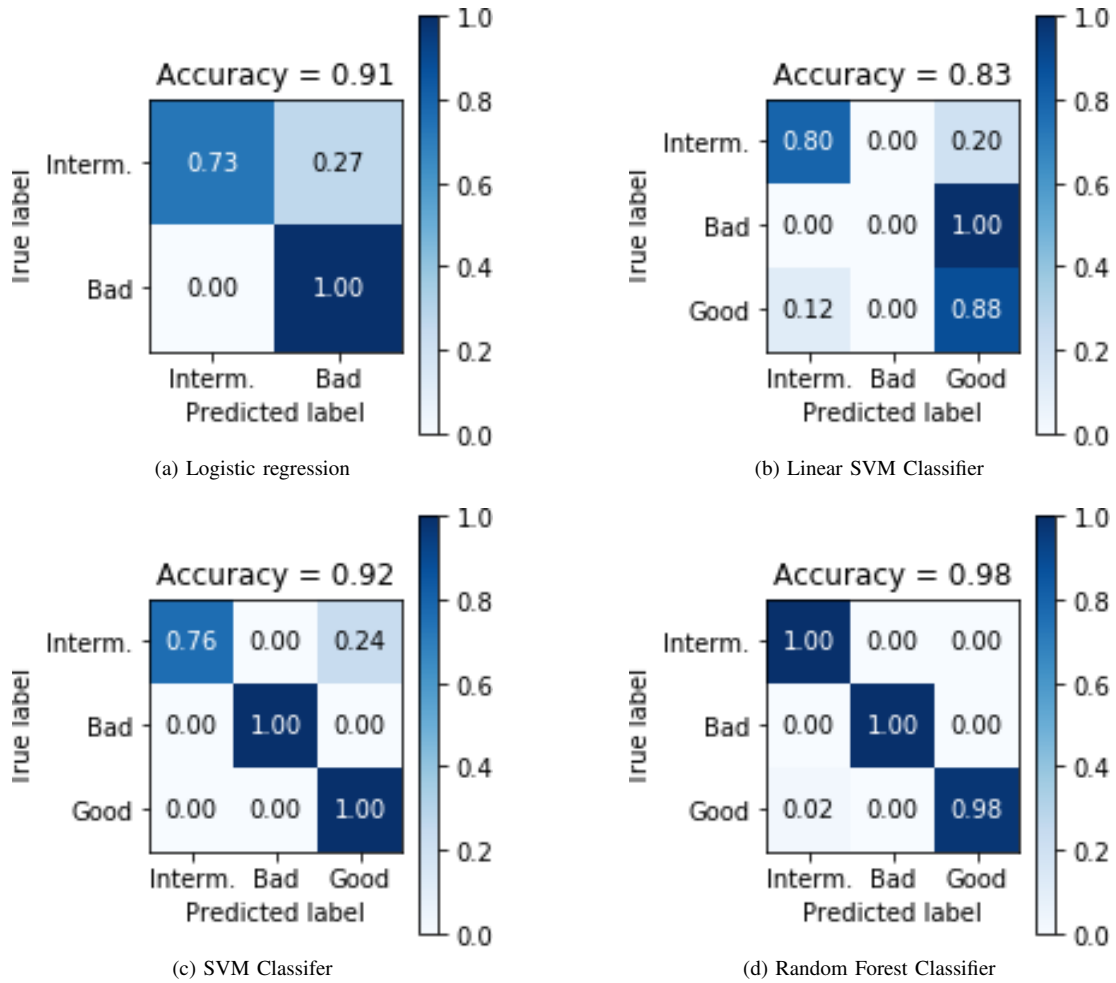
(c) SVM Classifer

(d) Random Forest Classifier

Fig. 3: Confusion matrix with RSSI on $Link_1$.

Unlike Random Forest which classifies practically all the links in their respective classes, Linear Regression classifies up to 11.8% of good links as intermediate and 0.3% of good links as bad ones, which can prevent the use of good links. More penalizing for network performance, the Linear SVM classifies 11% of bad links as intermediate and 5.2% as good (this last case can strongly impact packet delivery since worse links are classified as good).

Thereby, we conclude that the Random Forest classifier is the suited model to use when predicting link quality in a TSCH network using the RSSI feature.

