

Predicting Round-Trip Time Distributions in IoT Systems using Histogram Estimators

Christofer Flinta*, Wenqing Yan[†], and Andreas Johnsson*[†]

* Ericsson Research, Sweden, Email: {christofer.flinta, andreas.a.johnsson}@ericsson.com

[†] Uppsala University, Department of Information Technology, Sweden, Email: {wenqing.yan, andreas.johnsson}@it.uu.se

Abstract—In this paper we describe and evaluate an approach for predicting conditional RTT probability distributions in an IoT system. From the distributions we derive conditional mean and quantiles, for example, which are essential for performance management and service assurance. The distribution is represented by a histogram, which requires a discretized target space, trained using supervised learning of a random forest classifier.

We evaluate the approach using data traces obtained from experimentation in a realistic IoT testbed. The results show high model performance in prediction of quantiles and aggregated distributions, and the trends for conditional mean are captured.

For the operator, the approach enables low-overhead and tractable IoT performance assessment, especially compared to traditional approaches using for example active measurements.

Index Terms—Network Management, Internet of Things (IoT), Conditional Density Estimation (CDE), Machine Learning.

I. INTRODUCTION

Internet of Things (IoT) is an emerging technology that comes with great opportunities in health care, digitalized industry such as mining and manufacturing, and smart homes. The purpose of an IoT system is to monitor, and sometimes control, an environment or process using multiple wireless sensors and actuators. These are connected via an infrastructure to a customer or end user. See Figure 1 as a schematic example, where a set of wireless sensors/actuators are connected to the infrastructure via a wireless gateway. The wireless infrastructure and its devices need to be managed to ensure reliability and robustness. However, many management challenges are still only partly addressed, including low-overhead performance assessment and observability of the network [1].

Telecommunication operators are accustomed to use Internet Protocol (IP) active measurements, such as ICMP [2] or TWAMP [3][4], for observing and measuring network performance, often with respect to a Service Level Agreement (SLA). They expect that new technology, such as IoT, provides similar and low-overhead tools for performance assessment, since this is the foundation for anomaly detection and troubleshooting. This in turn, lowers the costs for operations while providing a smoother service for the consumers.

In previous work [5] we developed and evaluated an active measurement tool based on TWAMP for IoT devices. The main contribution of that work was an in-depth discussion of the necessary trade-offs between active-measurement frequency and overhead in terms of power consumption and network load. To compensate for the active measurement

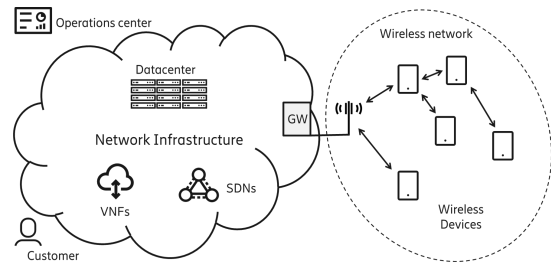


Fig. 1. An IoT system includes devices, wireless sensor networks (WSN), gateways (GW), a network infrastructure, operators, and customers.

overhead we developed a novel approach based on machine learning for predicting measurement results, using data features collected from the IoT system [6]. The prediction error of round-trip time (RTT) and loss were relatively high, and one contributing factor was fluctuations in the wireless network that were not fully captured by the features.

In this paper, which is a continuation of our previous work, we take another modeling approach aiming at capturing the network complexity, where we go beyond predicting point values for RTT, and instead predict performance distributions conditioned on the network state, which can be used for visual inspection and calculation of important statistics such as mean and quantiles. Specifically, we adapt and evaluate a histogram-based approach that is applied to prediction of RTT conditional distributions. The distribution is represented by a discrete histogram which is trained using supervised learning of a random forest classifier, following the approach proposed in [7], and further developed in [8].

This paper makes the following contributions. We describe how the histogram-based approach for estimation of discretized conditional distributions is applied to IoT performance assessment. Further, we show through extensive experimentation that the conditional probability distribution is accurately estimated. Network complexity and fluctuations are manifested as long-tail RTT distributions. Aggregated distributions, and quantiles are estimated with high accuracy, while the conditional mean derived from the distribution remains a challenge. For modeling we use realistic data traces obtained from our IoT testbed at Uppsala University. Statistics from the network and devices are passively collected, whereas RTT is obtained using our TWAMP implementation for IoT devices.

From an operator perspective, the approach is highly useful,

for example in deriving probabilistic bounds for RTT, and other performance metrics. This is a powerful tool when assessing risks associated with SLAs.

The remainder of the paper is organized as follows. Section II gives a background and related work. Section III describes the problem setting and approach. Section IV describes the experimental setup and data traces, and Section V provides the evaluation. A discussion is located in Section VI, whereas Section VII provides conclusions and future work.

II. BACKGROUND AND RELATED WORK

The performance of a network can be quantified using metrics such as one-way delay, RTT, delay variation, and packet loss. A formal definition of these metrics is available in ITU-T Y.1540 [9]. Many telecom operators use active measurements, with protocols based on packet reflection, for metric assessment. Measurement packets are sent from a controller to a reflector which sends back response packets to the controller. Time stamps are generated at the transmit and receive events on both sides, as in IETF TWAMP [3][4] or only at the sender side, as in IETF ICMP [2]. In TWAMP each probe packet is time stamped 4 times, enabling estimation of RTT, one-way delay in both directions if clocks are in sync, and the reflector processing time.

