Rethinking IoT Network Reliability in the Era of Machine Learning

Fafoutis, Xenofon; Marchegiani, Letizia

Published in:
Proceedings of the 12th IEEE International Conference on Internet of Things

Link to article, DOI:
10.1109/ithings/greencom/cpscom/smartdata.2019.00189

Publication date:
2019

Document Version
Peer reviewed version

Rethinking IoT Network Reliability in the Era of Machine Learning

Xenofon Fafoutis* and Letizia Marchegiani†
*Technical University of Denmark, Denmark
Email: xefa@dtu.dk
†Aalborg University, Denmark
Email: lm@es.aau.dk

***Abstract*—In the Internet of Things (IoT), wireless sensor networks are often paired with machine learning frameworks to deliver applications of high societal impact and support critical infrastructures. In this context, this paper investigates the relationship between network reliability and the reliability of the machine learning framework in terms of prediction accuracy. Our experimental analysis leverages six data sets of various degrees of information redundancy and considers four machine learning algorithms that are commonly used for classification. In turn, packet loss is inserted in the raw input data, emulating various networking loss patterns in terms of burstiness. The experimental results consistently demonstrate a non-linear relationship between the reliability of the network and the accuracy of the machine learning classifier, indicating that not all data packets are equally valuable to the application performance. We conclude with recommendations for IoT practitioners and IoT system designers.**

***Index Terms*—Reliability, Machine Learning, Missing Data, Internet of Things**

1. Introduction

The emerging Internet of Things (IoT) brings wireless sensing technology into various industries and application domains. Indeed, the IoT goes beyond smart consumer electronics, bringing wireless embedded systems in critical infrastructures, such as industrial sensor networks [1], health and care services [2], and city infrastructures [3], among others. The critical nature of these IoT application domains drives the need for dependable sensor networks. A vital component of dependable IoT networking is the reliability of the data communication network, expressed as the ratio of the packets delivered to the destination over the packets sent by the source: the Packet Delivery Rate (`PDR`). The order of magnitude of network reliability is often measured in terms of a number of *nines* that correspond to the long-term probability of successful delivery. Wired networks can offer very high degrees of reliability. For instance, there are reports that the IEEE 802.1 TSN (Time Sensitive Networking) standard [4] offers seven *nines* (99.99999%) of reliability [5]. Due to the nature of wireless communication, on the other hand, reliability is significantly more challenging in wireless networks. Indeed, the quality of a wireless link is very volatile, as it depends on several environmental parameters. For instance, wireless networks – particularly the ones deployed in the unlicensed bands – are prone to interference. In addition, link quality also depends on reflections (multi-path fading) and obstacles (shadowing).

IEEE 802.15.4e [6] is a recent addition to the IEEE 802.15.4 standard for low-power wireless networks. The amendment introduces TSCH (Time Slotted Channel Hopping), a time-synchronous Medium Access Control (MAC) protocol that traces its origins to industrial wireless standards, *i.e.*, WirelessHart [7] and ISA-100.11a [8], and aims to bring wire-like reliability to low-power networks. TSCH achieves this by keeping the nodes of the network globally synchronised and orchestrating the usage of the medium via schedules. TSCH schedules can, indeed, be completely free of collisions, if they allocate no more than one transmitter to a particular timeslot. In turn, external interference is avoided by channel hopping [9]. With channel hopping at the link-layer, upon a channel error, the re-transmission is scheduled at a different channel, mitigating the probability of consecutive errors that eventually lead to packet loss. Duquennoy *et al.* [10] have shown that TSCH can achieve five *nines* of reliability (99.999%) in various test-bed environments. Elsts *et al.* [11] adopted TSCH for a Health IoT sensing platform and achieved, on average, more than 99.9% reliability in a series of long-term residential deployments (up to 12 months) in a city environment. Similar works document real-world deployments that confirm that TSCH is able to achieve very high levels of reliability at outdoor environments [12] and smart buildings [13] as well.

In this paper, we attempt to identify the value of network reliability in IoT sensing applications, such as the aforementioned examples. We are particularly interested, and thus limit the scope of the paper, to IoT applications that consist of a IoT network of (typically resource-constrained) sensing devices that is paired with machine learning frameworks for knowledge extraction. In such IoT contexts, the IoT network is responsible for data acquisition and, in turn, the machine learning framework is responsible for extracting knowledge from the collected data. In this context, network reliability plays an important yet indirect role to the system performance; the ultimate performance indicator is the reliability of the prediction. In other words, the paper

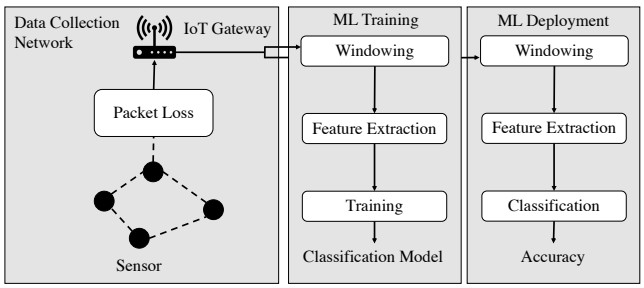

Figure 1. **System Architecture:** IoT sensing platforms generate raw data and communicate them to the IoT gateway over a low-power wireless network, whereby packet loss may occur. The raw data is, in turn, processed by a Machine Learning (ML) framework to train a model (training phase) and perform classifications (deployment phase). The reliability of the IoT network indirectly affects the accuracy of the classification task.

investigates how the reliability of the IoT network affects the reliability of the prediction at the application layer. It is the authors' hope and belief that a better understanding of this relationship can trigger cross-layer research on dependable wireless networking, leading to IoT networking protocols that are designed to exploit the mechanics of data-driven knowledge extraction for resource-efficiency.