However, the RSSI metric is only measured for successfully received packets. Therefore, when a radio link suffers from excessive packet losses, the RSSI metric may overestimate the link quality by not considering the information of lost packets [3], [9], [10].

VI. COMPARISON OF ML TECHNIQUES USING PDR

Re-transmissions of packet in the network significantly decrease the node energy and consequently the network lifetime. PDR enables to determine the average number of packets that need to be re-transmitted to reach their destination. Still considering $Link_1$ used before, Fig 5 presents the evolution of PDR over time on all the channels.

With regard to the evolution of PDR values over the sixteen channels of $Link_1$ (Fig 5), many channels (13, 14, 15, 24) move permanently from lower to higher values. Channel 22 does not have any PDR value while channel 21 just has PDR values for a short period. The other channels have either good or Bad PDR values.

A. Definition of classes for prediction using PDR

We define tree classes of link quality for PDR feature:

- A link is of *Good* quality if its PDR is greater than or equal to 0.75;

(a) Logistic regression

(b) Linear SVM Classifier

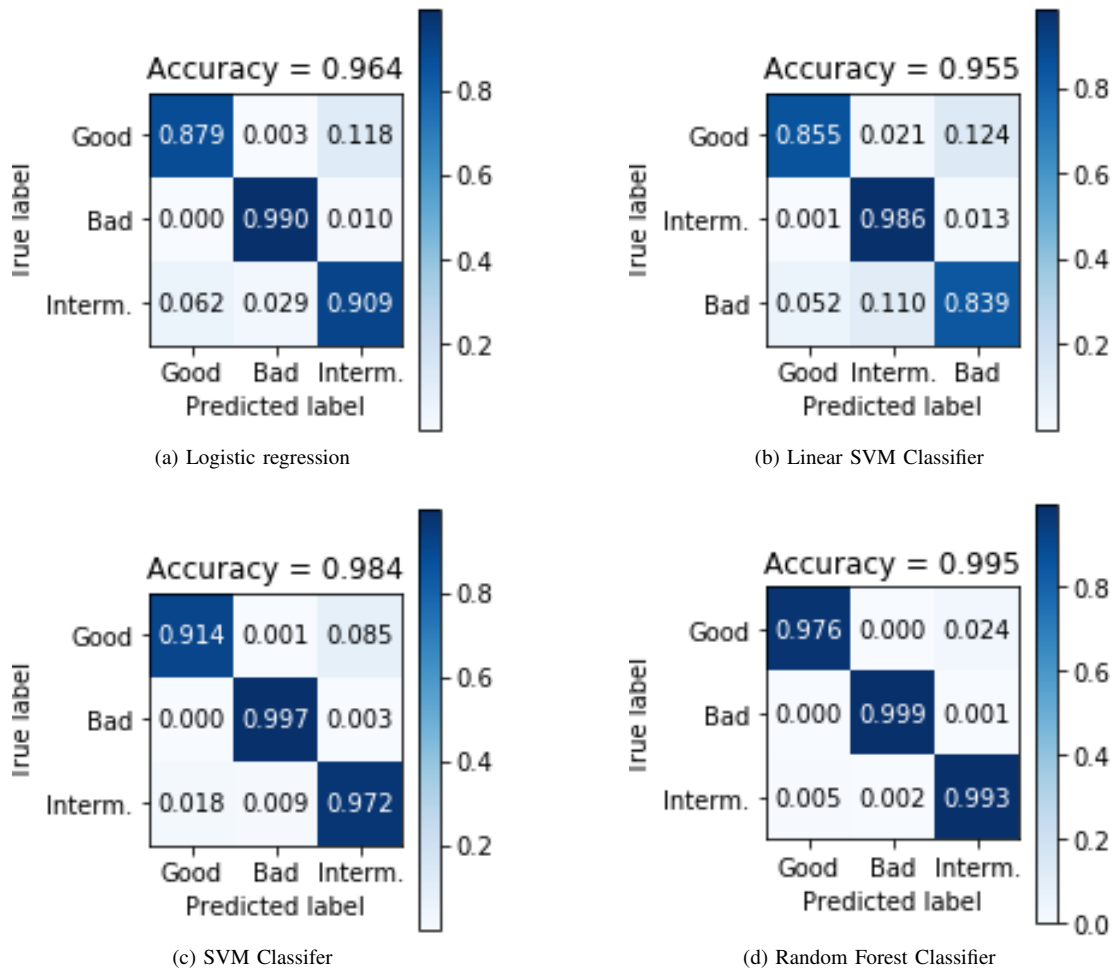(c) SVM Classifer

(d) Random Forest Classifier

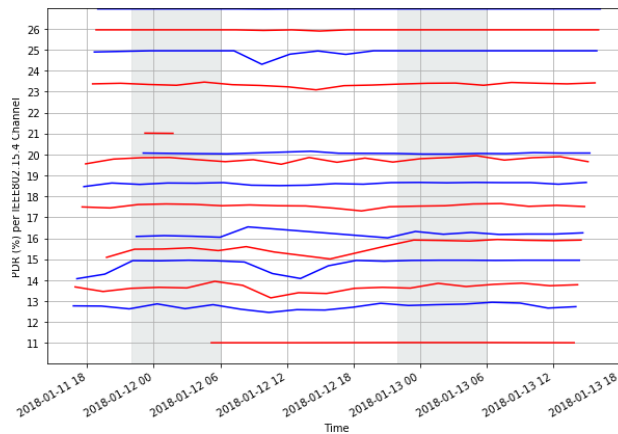Fig. 4: Confusion matrix with RSSI on all the Links.

Fig. 5: PDR evolution over time of $Link_1$.

- If the PDR is less than or equal to 0.3, we consider the link is of *Bad* quality;
- Otherwise ($0.3 < PDR < 0.75$), we consider that it is a link of *Intermediate* quality.

B. Predictions based on PDR for $Link_1$

Fig 6 presents the confusion matrices resulting from the prediction using LR, Linear SVM, SVM and RF on $Link_1$.

(a) Logistic regression

(b) Linear SVM Classifier
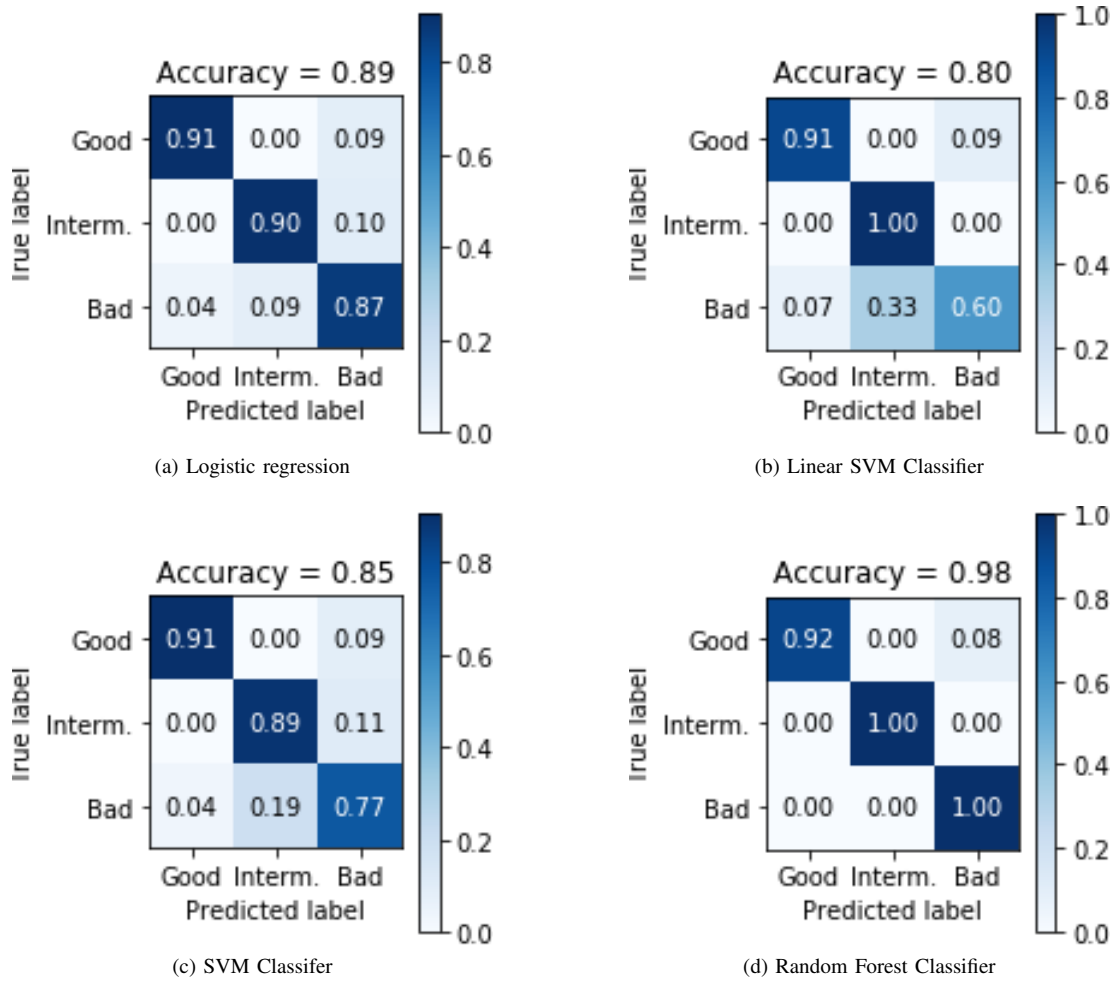
(c) SVM Classifer