Several studies have described strategies for reducing the overhead associated with active measurements in IoT environments. For example, in [10] the authors describe a light-weight approach for active measurements in heterogeneous IoT networks consisting of both IEEE 802.11 and IEEE 802.15.4 networks. They discuss RTT measurements, and propose a device communication technology identification approach. Further, active measurements have also been used for bandwidth estimation in IEEE 802.15.4 networks. For example, an enhanced approach considering MAC layer overhead for accurate estimation results is given in [11].

Another approach for reducing measurement overhead, that is gaining traction in the network management community, is to estimate the network performance metrics using other available data from the network, and leveraging techniques from statistics and machine learning. For example, in our previous work [6] we described an approach based on supervised learning for predicting RTT and loss, based on sensor data and routing statistics, in an IoT environment. Other examples include [12] where the throughput of a wireless sensor network is modeled using measurements of radio receiver signal strength indicators (RSSI). Further, in [13] the RTT is modeled as a linear function of the distance, omitting environmental dynamics and routing. A more extensive review of other predictive techniques for IoT is available in [14].

A common denominator in existing work is that the predictions provide point estimates of continuous variables such as RTT and loss. However, network operators are interested in additional information describing the distribution of the performance metric, such as quantiles, conditioned on the current network state. The quantiles of interest can also vary over time and scenario.

A simple, computationally efficient, and effective approach for estimation of the conditional distribution is described in [7]. Specifically, the approach (often called histogram-based) finds the discretized conditional density using class probabilities, that are estimated using supervised machine learning. From the conditional density they calculate various statistics such as the conditional mean. The approach is further developed in [8] where class probability estimation is combined with univariate normal and kernel density estimators for discretized conditional density estimation. Extensive experimentation with multiple data sets show improvements in many cases.

Further, online-learning approaches to estimation of discrete and also continuous conditional densities are proposed in [15] [16]. The approaches are designed to overcome challenges related to, for example, streaming data of which they would like to model joint or conditional densities for. In this work we take the baseline approach to perform discretized conditional density estimation where we rely on histograms and a random forest classifier. Future work will study whether the refined methods are beneficial for our scenario.

In [17] the authors describe an algorithm for conditional density estimation based on random forest regression, where the regression tree algorithm has been modified in several steps to accommodate the requirements of density estimation. The authors do not provide an extensive evaluation, however they do provide an open source library. Future work will study whether their approach proves beneficial in our case.

Further, works on conditional density estimation for network and service management have recently been published. Specifically, the authors of [18] describe a method for estimating continuous conditional distributions of service metrics (e.g. response time for a data base service, and frame rate for a VoD service) using mixture density networks based on Gaussian kernels. A demonstrator of that work was later described, together with co-authors of this paper, in [19].

III. PROBLEM SETTING AND APPROACH

This section describes the problem setting and the details of the approach for predicting discretized conditional distributions of RTT using a histogram estimator.

Figure 1 shows the system under consideration. In an IoT system, a set of devices are connected via an IoT gateway to a network infrastructure. The devices are assumed to execute a service, such as collecting data via sensors or affecting their environment via actuators. The service operates under strict SLAs with requirements on network performance.

In this context we model the RTT for the path between two devices, or a device and the gateway, as a random variable Y . Further, features from a device i , $i = 1..s$, are modeled as random variables X^i , and includes device position, signal quality, sensor output, and routing data, for example. The device statistics X is the concatenation of all device statistics, that is $X = [X^1, X^2, \dots, X^s]$.

Both X and Y evolve over time, influenced by the network load, radio dynamics, the terrain, and movements. This evo-

lution is modelled as discrete time series. Features X may, due to practical reasons, not be collected exactly at the same time as Y_t , and in this case X is binned over a time interval containing the time stamp of Y_t .

In previous work [6] we have developed machine-learning models for predicting a point value Y from the device statistics X . In this paper we go beyond predicting point values for RTT, to accommodate for fluctuations and uncertainty stemming from the IoT system, by adopting a formalism for learning a discretized conditional density model $p_{\text{model}}(y|x)$ from empirical data $\hat{p}_{\text{data}}(y|x)$. The empirical data is sampled from the true, but hidden, distribution $p_{\text{data}}(y|x)$. Important statistics such as mean and quantiles can then be derived from $p_{\text{model}}(y|x)$. Benefits of this approach include the possibility of determining probabilistic bounds, for example, which is critical in understanding risks associated with SLAs.

A. Modeling conditional distributions using Histograms

The approach taken in this paper discretizes the target space Y by modelling the empirical conditional density function $\hat{p}_{\text{data}}(y|x)$ using a histogram estimator, as described in [7] [8]. We assume that $\hat{p}_{\text{data}}(y|x)$ is non-zero on a continuous interval of the target space, for all x . We call this interval the *histogram interval*. It is partitioned into C bins with bin width r_c , where $c \in [1, C]$. Each bin is interpreted as representing a class c . Class probabilities $f(c_y|x)$, where c_y is the class containing target value y within bin c , are estimated using a multi-class classification algorithm. For this work, we have chosen a Random Forest model [20].

For a given x , the conditional density function $p_{\text{model}}(y|x)$ over the histogram interval is calculated by spreading out each class probability evenly within its bin, namely,

$$p_{\text{model}}(y|x) = \frac{f(c_y|x)}{r_{c_y}} \quad (1)$$

where r_{c_y} is the bin width of class c containing y . This density function can be understood as a step function over the histogram interval with a constant density within each bin. The density is set to zero in the open intervals to the left and right of the histogram interval. Numerically, $p_{\text{model}}(y|x)$ can be represented as a normalized vector with C components, or in other words a normalized histogram.