In related work, [14] investigates the efficiency of classifiers using datasets with simulated data loss and proposes an evaluation methodology for missing data techniques. More specifically, their analysis focuses on random missing data on the feature set due to arbitrary reasons. Different to [14], our focus is particularly on data lost in the IoT data collection network, whereby the missing data patterns derive from the reliability of the network. Moreover, our analysis considers data loss in the raw data generated by the IoT sensing devices, and not on the extracted features.

In summary, the contributions of the paper can be summarised as follows. We analyse a typical IoT application in which raw data are generated by an IoT sensing device, collected over an IoT network, and provided to a machine learning framework for knowledge extraction (Section 2.1). In this setting, we consider six datasets of various degrees of information redundancy in terms of sampling frequency and number of sensing modalities (Section 3.1). In turn, we emulate various packet loss patterns at the data collection sub-system (Section 2.2), and we investigate how the reliability of the network affects the reliability of four commonly used classifiers (Section 3). Lastly, we conclude with recommendations for IoT practitioners and a discussion on how this relationship can be exploited to improve the resource-efficiency of reliable IoT networking protocols (Section 4).

2. Methodology

2.1. System Architecture

We consider a typical IoT system architecture, as shown in Fig. 1, whereby raw sensor data are generated by low-end sensing devices. The raw data is, in turn, transferred over a low-power wireless network to an IoT gateway device that lies at the root of the low-power network. Thereafter, the raw data are either processed at the IoT gateway, following the principle of Fog Computing, or transferred and processed at the cloud, following the principle of Cloud Computing. Regardless of whether it occurs at the IoT gateway or the cloud, the process of knowledge extraction follows a typical time-series machine learning processing chain: firstly, the stream of raw data is segmented into (potentially overlapping) windows; secondly, for each window, features are extracted from the raw data (feature extraction); lastly, the extracted features are provided as input to a machine learning algorithm either for training a model (training phase) or for making a prediction (deployment phase).

In this architecture, we consider that the process of data acquisition via the low-power IoT network is characterised by a certain degree of reliability. Without loss of generality, for the remainder of the paper, we will assume that each sensing device transmits the raw sensor data to the IoT gateway in packets that contain a single data sample (or a data sample from each sensing modality assuming the sensing node hosts multiple sensing elements). Therefore, the availability of the raw data depends on the reliability of the network, denoted as Packet Delivery Rate (PDR). For example, a PDR of 99.9% expresses that on average one out of 1000 samples is not available to the machine learning architecture.

It is stressed that, from the perspective of the user, the reliability of the network is not directly relevant. Instead, the user experiences the reliability of the application as a whole, expressed as the prediction accuracy of the machine learning model. Nevertheless, any data loss at the IoT network introduces error into the process of knowledge extraction, thus contributes to a reduction of the prediction accuracy of the machine learning model. On one extreme, with perfect reliability (PDR$= 1$), the machine learning algorithm has all the data available and, thus, can reach its maximum potential. On the other extreme, with no data (PDR$= 0$), the machine learning algorithm can only make a random guess. In a classification task among N classes, this dictates a lower bound in the prediction accuracy, $1/N$.

It is noted that a broadband link from the IoT gateway to the cloud is expected to be several orders of magnitude more reliable than the reliability of the low-power wireless network, due the natural challenges of wireless communication (*e.g.*, interference, fading, shadowing, etc). For that reason, we consider the probability of data loss in the link from the IoT gateway to the cloud negligible and we assume 100% reliability for that link. Therefore, the analysis of this paper covers both the cases of fog computing (knowledge extraction is located at the IoT gateway) and cloud computing (knowledge extraction is located at the cloud). We highlight, however, that the case of knowledge extraction on the sensing device, either in full or partially (*e.g.*, embedded feature extraction [15]), is out of the scope of this paper.

Lastly, this paper focuses on the worst case scenario whereby both the training data and the testing data are

TABLE 1. VALUES OF PROBABILITIES p AND q USED IN EXPERIMENTS

	PDR	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$b = 0$	p	0.99	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0
	q	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$b = 0.3$	p	0.693	0.63	0.56	0.49	0.42	0.35	0.28	0.21	0.14	0.07	0
	q	0.007	0.07	0.14	0.21	0.28	0.35	0.42	0.49	0.56	0.63	0.7
$b = 0.6$	p	0.396	0.36	0.32	0.28	0.24	0.2	0.16	0.12	0.08	0.04	0
	q	0.004	0.04	0.08	0.12	0.16	0.2	0.24	0.28	0.32	0.36	0.4
$b = 0.9$	p	0.099	0.09	0.08	0.07	0.06	0.05	0.04	0.03	0.02	0.01	0
	q	0.001	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1

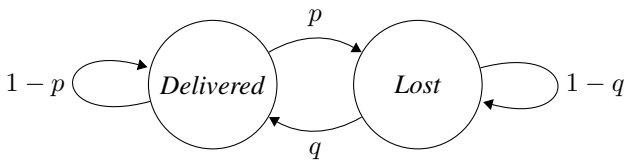

Figure 2. Packet loss is generated by a two-state Markov model.

characterised by data loss during data collection. Here, one may also consider the case whereby the training data are collected via expensive or impractical, yet reliable, means and only the deployment phase is characterised by packet loss. We consider the latter an interesting direction for future work.