(d) Random Forest Classifier

Fig. 6: Confusion matrix with PDR on $Link_1$.

As shown in Fig. 6, the Random Forest classifier has the highest accuracy followed by Logistic Regression, SVM and Linear SVM classifier, respectively.

C. Prediction based on PDR over all the links

Results of prediction over all the links, and based on PDR metric are presented in Fig 7.

The Random Forest classifier outperforms the other models with an accuracy of 99.3%. It is followed by Linear SVM (96.9%), SVM (96.4%) and Logistic Regression (95.3%), respectively.

Linear regression classifies up to 14.8% of good links as intermediate and 0.3% as bad ones, which can prevent the use of good links. Linear SVM classifies 8.2% of good links as bad and 4% of bad link as good (this prevents the use of good links whereas bad links can be used, leading to a decrease in QoE).

The Random Forest classifier is then recommended to use when making prediction using the PDR feature in TSCH networks. Although it has the best accuracy, Random Forest classifier predicts 2.8% of good links as bad and 0.9% of bad links as good. This eventually decreases the QoE offered by the network.

We believe that compared to the predictions based on RSSI, the predictions based on PDR are more reliable since PDR considers all the packets sent on the link, whereas RSSI only considers successfully received packets. It would also be interesting to investigate the predictions based on both PDR and RSSI to take advantages of both metrics. By doing so, we would be able to solve one of the challenges pointed by Baccour et al. [3], which said that predicting link quality is to find an optimal trade-off between the stability of the link quality estimator and the ability to cope with link quality dynamics.

VII. IMPROVED PREDICTION USING BOTH PDR AND RSSI

To have an efficient link quality estimation which provides fine grain classification of links, we jointly use the PDR and RSSI metrics. The goal is to study whether combining these two metrics, a software metric and a hardware metric, may provide