In this paper, the histograms are created with a bin width r_c that is constant for all bins. Some bins may not cover any of the y values in the training set. In such cases, we set the probabilities (and thus the densities) for these bins to zero. This is particularly important when r_c is small.

B. Estimation of conditional mean

To calculate the conditional mean of $p_{\text{model}}(y|x)$ over the histogram interval, we use the estimated class probabilities $f(c_y|x)$ as a probability mass function and assume this function to be constant over the bin interval. The conditional mean

of the target is calculated as the weighted mean over the class mid-points

$$\begin{aligned} E[Y|X] &= \int_Y p_{\text{model}}(y|x) y dy = \int_Y \frac{f(c_y|x)}{r_{c_y}} y dy \\ &= \sum_{c=1}^C (f(c_y|x) \frac{1}{r_{c_y}} \int_{Y_c} y dy) = \sum_{c=1}^C f(c_y|x) \bar{y}_c \end{aligned} \quad (2)$$

where Y_c is the interval of bin c , and $\bar{y}_c$ is the arithmetic mid-point between the left and right bin edge. Note that the two open intervals outside the histogram interval do not contribute to the expectation, since p_{model} is zero in those parts.

C. Estimation of quantiles

In general, the quantile a of a probability density function $p(y)$ is defined by $q = \Pr(y \leq a) = \int_{y=-\infty}^a p(y) dy = P(a)$, where $P(a)$ denotes the cumulative distribution function (cdf) of $p(y)$ and $q \in [0, 1]$.

Thus, given q , our aim is to estimate $a = a_{\text{data}}(x, q)$ for $p_{\text{data}}(y|x)$ by computing $a_{\text{model}}(x, q)$. Since the cdf $P_{\text{model}}(x, a)$ is strictly monotonic in a we can compute the inverse mapping and obtain $a_{\text{model}}(x, q)$.

For a histogram, the cdf $P_{\text{model}}(x, a)$ is 0 to the left of the histogram interval and 1 to the right. Inside the interval, it has a constant slope within each bin. The quantile value $a_{\text{model}}(x, q)$ is estimated by first finding the bin its value falls into and then interpolating within that bin.

Numerically, we calculate a vector P_c mapped to the bin edges b_c as a cumulative sum over the predicted class probabilities $f(c_y|x)$ from left to right, where $P_0 = 0$ at the left-most bin edge b_0 and $P_C = 1$ at the right-most bin edge b_C :

$$\begin{aligned} P(y|x)_0 &= 0 \\ P(y|x)_c &= P(y|x)_{c-1} + f(c_y|x) : 1 \leq c \leq C \end{aligned} \quad (3)$$

Then, $a_{\text{model}}(x, q)$ is estimated by finding the largest $k \in [1, C]$ with $P(y|x)_k < q$, and interpolating between the bin edge b_k and the next edge b_{k+1} :

$$a_{\text{model}}(x, q) = b_k + (b_{k+1} - b_k) \frac{q - P(y|x)_k}{P(y|x)_{k+1} - P(y|x)_k} \quad (4)$$

An illustration with respect to $a_{\text{model}}(x, q)$ is provided in Section V which provides evaluation results.

D. Aggregated distribution over a time interval

We also want to estimate the distribution of aggregated density functions for different x over a time interval. Partly for evaluation purposes, but mainly as a tool for a monitoring and understanding characteristics of the network.

For histograms the set of bin intervals is independent of x . The aggregated density can therefore be calculated separately as an average for each bin

$$p_{\text{agg}}(y_c) = \frac{1}{m} \sum_{t=1}^m p_{\text{model}}(y_c|x_t) \quad (5)$$

where m is the sample size within the time interval of interest.

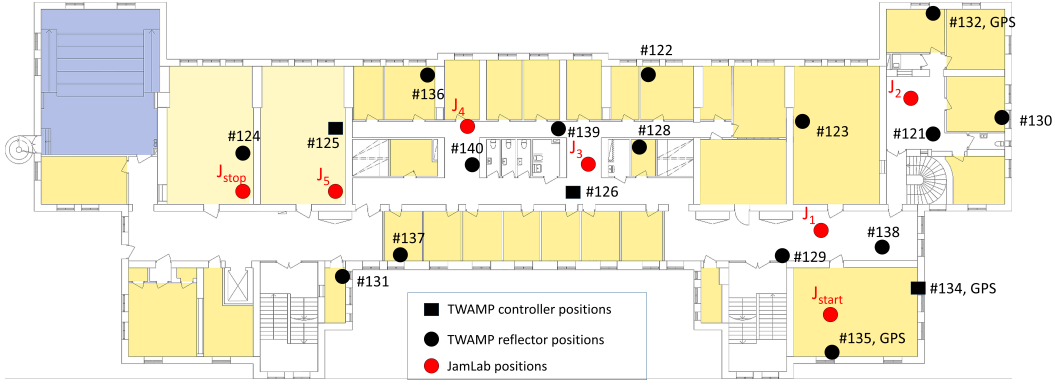


Fig. 2. Placement of motes in the testbed, the TWAMP controller positions, and JamLab positions.

IV. EXPERIMENTAL SETUP AND TRACES

A. Testbed

The EWSN'17 testbed [21] at Uppsala University has been used to generate data traces (X, Y) for evaluation of the discretized conditional density estimation approach.