2.2. Packet Loss Insertion Model

The goal of this paper is to study how network reliability affects the reliability of a classification framework. To that end, we employ a complete data set that has no data loss, and we insert artificial packet loss based on the packet loss insertion model that is described in this section.

We insert artificial packet loss that follows a two-state Markov model, also known as Gilbert model [16]. This model is frequently used in the literature to simulated packet loss in streaming data due to its effectiveness and simplicity [17], [18]. The two-state Markov model, shown in Fig. 2, operates as follows. The system is in either in the *Delivered* state or in the *Lost* state. In the former state, the data packet is delivered successfully to the IoT gateway. In the latter state, the data packet is lost in transit. With each transmission the system is characterised by a probability p to change from the *Delivered* state to the *Lost* state and a probability q to change from the *Lost* state to the *Delivered* state.

It can be observed that if $p = 1 - q$, the probability of a packet loss does not depend on its previous state, therefore models an environment whereby the packet losses are statistically independent, such as due to collisions or short-term interference. Yet, the effectiveness of the two-state Markov model derives from the fact that it is able to capture the burstiness of packet loss in communication networks. Indeed, if $p < 1 - q$ the probability of a packet loss is greater after another packet loss rather than after a successful delivery. This effectively models packet loss that is bursty by nature, such as long-term interference or hardware failures. For the remainder of the paper, we define a burst factor, $b = 1-p-q$. A burst factor $b = 0$ inserts packet losses that are statistically independent (*i.e.*, $p = 1 - q$). A burst factor $b > 0$ is characterised by a higher probability to insert consecutive packet losses. At extreme levels ($b \to 1$), the probability of a loss is very low ($p \to 0$), yet once a loss occurs the probability that a consecutive packet loss is very high ($q \to 0$), resulting to prolonged bursts of packet loss. The overall packet delivery rate of the two-state Markov model is given by:

$$\texttt{PDR} = 1 - \frac{p}{p+q} \tag{1}$$

In the experiments that follow (Section 3), we compare different burst factors b at the same PDR values. Table 1 summarises the values of the probabilities p and q used in the experiments.

3. Experimental Analysis

3.1. Application Use Case and Dataset

For the experimental analysis, we consider a use case from the Health IoT domain, namely classification of activities of daily life using wearable sensors. This is a commonly studied problem with numerous applications, such as assisted living for the elderly and long-term behavioural analytics for individuals that suffer from chronic illness [19].

Specifically, we employ a dataset collected using an early prototype of the third generation of the SPHERE Wearable [20]. The wearable device employs two inertial sensing elements: the MC3672 accelerometer [21] and the ICM20948 operating as a gyroscope [22]. The two sensing components are placed on the prototype board with their respective x, y and z axes aligned. The accelerometer was configured to operate in the ± 8 g range of acceleration amplitude and at 12-bit resolution, which translates to 3.9 mg sensitivity. The gyroscope was configured to operate at 16-bit resolution, resulting to a sensitivity of 7.6 mds^{-1}. The sensors are sampled at 18 Hz. In is noted that the literature suggests that the accuracy of activity classification improves with increased sampling frequency, yet with no significant gains above 20 Hz [19], [23], [24].

The dataset is composed of a set of nine loosely scripted activities of daily life performed by seven volunteers, aged

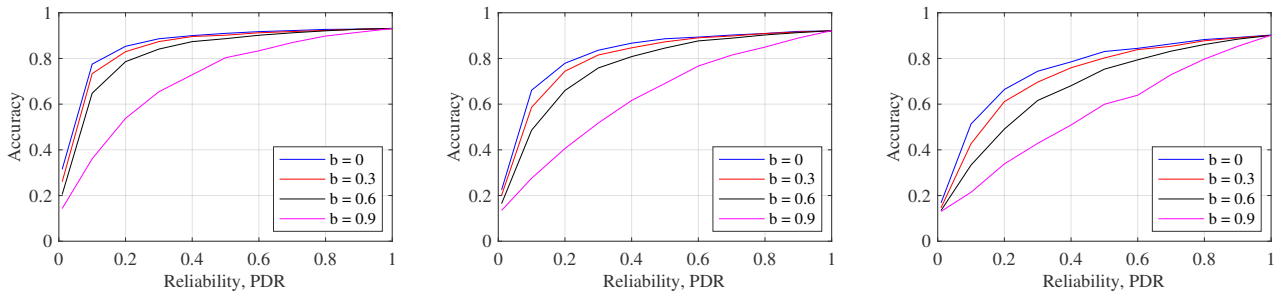

Figure 3. Network reliability vs. classification accuracy (**RF**) using only the accelerometer at various sampling frequencies: 18 Hz (left), 9 Hz (middle), 4.5 Hz (right).