(a) Logistic regression

(b) Linear SVM Classifier
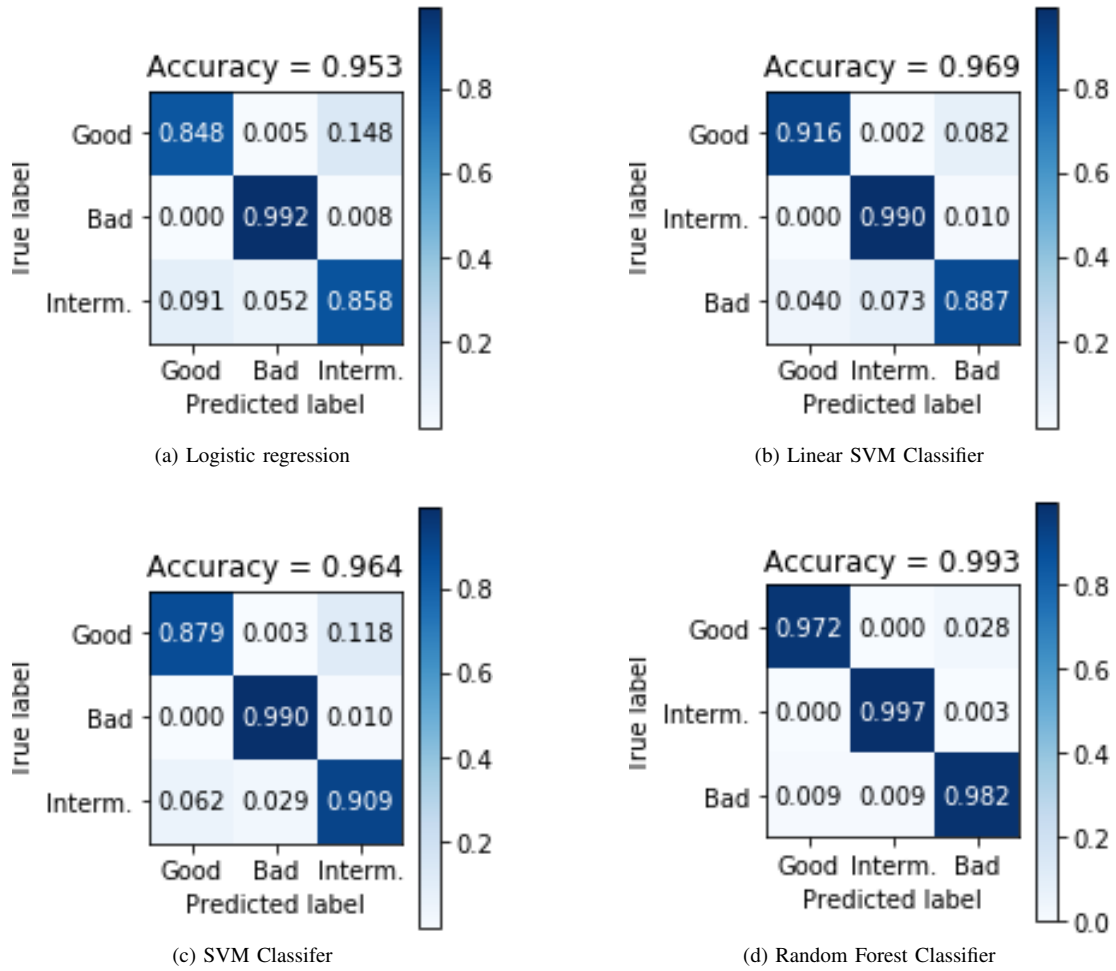
(c) SVM Classifer

(d) Random Forest Classifier

Fig. 7: Confusion matrix with PDR on all the Links.

a more accurate prediction. However, as the PDR metric takes into account all the sent packets and not only the received ones as RSSI does, the PDR metric is considered as the first criterion.

A. Definition of classes using PDR and RSSI

The classes we define for the prediction using the two metrics are given in Table II.

	$PDR \leq 0.3$	$0.3 < PDR < 0.75$	$PDR \geq 0.75$
$RSSI \leq -87$	Bad	Bad	Good
$-87 < RSSI < -85$	Bad	Intermediate	Good
$RSSI \geq -85$	Bad	Intermediate	Good

TABLE II: Classes defined using both PDR and RSSI.

- A link is of *Good* quality if its PDR is greater than or equal to 0.75;
- if its PDR is less than or equal to 0.3, the link is considered as being of *Bad* quality; It is also *Bad* if its RSSI is less than or equal to -87. This last case is the only difference with the classification based on PDR only. and its PDR is between 0.3 and 0.75.
- Otherwise, the link is considered as being of *Intermediate* quality.