The testbed consists of 18 Tmote Sky Motes¹. A mote (or IoT device) is a tiny but complete execution environment including CPU, memory, radio, antenna, battery, and sensors for temperature, humidity, and light. The mote placement is shown in Figure 2. The motes reside in a building that constitutes a challenging environment for wireless communication due to thick stone walls and rebars. There is also an uncontrolled impact from faculty members and students, and their wireless devices. The motes communicate over IPv6 enabled by IETF 6LoWPAN [22], running on top of IEEE 802.15.4. All motes in the testbed are configured to run the default Contiki Radio Duty Cycle and MAC protocols, providing low-power communication and operation as described in IETF 6LoWPAN. Further, IETF RPL [23] enables a dynamic tree-shaped routing topology relying on transient network performance. The root of the tree is denoted *RPL root*.

For the purpose of experiment administration and monitoring, separate monitoring modules, connected to an administrative central node over a fixed network, based on Raspberry Pi 3 are attached to each mote. The modules are used for easy and large-scale reprogramming of motes, power profiling, and high-precision event detection, without interfering with the motes nor the wireless medium during execution of experiments. They are connected to the motes via USB to enable logging and reprogramming. Three monitoring modules are connected to GPS time synchronized clocks, marked with "GPS" in the figure, enabling precise and accurate time stamping. This capability has been used in [5] to verify the device time stamping precision and accuracy. Further, that work also provides more details of the testbed.

¹<http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>

TABLE I
EXPERIMENTS AND PARAMETERS.

Exp. ID	TWAMP Ctrl.	Exec. Time	JamLab pos.
1	#125	144h	-
2	#126	144h	-
3	#134	144h	-
4	#126	24h	J_{start}
5	#126	24h	J_{stop}
6	#126	7h	$J_{start} \rightarrow J_{stop}$

B. TWAMP Active Measurements for Contiki

Our IETF TWAMP Light [3] tool for Contiki 3.0 environments², the predominant OS for IoT devices, was used to enable measurements of RTT between the motes. A mote either acts as a TWAMP controller or a reflector. The TWAMP controller is placed on one of the motes [125, 126, 134], see Figure 2. Note that there is only one controller for a given experiment. The testbed is configured to co-locate the TWAMP controller and the RPL root functionality.

The TWAMP controller conduct measurements towards the reflectors in a round-robin fashion. The probe-packet interval, that is the time between two consecutive measurements for a controller-reflector pair, is set to 6 seconds to reduce measurement interference. The value is found through experimentation and is valid for this specific testbed. Further, for each controller-reflector pair, in each round, the measurements are conducted 5 consecutive times. The probe-packet size is 100 bytes, including packet headers, payload, and preambles.

C. Radio interference

The testbed is mounted in a lively radio environment with several sources of noise and interference. In addition, we use *JamLab* [24] to create additional controlled radio interference. JamLab is a system that can record and playback interference patterns as well as generate customizable and repeatable interference in real-time. The system is executed on a separate Tmote Sky device.

²<http://http://www.contiki-os.org/>

Three scenarios are investigated; a JamLab mote at position J_{start} , at position J_{stop} , and moving the JamLab device from J_{start} to J_{stop} at discrete times, with intermediate positions J_i , see Figure 2. The JamLab device generates interference by emulating video streaming, at a high transmission power. No other traffic, except for TWAMP packets, is carried over the WSN testbed during the experiments.

We choose to inject noise only at the radio layer, mainly because we do not expect large volumes of cross traffic in a commercially deployed IoT network. Rather, we expect mainly controlled application traffic between devices and the gateway.

D. Experiments

Six data traces are used for evaluation of the discretized conditional density estimation approach. Each data trace consists of a time series of features X_t and targets Y_t , in this case RTT. The traces are outputs from testbed experiments, summarized in Table I, and briefly described below.

Experiments 1-3 ran for 6 days each, and were executed without JamLab. The objective is to study the behaviour stemming from three topologies; two trees with TWAMP controllers co-located with the RPL root on motes 125 (to the left in Figure 2) or 134 (right), and one star-like topology with the controller and RPL root at mote 126 (middle).

In experiment 4 and 5 the JamLab device was stationary at either J_{start} or J_{stop} . The TWAMP controller and RPL root resided on mote 126, since such star-like topologies are common in many industrial settings for example. These experiments were executed for 24h each, to limit disturbance from JamLab in the building hosting the testbed.

The last experiment, with identifier 6, considered a scenario where the JamLab device moved from position J_{start} to J_{stop} via the intermediate positions shown in Figure 2. The JamLab device stayed at each position for 1h. This experiment lasted for 7h, also in order to limit disturbances.

E. Data traces

Data traces in terms of features X and targets Y are collected from the testbed as described above. The data X and Y forms the empirical distribution $\hat{p}_{data}(y|x)$ that is used to train and evaluate $p_{model}(y|x)$. The features X are summarized in Table II.

The first set of features (x_{con} , y_{con} , x_{ref} , y_{ref} , D) relates to the position of the TWAMP controller and reflector, respectively, as well as the distance between them. Time of day T and weekday W are also recorded for each data point. Further, the IP-addresses, IP_{con} and IP_{ref} , of the TWAMP controller and reflector are collected.

Three features describe the conditions at the controller, namely the radio channel R_{CH} , link quality indicator R_{LQI} , and received signal strength indication R_{RSSI} . Further, the traces contain RPL rank observed at each device $[r_1, \dots, r_k]$, and the number of neighbours within radio reach for a specific device $[n_1, \dots, n_k]$. Light radiation registered by the controller is denoted in L . A more in-depth description of the features is found in [6]. The target metric Y corresponds to RTT between devices, measured using TWAMP.