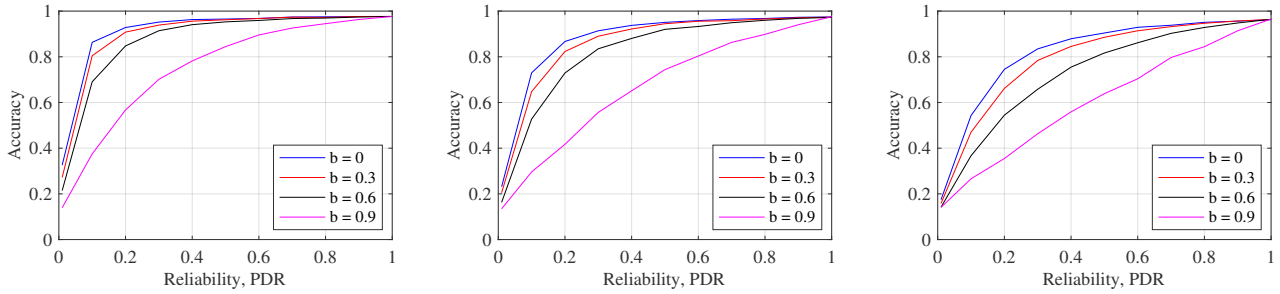

Figure 4. Network reliability vs. classification accuracy (**RF**) using both the accelerometer and the gyroscope at various sampling frequencies: 18 Hz (left), 9 Hz (middle), 4.5 Hz (right).

between 23 and 36 years, 3 females and 4 males. The nine activities of daily life are the following: sitting, standing, walking, turning leftwards, turning rightwards, running, jumping, and exercising. The participants were instructed to repeat these activities for 2.5 minutes. In an attempt to maximise the variability in the dataset, each participant was let free to interpret how to perform each activity, without any particular instructions. The dataset was collected by Zalewski, who documents it in further detail in [25].

Aiming to investigate the effect of network reliability on the classification accuracy using datasets of various degrees of information redundancy, we generate several input datasets out of the original, focusing on two dimensions, namely the number of sensing modalities and the sampling frequency. In particular, we consider the case of severely resource-constrained wearable sensors that, for energy-efficiency, employ only an accelerometer and the case whereby both an accelerometer and a gyroscope are employed. Moreover, we downsample the respective datasets by a factor of two and by a factor of four, generating additional datasets that emulate lower sampling frequencies: namely 18 Hz, 9 Hz and 4.5 Hz (including the original).

Before passing the data to the machine learning frameworks, we insert data loss using the two-state Markov model (see Section 2.2) configured with the probabilities p and q as shown in Table 1.

3.2. Machine Learning Framework

The machine learning framework takes as input the six datasets with missing data due to packet loss, as described in Section 3.1.

Initially, the data is segmented into windows of 1.6 seconds. This window size is in line with the conclusion of studies that investigate the effect of different window sizes on classifying activities of daily life [19], [26]. In addition, aiming to capture the temporal nature of the activities, a 50% overlapping window is used, as in [15]. This configuration results to a total of approximately 9000 windows (approximately 1000 windows for each of the 9 activities).

In the next step, statistical features are extracted from each window. In particular, we extract features from the temporal domain, as there is literature that suggests that these features provide a good balance between classification accuracy and resource requirements [15], [27]. These include the following: (i) maximum; (ii) median; (iii) minimum; (iv) mean; (v) variance; (vi) standard deviation. These temporal features are extracted for each of the three axes of the accelerometer and gyroscope respectively, resulting to a total number of 18 features for the accelerometer-only datasets and 36 features for the accelerometer and gyroscope datasets. If the window misses accelerometer/gyroscope samples due to packet loss, the features are calculated normally, ignoring the missing values. In total, approximately 1000 feature vectors are extracted from the input data for each of the 9 activities.

In turn, each of the 9 arrays of feature vectors is randomly split into training and testing sets (80%-20%). This ensures balanced representation of the 9 activities in the training and testing sets. The training set is used to train a classifier and the testing set is used to evaluate its performance in terms of classification accuracy. It is noted

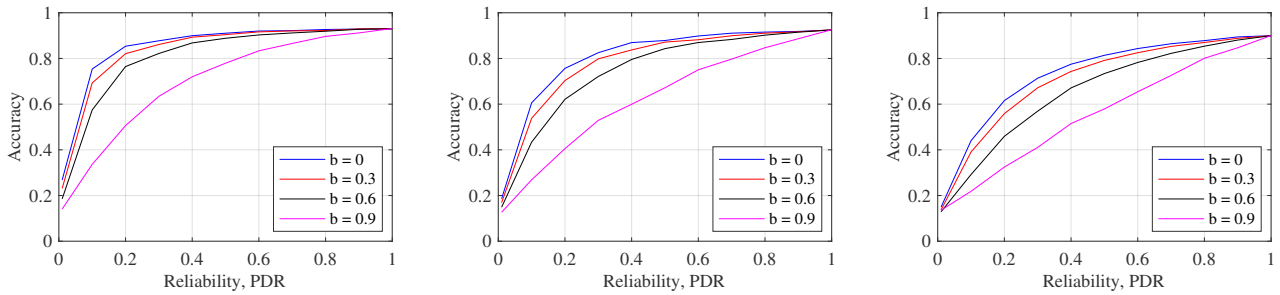

Figure 5. Network reliability vs. classification accuracy (**KNN**) using only the accelerometer at various sampling frequencies: 18 Hz (left), 9 Hz (middle), 4.5 Hz (right).