B. Predictions based on both PDR and RSSI for $Link_1$

Results of predictions using LR, Linear SVM, SVM and RF are shown in Fig 8 for $Link_1$. The accuracy of RF is the highest. It is followed by SVM, Linear SVM and Logistic Regression, respectively.
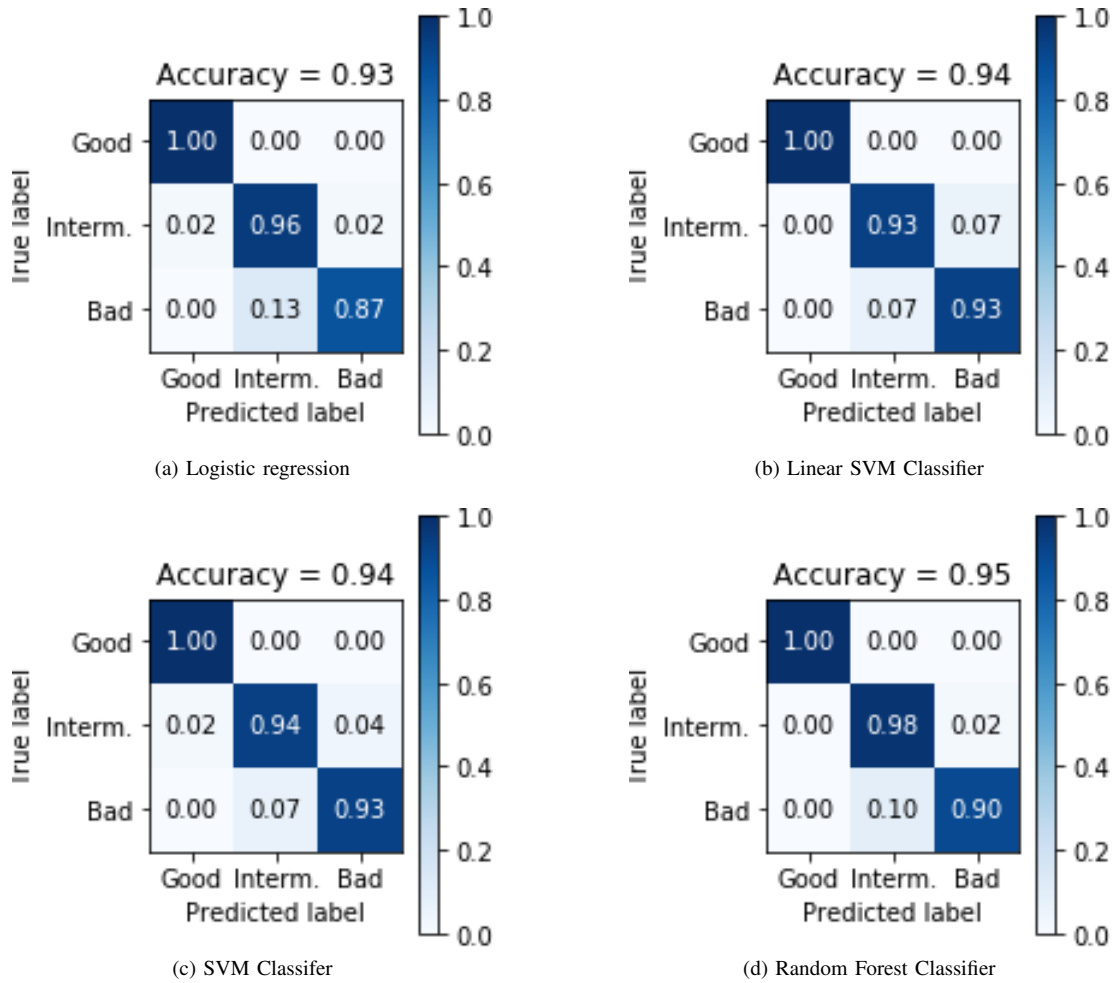
(a) Logistic regression

(b) Linear SVM Classifier

(c) SVM Classifer

(d) Random Forest Classifier

Fig. 8: Confusion matrix with PDR and RSSI on $Link_1$.

C. Prediction based on PDR and RSSI over all the links

Prediction on all the links using both PDR and RSSI has also been done, and results are presented in Fig 9.

With PDR+RSSI, RF has the highest accuracy (98.3%), followed by Logistic Regression (96.1%), SVM (95.8%) and linear SVM (94%), respectively.

From all the results presented in this paper, we conclude that among the ML techniques tested, RF provides the best accuracy, whatever the metric used. The question is now, which is the best metric to use with RF? RF with PDR predicts 2.8% of Good links as Bad whereas RF with PDR+RSSI predicts only 0.1% of Good links as Bad and does not predict any Bad link as Good. Such prediction errors may significantly decrease the QoE of the network, since Good links will not be used, while Bad links predicted Good will be used. Furthermore, Fig. 10 depicts the percentage of links per class. It shows that PDR+RSSI recognizes the highest number of Good links and the highest number of Bad links. That is why it is preferred, even if the accuracy of RF wih PDR (99.3%) is greater than the one using both PDR and RSSI (98.3%).