TABLE II
FEATURES X COLLECTED FROM THE TESTBED.

Features	Description
x_{con}, y_{con}	Position of the controller
x_{ref}, y_{ref}	Position of each reflector
D	Distance (m) between controller and each reflector
T	Time of day, a categorical feature $\in [0, 5]$
W	Weekday, a categorical feature $\in [0, 6]$
IP_{con}, IP_{ref}	IPv6 addresses for controller and reflector
R_{CH}	Radio Channel, an ordinal feature $\in [1, 16]$
R_{LQI}	LQI measured at the controller
R_{RSSI}	RSSI measured at the controller
$r_1, \dots, r_{k-1}, r_k$	RPL rank for k motes
$n_1, \dots, n_{k-1}, n_k$	Number of neighbors for k motes
L	Light radiation registered by the controller

V. EVALUATION

This section provides an evaluation of the discretized conditional density estimation approach, described in Section III, using the traces obtained from the IoT testbed, described in Section IV. The objective is to assess the predictive performance with respect to RTT conditional mean, RTT quantiles, and aggregated distributions. First, however, a few notes on methodology, features, parameters, and implementation.

The models are trained using features in Table II. However, *to make the models independent on the number of devices in the network*, which is important due to the dynamics of an IoT system, we only include the RPL rank and number of neighbors for the reflector nodes. Future work may include other statistics derived from $r_{1..k}$ and $n_{1..k}$.

Further, for evaluation of the models we split each data trace in a train and test set, respectively, with a 70/30 ratio. Data points for each set are randomly selected.

The class-probability estimator corresponding to Equation 1 is implemented using a random forest multi-class classifier with C bins. A hyper-parameter search suggests that a max tree depth of 7, and 7 trees in the forest provides high performing models that avoids over-fitting. The same parameter values are used for all models across experiments. The tuning details are omitted due to page limitations.

A set of parameters defining $p_{model}(y|x)$ needs to be fixed before training. The bin width r_c is set to 0.1 second, corresponding to approximately 100 bins in the histogram.

Further, the lowest bin edge value in all histograms is set to zero, since the minimum RTT values for all traces are very close to zero. The highest bin edge value for each trace is set to y_{max} (corresponding to maximum RTT found in the training set) rounded up to nearest integer. In all our experiments the highest bin edge turns out to be 10 seconds.

Note that for a test set there is no guarantee that all y values are within the histogram interval created from the training set. Instead of omitting y values outside of the interval, they are mapped to either the lowest bin or the highest bin, respectively.

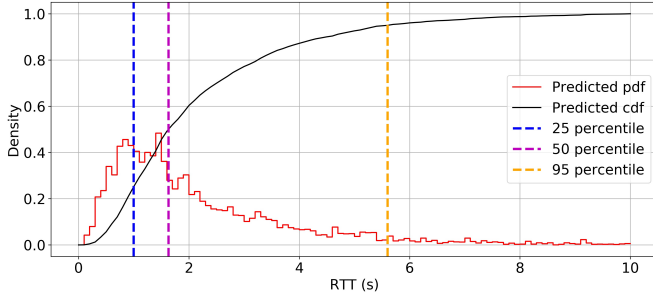


Fig. 3. Visualization of the pdf $p_{\text{model}}(y|x_t)$, cdf $P_{\text{model}}(y|x_t)$, and three quantiles, for a specific data point x_t in Experiment 1.

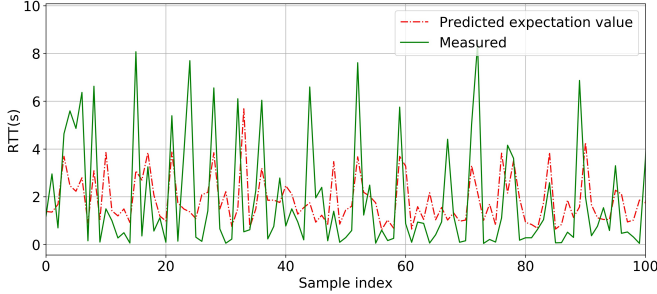


Fig. 4. Comparison of predicted and measured conditional mean (RTT) using data from Experiment 6.

Finally, the models are implemented using Python Scikit, and the evaluation is executed on a HP Elitebook 840 G5. Model training time is approximately 1 second per trace, while calculating the evaluation metrics takes additional 1-2 seconds.

A. Example of histogram estimation

The output of a trained discretized conditional distribution model $p_{\text{model}}(y|x)$ is a histogram with C number of bins, each with bin width r_c , as described in Section III.

Figure 3 provides a visualization of the output from the model, for a given x_t from Experiment 1. The predicted pdf is shown together with the corresponding cdf. Three percentile values are marked by vertical lines. We can see, for instance, that the 95th percentile of the interpolated cdf is about 5.6 seconds, whereas the median is close to 1.5 seconds.

It should be noted that the model does not need to be retrained in case other quantiles are of interest, it is rather a matter of recalculating Equation 4, where the complexity scales with C .

B. Predicting RTT Mean Values

We use two metrics to evaluate the model accuracy in predicting mean values; the *Normalized Mean Absolute Error* (NMAE), and *Coefficient of Determination* R^2 , defined as

$$NMAE = \frac{1}{\bar{y}} \left(\frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i| \right) \quad (6)$$

TABLE III
ESTIMATION ERRORS FOR HISTOGRAM-BASED CONDITIONAL MEAN COMPARED TO A STANDARD RANDOM FOREST REGRESSOR.