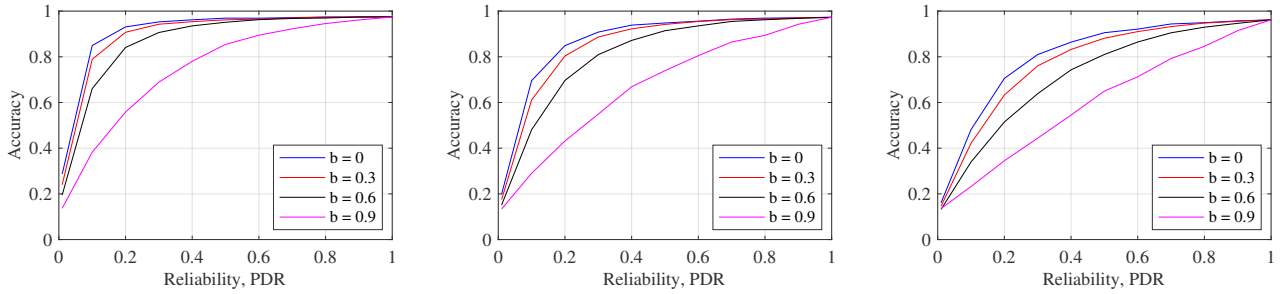

Figure 6. Network reliability vs. classification accuracy (**KNN**) using both the accelerometer and the gyroscope at various sampling frequencies: 18 Hz (left), 9 Hz (middle), 4.5 Hz (right).

that the probability to randomly identify the correct class is approximately 11.11%. This constitutes a lower performance bound for the classification framework.

In this work we study how network reliability affects the performance of the classifier, and we are interested to investigate if different classification frameworks are affected differently by packet loss in the data collection network. To that end, we employ four machine learning algorithms for the classification task, namely Random Forest (RF), k-Nearest Neighbours (KNN), Support Vector Machines (SVM), and Deep Neural Networks (DNN). In the remainder of this section, we briefly introduce the machine learning algorithms and provide further details on their configuration.

Random Forest (RF) [28] is a statistical learning framework which generates a set of decision trees from randomly selected subsets of the training data, and provides a classification which combines the decisions of each single tree. We train a random forest of 20 trees with a minimum leaf size of 1.

k-Nearest Neighbours (kNN) [29] is a non-parametric classification framework (no assumptions are made on the distribution of the data) based on a majority vote scheme, where a sample is assigned to the class most common among its k nearest neighbours. Several measures of distance/nearness are possible, depending on the context. Our classifier is using the euclidean distance and $k = 5$ to make a prediction. The input features are centred to their mean and scaled to their standard deviation.

Support Vector Machine (SVM) [30] is a supervised classification framework which aims to find the hyperplane that maximises the distance between samples belonging to different classes. It is intrinsically a binary classifier, but techniques are available to operate in multi-class environments. We implement a multi-class SVM classifier as a series of 9 one-against-all binary classifiers. We employ the Radial Basis Function (RBF) kernel. The input features are centred to their mean and scaled to their standard deviation.

Deep Neural Network (DNN) [31] is a neural network with more than two layers, composed of nodes (*i.e.*, neurons) characterised by a specific activation function. We use a DNN with two hidden layers of 18 nodes, equipped with hyperbolic tangent sigmoid transfer functions. The output layer relies on a softmax function. The training is carried out by minimising the cross-entropy cost function.

3.3. Experimental Results

The experimental results are presented in Figs. 3 to 10. Each figure incorporates three plots that correspond to the three sampling frequencies, namely 18 Hz, 9 Hz, and 4.5 Hz, resulting to a total of 24 plots that correspond to all the combinations of the six datasets (Section 3.1) and the four machine learning algorithms (Section 3.2). In turn, each plot shows the relationship between the reliability of the data acquisition network (PDR) and the reliability (*i.e.*, classification accuracy) of the machine learning framework that operates at the application layer. Each line in the plot corresponds to a different packet loss pattern. On one extreme a burst factor $b = 0$ represents the case whereby packet losses are statistically independent (*e.g.*, fading, short-term interference, short-term shadowing). On the other extreme, a burst factor $b = 0.9$ represents the case whereby packet loss

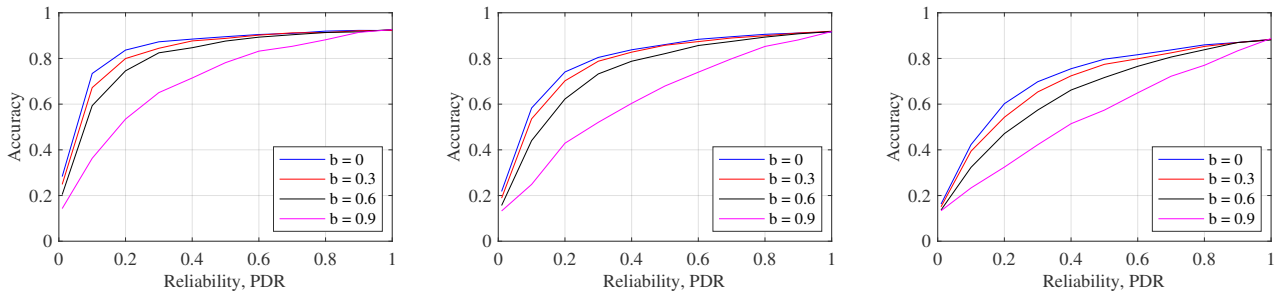

Figure 7. Network reliability vs. classification accuracy (**SVM**) using only the accelerometer at various sampling frequencies: 18 Hz (left), 9 Hz (middle), 4.5 Hz (right).