VIII. CONCLUSION

In this paper, we propose to predict link quality in TSCH networks supporting IoT applications. We studied two link quality estimators: a hardware one, RSSI, and a software one, PDR. We compared the accuracy provided by different machine learning techniques, namely Logistic Regression, Linear Support Vector Machine, Support Vector Machine and Random Forest classifier. Whatever the link quality estimator used, RSSI, PDR or both, the Random Forest (RF) classifier model outperforms the other models.

Since using *Bad* links that have been predicted *Good* strongly penalizes network performance in terms of end-to-end latency, end-to-end reliability and network lifetime, the joint use of PDR and RSSI improves the accuracy of link quality prediction. Hence, we recommend to use the Random Forest classifier applied on both PDR and RSSI metrics.
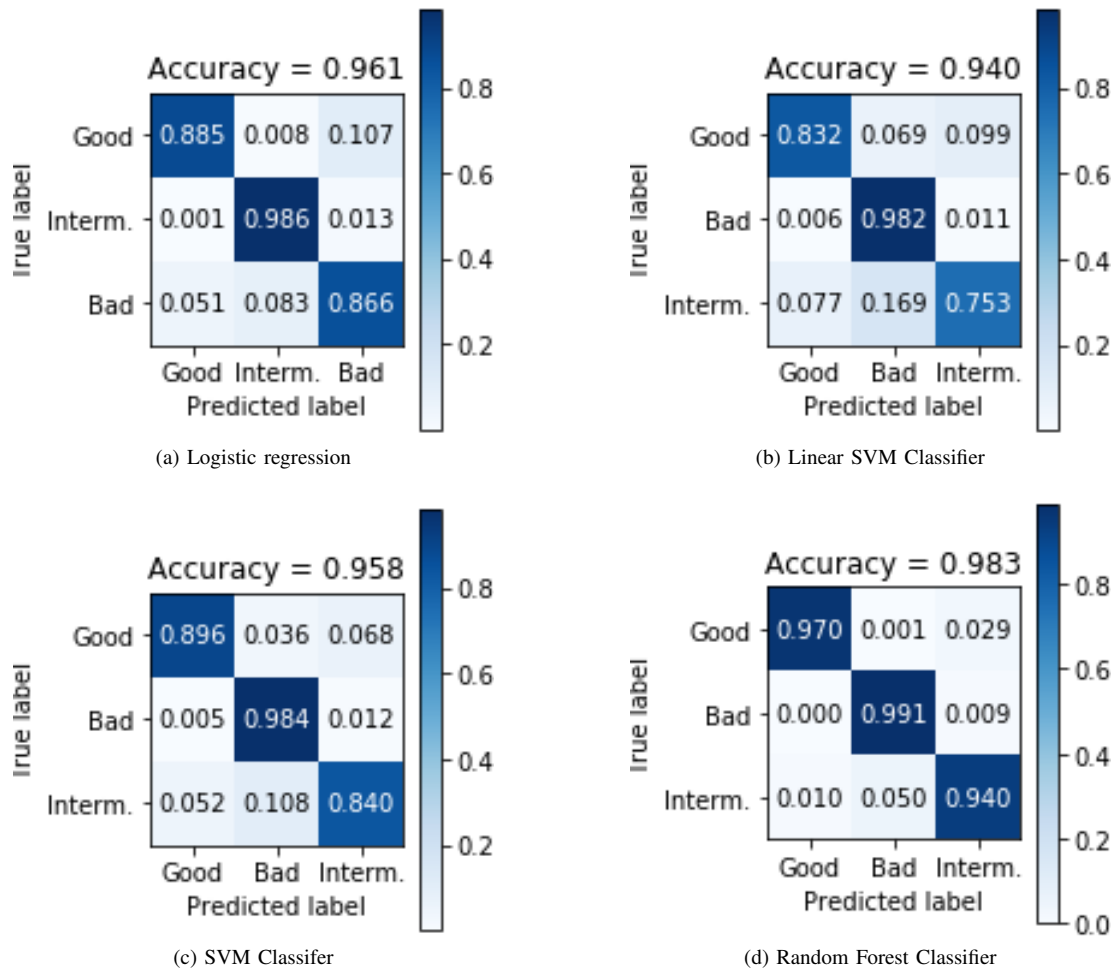
(a) Logistic regression