Experiment ID	Histogram Estimator		Random Forest Regressor	
	NMAE (%)	R^2	NMAE (%)	R^2
1	58.5	0.377	56.1	0.421
2	57.7	0.338	56.6	0.377
3	55.6	0.459	53.4	0.495
4	50.5	0.696	50.2	0.701
5	53.2	0.624	50.3	0.655
6	73.3	0.232	69.8	0.208

$$R^2 = 1 - \left(\frac{\sum_{i=1}^m (y_i - \hat{y}_i)^2}{\sum_{i=1}^m (y_i - \bar{y})^2} \right) \quad (7)$$

where $\hat{y}_i$ is the expectation of $p_{\text{model}}(y|x_i)$, y_i is the true value, and $\bar{y}$ is the average of the observations y_i of the test set of size m . $R^2 = 1$ means perfect prediction and $R^2 = 0$ corresponds to the accuracy of a naive estimator that just predicts the sample mean.

Having the trained histogram model $p_{\text{model}}(y|x)$ we calculate the mean value of RTT using Equation 2. We also compare the performance of $p_{\text{model}}(y|x)$ with a standard random forest regression model, which was also used in our previous work [6]. The results are available in Table III.

The NMAE for the histogram approach spans the range from 50.5% in experiment 4, up to 73.3% in the most complex scenario corresponding to experiment 6. Also, the R^2 results are correlated with the NMAE results, for all 6 experiments. Notable is that the R^2 results indicate that the models perform better than a naive estimator ($R^2 > 0$), specifically for experiments 4 and 5. Further, the histogram-based approach shows similar results to, and hence does not improve over, the standard random forest model.

To aid interpretation of the high errors we turn the attention to Figure 4 that shows predicted and measured RTT (data points from Experiment 6). A key observation is that the predicted RTT follows the trend of the measured RTT, however misses the amplitude of the spikes, mainly. An error analysis indicates that this is the main contributing factor to the error values observed in Table III. Similar observations have been made for the other experiments. We further elaborate on possible explanations in Section VI.

C. Predicting RTT Quantiles

Evaluating quantiles derived from $p_{\text{model}}(y|x)$ is more challenging since there is no explicit ground truth for each data point x_t . Instead, we make use of the Glivenko-Cantelli Theorem which provide means for an unbiased and consistent estimator of q , denoted $\hat{q}$, and evaluate over the entire test set (see e.g. [18]). The estimator is defined as

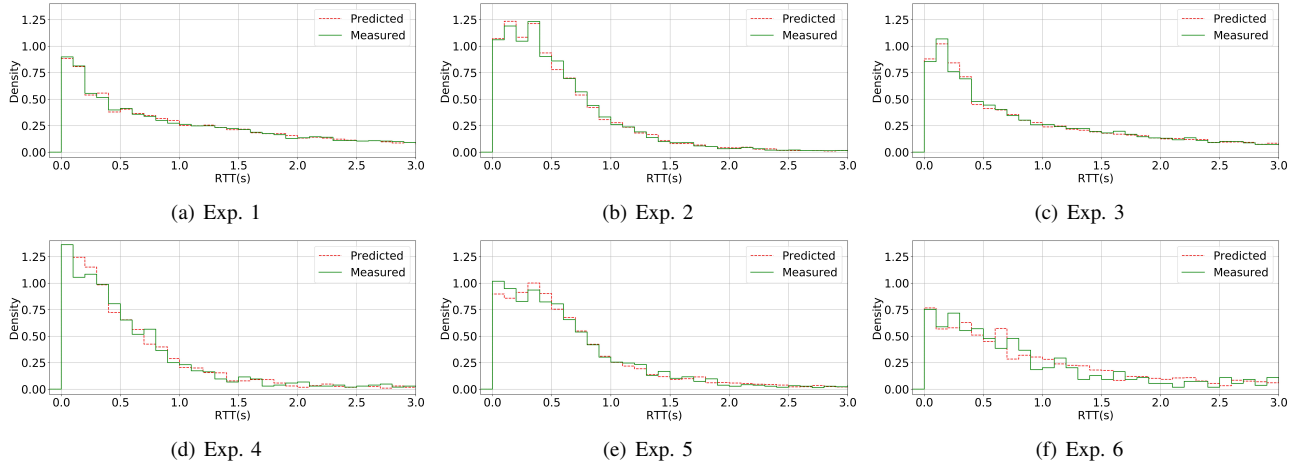


Fig. 5. Visualization of empirical and aggregated conditional distributions. Note that the distributions are cut at 3 seconds for increased readability.

TABLE IV
ESTIMATED $\hat{q}$ FOR RTT COMPARED TO THE SELECTED q -VALUES 0.25, 0.50, AND 0.95.

Experiment ID	q -values		
	0.250	0.500	0.950
1	0.247	0.505	0.951
2	0.238	0.506	0.956
3	0.240	0.505	0.959
4	0.244	0.460	0.927
5	0.270	0.525	0.958
6	0.261	0.500	0.947

TABLE V
ASSESSMENT OF AGGREGATED DISTRIBUTIONS USING HISTOGRAM INTERSECTION AND KULLBACK-LEIBLER DIVERGENCE

Experiment ID	D_{HI}	D_{KL}
1	0.969	0.185
2	0.972	0.406
3	0.965	0.223
4	0.910	0.339
5	0.924	0.273
6	0.817	0.216

$$\hat{q} = \frac{1}{m} \sum_{t=1}^m 1\{y_t \leq a_{model}(x_t, q)\} \quad (8)$$

where 1 is the indicator function, and m the sample size. The estimator is computationally efficient.