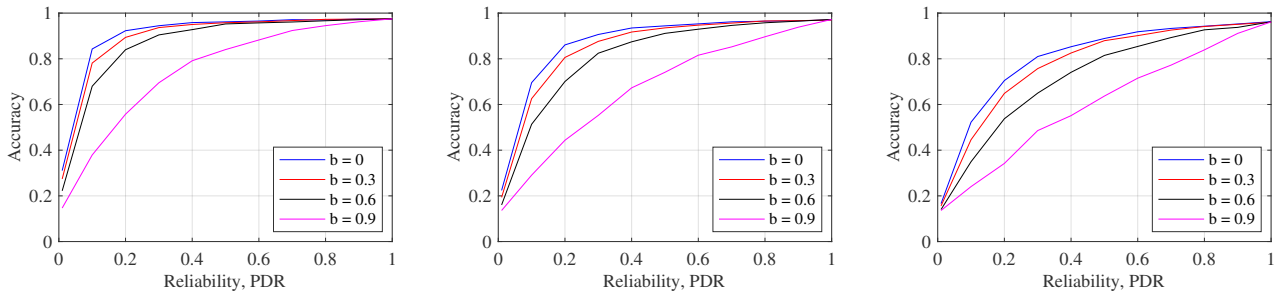

Figure 8. Network reliability vs. classification accuracy (**SVM**) using both the accelerometer and the gyroscope at various sampling frequencies: 18 Hz (left), 9 Hz (middle), 4.5 Hz (right).

events are rare, yet once they happen several consecutive packets are lost (*e.g.*, reboots, hardware failures). The intermediate burst factors represent cases whereby there is some level of correlation between packet losses (*e.g.*, extended periods of interference). Each experiment was repeated up to 100 times and the figures report the *mean* classification accuracy. The standard deviation (not shown in the figures) was $\sigma < 0.02$ for all cases with two exceptions: $\sigma = 0.0224$ for DNN with accelerometer-only at 18 Hz dataset and $b = 0.9$, and $\sigma = 0.0366$ for DNN with accelerometer-only at 9 Hz dataset and $b = 0$.

Looking at all 24 plots, we can observe that the relationship between network reliability and classification accuracy has a hyperbolic nature. One one end, increasing the network reliability has a positive effect on the reliability of knowledge extraction, yet this improvement has diminishing returns: the accuracy of the classifier eventually plateaus at some maximum level. On the other end, as fewer data samples reach the machine learning framework, the accuracy of the classifier eventually collapses and asymptotically reaches the level of a random guess (*i.e.*, approximately 0.11 in our case). This trend, which is consistent in all datasets and regardless the employed machine learning algorithm or the packet loss pattern (*i.e.*, burst factor b), comes in contradiction with the fairly common assumption of the IoT networking community that network reliability has a linear relationship with application performance, *i.e.*, all data packets are of equal value. Our experiments suggest that this is not true for applications that are based on data-driven machine learning frameworks.

With regard to the influence of different packet loss patterns, we can observe that the machine learning frameworks are tolerant to packet loss as long as the lost data samples are distributed in time. Indeed, the lower the burst factor b, the higher the tolerance. However, when a critical amount of data within a window is lost, the effectiveness of the whole window is compromised. Indeed, when the burst factor b is very high, there is a very high probability for consecutive errors that invalidates multiple windows. As a result, the accuracy of the classifier collapses more rapidly. As an intuitive example, consider the case of a relay node in the IoT network that reboots due to a software error and reconnects after one minute. In this scenario, the classifier is unable to operate reliably during that minute. If, on the other hand, the packet loss is uncorrelated and thus distributed in time over multiple windows, the classifier is able to operate more reliably. In future work, we plan to investigate if frequency-domain features lead to a similar behaviour.

As anticipated, the experimental results also demonstrate that higher information redundancy in the input data makes the system more resilient to packet loss. Indeed, the accelerometer-only dataset is less resilient to packet loss than the case that leverages both sensing modalities, and the same holds for the sampling frequency. In the hypothetical scenario whereby the input data have no information redundancy whatsoever, it is easy to imagine that the importance of network reliability would be vital for the application performance. However, in practice, this would never be the case. Indeed, while techniques are available to for feature selection and dimensionality reduction, such as Principal Component Analysis (PCA) [32], or to rank features according to the their impact on classification performance

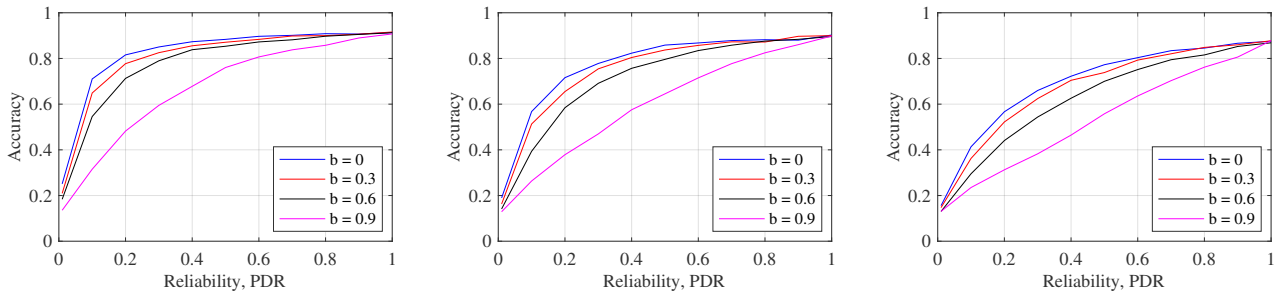

Figure 9. Network reliability vs. classification accuracy (**DNN**) using only the accelerometer at various sampling frequencies: 18 Hz (left), 9 Hz (middle), 4.5 Hz (right).