(b) Linear SVM Classifier

(c) SVM Classifer

(d) Random Forest Classifier

Fig. 9: Confusion matrix with PDR and RSSI on all the Links.
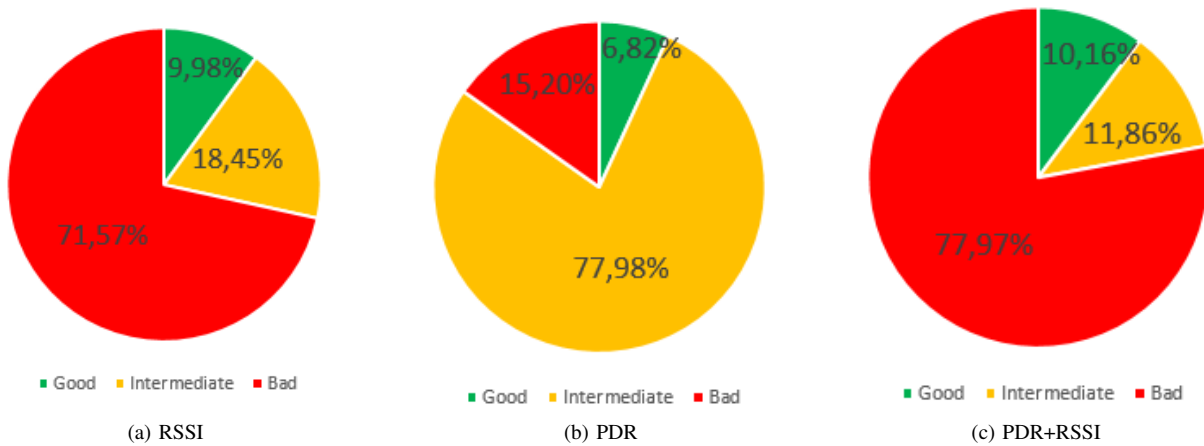

(a) RSSI

(b) PDR

(c) PDR+RSSI

Fig. 10: Percentage of links per class.

As a further work, we plan to use the link quality prediction presented in this paper to improve the clustering protocol of motes in an IoT network.

References

[1] D. Aguayo, J. Bicket, G. J. Biswas, and R. Morris, "Link-level Measurements from an 802.11b mesh network," in *acm sigcomm*, 2004, pp. 121–132.

[2] G. Kirubasri and N. Uma Maheswari, "A Study on Hardware and Software Link Quality Metrics for Wireless Multimedia Sensor Networks," *Int. J. Advanced Networking and Applications*, vol. 8, no. 3, pp. 3103–3109, Dec. 2016. [Online]. Available: www.ijana.in/papers/V8I3-7.pdf

[3] N. Baccour, "Radio link quality estimation in wireless sensor networks: a survey," *ACM Transactions on Sensor Networks (TOSN*, p. 35, 2012.

[4] "What is machine learning? a definition," https://www.expertsystem.com/machine-learning-definition/, accessed: 2019-05-27.

[5] M. Alexandros, Z. P. Georgios, F. Xenofon, E. Atis, O. George, and T. Theo, "Impact of guard time length on ieee 802.15.4e tsch energy consumption," in *IEEE International Conference on Sensing, Communication and Networking*, 2016.

[6] T. Chang, T. Watteyne, K. Pister, and Q. Wang, "Adaptative synchronization in multi-hop tsch networks," *Computer Networks*, vol. 76, pp. 165–176, 2015.

[7] K. Brun-Laguna, P. Henrique Gomes, P. Minet, and T. Watteyne, "Moving Beyond Testbeds? Lessons (We) Learned about Connectivity," *IEEE Pervasive Computing, Special Issue on Beyond Testbeds: Real-World IoT Deployments*, 2018, to appear in 2018.

[8] M. Venkatesh, "What is training data, really," https://blog.playment.io/what-is-training-data/#What_is_Training_Data, accessed: 2019-06-07.

[9] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "Four bit wireless link estimation," in *6th Int. Workshop on Hot Topics in Networks (HotNets VI)*, ACM SIGCOMM, 2007.

[10] C. Gomez, A. Boix, and J. Paradells, "Impact of lqi-based routing metrics on the performance of a one-to-one routing protocol for ieee 802.15.4 multihop networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 6, 2010.