To evaluate the prediction accuracy with respect to quantiles, we use the estimator in Equation 8 to obtain $\hat{q}$, and compare the results with the selected q -values of interest, which in this case is 0.25, 0.50, and 0.95 (corresponding to the 25th, 50th, and 95th percentiles). Note however that other q -values can be queried without retraining $p_{model}(y|x)$ (and thus other percentiles). Table IV shows the results for each experiment.

For the first three experiments, without a jammer device but with large training sets, the estimated q -values are very close to the selected values. The largest deviation is limited to 0.012, in the case for $q = 0.25$ in experiment 2. Further, for the three jammer experiments, where also the training sets are smaller, the estimated q -values are again close to the selected values. The deviation is again small and limited to 0.025, which can be observed in experiment 5 for $q = 0.50$.

A striking observation is that the q -values, and hence quantiles a , of the distribution can be accurately estimated

using feature set X , despite the challenges in predicting conditional means. This is a strong indication that the conditional distributions are accurately predicted.

D. Predicting Aggregated Distributions

To evaluate the accuracy of predicting aggregated distributions we make use of the Kullback-Leibler divergence, and histogram intersection metrics, both defined below.

The Kullback-Leibler divergence D_{KL} [25] is a metric to determine the difference between two distributions, in this paper p_{agg} and $\hat{p}_{data}$, and is defined as follows

$$D_{KL}(p_{agg}||\hat{p}_{data}) = \sum_{c \in C} p_{agg}(y_c) \ln\left(\frac{p_{agg}(y_c)}{\hat{p}_{data}(y_c)}\right) \quad (9)$$

for each y_c , where $D_{KL} \in [0, \infty]$. The aim is of course to train models with lowest possible divergence for the test set.

One potential issue when calculating D_{KL} is that some bins y_c may have class densities equal to zero, either because the bins are too narrow to capture data points in the training set, or because the classifier returned zero probabilities. This is mitigated by adding a small term ϵ to each class probability, and then normalize, similarly to the method in [8].

A second metric for evaluating aggregated distributions is the histogram intersection D_{HI} , defined as

$$D_{HI} = \frac{\sum_{c \in C} \min(p_{agg}(y_c), \hat{p}_{data}(y_c))}{\sum_{c \in C} \hat{p}_{data}(y_c)} \quad (10)$$

where p_{agg} and $\hat{p}_{data}$ are histograms with the properties described in this paper.

Figure 5 illustrates the predicted aggregated distribution versus the empirical data for each experiment, whereas Table V provides numerical results. All results are based on aggregation on the entire test set, for each experiment, respectively.

For experiments 1, 2, and 3 we can visually see an almost perfect overlap between the predicted aggregated distribution and the empirical distributions. This is also confirmed by the numerical results in Table V, where D_{HI} is 0.969 on average, and D_{KL} is relatively small, from 0.185 in experiment 1 to 0.406 in experiment 2.

For experiments 4 and 5, D_{HI} drops slightly, whereas D_{KL} shows similar values as for the first three experiments. Further, visual inspection of the distribution do provide confidence that the model performance is still high, and definitely useful.

For the most complex scenario in experiment 6, we see a further drop in D_{HI} , while D_{KL} remains in the same range as before. Again, from visual inspection of the graph in Figure 5 we argue that the overall shape of the predicted distribution follows the empirical distribution.

The results are in line with the observations made with respect to prediction of quantiles. The approach is capable in predicting the conditional distribution of RTT, specifically the approach capture sudden fluctuations that are challenging to capture for conditional mean given the current data set.

VI. DISCUSSION

In this paper we provide experimental evidence showing that we accurately can estimate RTT quantiles and aggregated distributions in IoT systems, as observed in Sections V-C and V-D. That is, the large spread in RTT, having its origin in for example fluctuations in the radio environment, is embedded in the model. This is exemplified by Figure 5, where all the distributions have very long tails, extending up to almost 10 seconds (albeit cut at 3 seconds in the figures).

On the other hand, the model performance with respect to conditional mean is lower, even though the predictions follow the general RTT trend, as seen in Figure 4. The main reason for this is that sudden fluctuations, stemming from the devices or radio conditions on the controller-reflector path, are not fully captured in X . That is, for two similar measurements X_{t_1} and X_{t_2} , the corresponding RTT values Y_{t_1} and Y_{t_2} may be different. For conditional mean, this will be manifested as noise and hence a large NMAE, or low R^2 , whereas for quantiles and aggregated distributions the noise is captured well by a flattened conditional probability distributions, from which we estimate quantiles and aggregates.

There are a number of mechanisms to potentially improve prediction of conditional mean. For example, increasing the sampling frequency of features would capture more fine-grained details of the behaviour. Further, it would also be possible to monitor additional features extracted from the

devices and protocols. Unfortunately, both approaches are associated with additional overhead in terms of energy, CPU, and memory usage at the device related to monitoring. Further, transferring additional features puts additional overhead on the already constrained wireless network of the IoT system.

An investigation with respect to feature importance for the specific feature set X shows that the position of the TWAMP controller and reflector devices, distance between them, the reflector rank, as well as LQI at the controller are the most important features. That is, the most important features mainly describe the conditions close to the controller and the reflector, respectively. In theory, one could add, for example, statistics based on rank values r_k of all devices participating in the packet forwarding between the controller and reflector to capture the path behaviour. However, the existing data sets do not provide means to determine the exact path of the TWAMP packets, and thus this is left for future work.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we presented and evaluated an approach for estimation of RTT conditional probability distributions between pairs of devices in an IoT systems. The distributions are represented by a histogram, providing a computationally inexpensive method, that is trained using a supervised random forest classifier. Important statistics including conditional mean and quantiles are then derived from the distribution.