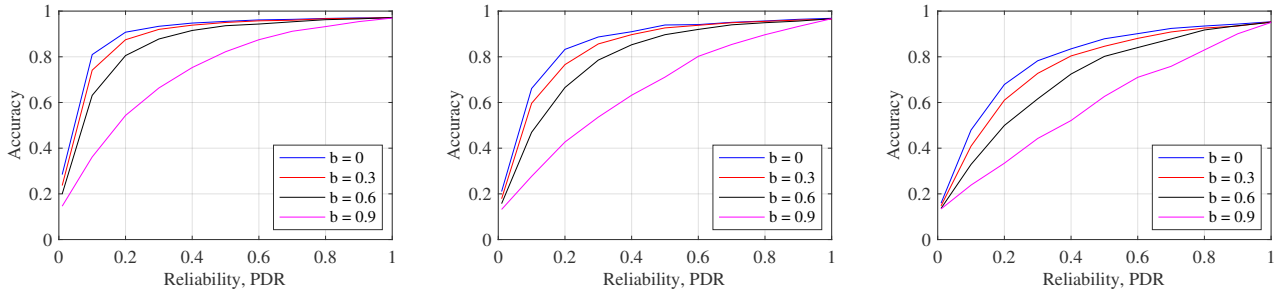

Figure 10. Network reliability vs. classification accuracy (**DNN**) using both the accelerometer and the gyroscope at various sampling frequencies: 18 Hz (left), 9 Hz (middle), 4.5 Hz (right).

e.g., [33], these kinds of techniques can be applied only once a subset of data has been collected and used to elaborate potential models and statistics. Redundancy reduction can be obtained at the training phase, but cannot be used to drive training data collection, on per sample basis; in practice, sensor data are typically collected with a high degree of redundancy. For example, Khan *et al.* [34] examined several datasets, collected for accelerometer-based human activity recognition, and concluded that the sampling frequencies can be reduced by at least 43% and the reduced data would still be more than 99% similar to the original data.

A final observation on the experimental results is that the accuracy of the knowledge extraction is never perfect; the maximum it reaches is 97.5% even when the IoT network is 100% reliable. This is indeed a general observation; perhaps with the exception of toy problems, it is very unlikely that machine learning applications yield more than 99% prediction accuracy. This suggests that, beyond a certain degree of network reliability, the machine learning framework constitutes the performance bottleneck.

4. Conclusion

In this paper, we focus on IoT applications that employ machine learning algorithms to extract knowledge from data that originates from a low-power wireless network of IoT sensing devices. In this context, we investigate how the reliability of the IoT network affects the accuracy of machine learning framework at the application layer. Our experimental results suggest that the relationship between network reliability and classification accuracy is of hyperbolic nature. In particular, as network reliability increases, classification accuracy also increases yet with diminishing returns. Similarly, as network reliability decreases, there is a collapsing point, beyond which the accuracy rapidly drops. These trends are confirmed by all examined scenarios, which investigate four machine learning algorithms on datasets of various sensing modalities and sampling frequencies.

The experimental results, presented in this paper, suggest that not all data packets have equal value, as often assumed in the IoT networking literature. (It is noted that techniques for traffic differentiation, whereby important packets are prioritised over best-effort packets (*e.g.*, [35], [36]) are impractical in our case, as they require *a priori* knowledge on the priority level of each data sample.) This non-linear relationship can be exploited for resource-efficiency. Indeed, network reliability has a cost either in terms of energy consumption or in terms of bandwidth when the IoT network operates close to saturation levels (*e.g.*, [37]). In such contexts, protocols can adapt parameters that control such performance trade-offs in ways that target to maximise the classification accuracy rather than the PDR. In future work, we plan to tailor TSCH schedules for data-driven machine learning applications.

We conclude with some design recommendations for IoT practitioners: (i) Unless the process of sensing at the end devices is optimised for efficiency, the machine learning framework is fairly resilient to packet loss. In such cases, 90% network reliability is likely to be sufficient. If the sensors are optimised and redundancy is low, there is less resilience to packet loss and, thus, 99% network reliability is recommended. (ii) Long packet loss bursts have much

higher impact on the application performance than sporadic packet loss. Mitigating sources of packet loss bursts (*e.g.*, reboots, routing path failures, etc) with appropriate redundancy should be prioritised. (iii) Since the accuracy of machine learning predictions rarely exceeds 99%, the end users of such IoT applications would hardly expect flawless operation. Therefore, employing ultra-reliable networking solutions (>99.9%) is unnecessary, especially if it compromises efficiency and maintenance costs.

References

[1] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov 2014.

[2] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. Kwak, "The internet of things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[3] C. A. Medina, M. R. Prez, and L. C. Trujillo, "Iot paradigm into the smart city vision: A survey," in *2017 IEEE International Conference on Internet of Things (iThings)*, June 2017, pp. 695–704.

[4] N. Finn, "Introduction to time-sensitive networking," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 22–28, JUNE 2018.

[5] Analog Devices, "Time sensitive networking: A silver bullet for the industrial internet of things?" Technical Article, 2018.