Extensive evaluation of the approach, using data traces from an IoT testbed, showed high model performance across all scenarios. For example, the quantiles of the distribution were predicted with an average error less than 7.6%. The long-term performance of the models, measured by comparing aggregated distributions, were also high, with a Kullback-Leibler divergence below 0.406. The model performance in predicting conditional mean was shown to be consistent with previous work, and the RTT trends were captured. The main challenge regarding conditional-mean prediction points towards sudden fluctuations in the wireless network not fully captured by the features. Increased feature-monitoring frequency may reduce the problem, however this comes with an increased overhead for the already resource-constrained network.

Predicting conditional distributions, and associated derived metrics, is an essential tool for SLA conformance management and service assurance. For example, the approach enables study of probabilistic bounds for RTT, which is a powerful tool for risk assessment associated with SLAs.

Future work includes an in-depth study of the trade-offs between increased monitoring frequency, overhead, and model accuracy. Further, feature engineering and more complex modeling approaches, for example exploiting temporal dependencies, for the distributions will be considered.

ACKNOWLEDGMENT

This research has been sponsored by the Swedish Foundation for Strategic Research (SSF) under grant SM15-0026, and by the Swedish Governmental Agency for Innovation Systems (VINNOVA) through project ITEA3 AutoDC.

REFERENCES

- [1] L. Da 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, 2014.
- [2] J. Postel, "Internet control message protocol," IETF RFC 792, 1981. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc792.txt>
- [3] K. Hedayat, R. Krzanowski, A. Morton, K. Yum, and J. Babiarz, "A two-way active measurement protocol (twamp)," IETF RFC 5357, 2008. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc5357.txt>
- [4] S. Baillargeon, C. Flinta, and A. Johnsson, "Ericsson two-way active measurement protocol (twamp) value-added octets," IETF RFC 6802, 2012. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc6802.txt>
- [5] A. Johnsson and C. Rohner, "On performance observability in iot systems using active measurements," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–5.
- [6] W. Yan, C. Flinta, and A. Johnsson, "Machine-learning based active measurement proxy for iot systems," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, pp. 198–206.
- [7] R. E. Schapire, P. Stone, D. McAllester, M. L. Littman, and J. A. Csirik, "Modeling auction price uncertainty using boosting-based conditional density estimation," in *ICML*, 2002, pp. 546–553.
- [8] E. Frank and R. R. Bouckaert, "Conditional density estimation with class probability estimators," in *Asian Conference on Machine Learning*. Springer, 2009, pp. 65–81.
- [9] "Internet protocol data communication service – ip packet transfer and availability performance parameters," ITU-T Y.1540, 2016.
- [10] L. Metongnon, E. C. Ezin, and R. Sadre, "Efficient probing of heterogeneous iot networks," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2017, pp. 1052–1058.
- [11] M. O. Farooq and T. Kunz, "Proactive bandwidth estimation for iee 802.15.4-based networks," in *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*. IEEE, 2013, pp. 1–5.
- [12] K. Wolosz, U. Bodin, and L. Riliskis, "A measurement study for predicting throughput from lqi and rssi," in *International Workshop on Multiple Access Communications*. Springer, 2012, pp. 89–92.
- [13] R. N. Duche and N. P. Sarwade, "Round trip delay time as a linear function of distance between the sensor nodes in wireless sensor network," *International Journal of Engineering Sciences & Emerging Technologies*, vol. 1, pp. 2231–6604, 2012.
- [14] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, and L. T. Yang, "Data mining for internet of things: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 77–97, 2013.
- [15] M. Geilke, E. Frank, A. Karwath, and S. Kramer, "Online estimation of discrete densities," in *2013 IEEE 13th International Conference on Data Mining*. IEEE, 2013, pp. 191–200.
- [16] M. Geilke, A. Karwath, E. Frank, and S. Kramer, "Online estimation of discrete, continuous, and conditional joint densities using classifier chains," *Data Mining and Knowledge Discovery*, vol. 32, no. 3, pp. 561–603, 2018.
- [17] T. Pospisil and A. B. Lee, "Rfcde: Random forests for conditional density estimation," arXiv preprint arXiv:1804.05753, 2018.
- [18] F. S. Samani and R. Stadler, "Predicting distributions of service metrics using neural networks," in *2018 14th International Conference on Network and Service Management (CNSM)*. IEEE, 2018, pp. 45–53.
- [19] F. Shahab, R. Stadler, A. Johnsson, and C. Flinta, "Predicting distributions of service metrics," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2019, pp. 745–746.
- [20] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.
- [21] M. Schuß, C. A. Boano, M. Weber, and K. Roemer, "A competition to push the dependability of low-power wireless protocols to the edge," in *EWSN*, 2017, pp. 54–65.
- [22] N. Kushalnagar and G. Montenegro, "Transmission of ipv6 packets over iee 802.15.4 networks," IETF RFC 4944, 2007. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc4944.txt>
- [23] A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "Rpl: Ipv6 routing protocol for low-power and lossy networks," IETF RFC 6550, 2012. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc6550.txt>
- [24] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. Zúñiga, "Jamlab: Augmenting sensor network testbeds with realistic and controlled interference generation," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IEEE, 2011, pp. 175–186.
- [25] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.