[6] "IEEE Standard for Local and Metropolitan Area Networks Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC Sublayer," IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011), April 2012.

[7] "WirelessHART Specification 75: TDMA Data-Link Layer," Std., Rev 1, HART Communication Foundation, 2008.

[8] ISA-100.11a-2011:, "Wireless systems for industrial automation:process control and related applications," *International Society of Automation (ISA) Std.*, vol. 1, May 2011.

[9] T. Watteyne, A. Mehta, and K. Pister, "Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense," in *Proceedings of the ACM PE-WASUN*, 2009, pp. 116–123.

[10] S. Duquennoy, J. Eriksson, and T. Voigt, "Five-nines reliable downward routing in RPL," *CoRR*, vol. abs/1710.02324, 2017.

[11] A. Elsts, X. Fafoutis, P. Woznowski, E. Tonkin, G. Oikonomou, R. Piechocki, and I. Craddock, "Enabling healthcare in smart homes: The sphere iot network infrastructure," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 164–170, December 2018.

[12] T. Watteyne, A. L. Diedrichs, K. Brun-Laguna, J. E. Chaar, D. Dujovne, J. C. Taffernaberry, and G. Mercado, "PEACH: Predicting Frost Events in Peach Orchards Using IoT Technology," *EAI Endorsed Transactions on the Internet of Things*, Jun. 2016.

[13] K. Brun-Laguna, A. L. Diedrichs, D. Dujovne, C. Taffernaberry, R. Lone, X. Vilajosana, and T. Watteyne, "Using smartmesh ip in smart agriculture and smart building applications," *Computer Communications*, vol. 121, pp. 83 – 90, 2018.

[14] S. Karadogan, L. Marchegiani, L. Hansen, and J. Larsen, "How efficient is estimation with missing data?" in *IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 2260–2263.

[15] A. Elsts, R. McConville, X. Fafoutis, N. Twomey, R. Piechocki, R. Santos-Rodriguez, and I. Craddock, "On-board feature extraction from acceleration data for activity recognition," in *Embedded Wireless Systems and Networks (EWSN)*, 2018, pp. 163–168.

[16] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell System Technical Journal*, vol. 39, no. 5, pp. 1253–1265, 1960.

[17] H. Toral, D. Torres, and L. Estrada-Vargas, "Simulation and modeling of packet loss on alpha-stable voip traffic," *Recent Advances in Signals and Systems*, 2009.

[18] M. Ellis, D. P. Pezaros, T. Kypraios, and C. Perkins, "A two-level markov model for packet loss in udp/ip-based real-time video applications targeting residential users," *Computer Networks*, vol. 70, pp. 384 – 399, 2014.

[19] N. Twomey, T. Diethe, X. Fafoutis, A. Elsts, R. McConville, P. Flach, and I. Craddock, "A comprehensive study of activity recognition using accelerometers," *Informatics*, vol. 5, no. 2, 2018.

[20] X. Fafoutis, A. Vefas, B. Janko, R. S. Sheratt, J. Pope, A. Elsts, E. Mellios, G. Hilton, G. Oikonomou, R. Piechocki, and I. Craddock, "Designing wearable sensing platforms for healthcare in a residential environment," *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 3, no. 2, pp. 1–11, 2017.

[21] mCube Technical Staff, *MC3672 3 - Axis Accelerometer Datasheet*, 2017.

[22] TDK Technical Staff, *ICM-20948 World's Lowest Power 9-Axis MEMS Motion Tracking Device*, TDK IvenSense, 2017.

[23] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[24] U. Maurer, A. Smailagic, D. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Int. Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, 2006.

[25] P. Zalewski, "Embedded machine learning for physical activity recognition," University of Bristol, 2018.

[26] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014.

[27] C. Erdas, I. Atasoy, K. Acici, and H. Ogul, "Integrating features for accelerometer-based activity recognition," *Procedia Computer Science*, vol. 98, pp. 522 – 527, 2016.

[28] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001.

[29] B. W. Silverman and M. C. Jones, "E. Fix and J.L. Hodges (1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951)," *International Statistical Review / Revue Internationale de Statistique*, vol. 57, no. 3, pp. 233–238, 1989.

[30] C. Cortes and V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[31] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359 – 366, 1989.

[32] I. Jolliffe, *Principal component analysis*. Springer, 2011.

[33] L. K. Hansen, S. Karadogan, and L. Marchegiani, "What to measure next to improve decision making? on top-down task driven feature saliency," in *2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)*. IEEE, 2011.

[34] A. Khan, N. Hammerla, S. Mellor, and T. Pltz, "Optimising sampling rates for accelerometer-based human activity recognition," *Pattern Recognition Letters*, vol. 73, pp. 33 – 40, 2016.

[35] X. Fafoutis, C. Orfanidis, and N. Dragoni, "Altruistic backoff: Collision avoidance for receiver-initiated mac protocols for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 10, no. 5, p. 576401, 2014.

[36] M. Arifuzzaman, M. Matsumoto, and T. Sato, "An intelligent hybrid mac with traffic-differentiation-based qos for wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 6, pp. 2391–2399, June 2013.

[37] A. Elsts, X. Fafoutis, J. Pope, G. Oikonomou, R. Piechocki, and I. Craddock, "Scheduling High-Rate Unpredictable Traffic in IEEE 802.15.4 TSCH Networks," in *13th Int. Conf. on Distributed Comput. in Sensor Syst. (DCOSS)*, 2017.