Sampling and Classifying Interference Patterns in a Wireless Sensor Network

NICHOLAS M. BOERS and IOANIS NIKOLAIDIS
University of Alberta
and
PAWEL GBURZYNSKI
Olsonet Communications Corporation

The low-powered transmissions in a wireless sensor network (WSN) are highly susceptible to interference from external sources. Meanwhile, the industrial, scientific, and medical (ISM) radio bands in which they operate are cluttered with radio frequency-emitting devices. Our work is a step towards enabling WSN devices to better understand the interference in their environment so that they can adapt to it and communicate more efficiently.

In this paper, we present our results from measuring noise and interference with a collection of mote-class synchronized receivers at sample rates that are, to the best of our knowledge, higher than previously described in the literature. These traces contain distinct interference patterns, each with a different potential for being exploited by cognitive radio strategies. Given the energy and space constraints of a WSN node, we explore succinct decision tree classifiers for the distinct patterns. We expand on a basic feature set to incorporate attributes based on the dip statistic and the Lomb periodogram, both of which address specific empirically observed behaviour. We present an approximation of the periodogram that makes its construction feasible for mote-class devices.

Categories and Subject Descriptors: C.2.1 [**Network Architecture and Design**]: Wireless communication; I.5.5 [**Pattern Recognition**]: Implementation—*Special architectures*

General Terms: Experimentation, Algorithms

Additional Key Words and Phrases: Classification, interference, sampling, wireless sensor networks

1. INTRODUCTION

A myriad of factors can affect wireless communication in dense and dynamic *urban* environments. Researchers often assume, whether working with simulators or physical devices, that wireless sensor network (WSN) nodes receive (a) a signal, (b) interference from peer devices, and (c) additive white Gaussian noise (AWGN). Rarely do they give any serious attention to external noise sources and interferers. In reality, the industrial, scientific, and medical (ISM) radio bands are heavily oc-

Authors' addresses: N. M. Boers and I. Nikolaidis, 221 Athabasca Hall, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, T6G 2E8; email: {nboers, nikolaidis}@ualberta.ca; P. Gburzynski, Olsonet Communications Corporation, 51 Wycliffe Street, Ottawa, Ontario, Canada, K2G 5L9; email: pawel@olsonet.com.

cupied with devices [Do et al. 2004], and the overly optimistic assumption of quiet channels rarely holds. External interference is indeed quite common, and it can seriously impact network performance (e.g., [Musaloiu-E. and Terzis 2008]).

A better understanding of interference is particularly relevant to cognitive networking. By determining the near-term occupancy of a channel through noise and interference observations, cognitive radios aim to opportunistically use unoccupied frequency bands [Yucek and Arslan 2009]. In our work, we attempt to characterize channels (at a high-level) that are potentially occupied, a step needed by cognitive radios. In fact, rather than just producing a binary occupied/unoccupied classification, our approach produces a class. Some of the classes, for example those exhibiting signs of spread spectrum or ultra-wideband (UWB) interferers, could, depending on the cognitive networking scheme, be used at the same time for narrowband transmissions by the cognitive network. However, in this study, we do not investigate *how* the channels will be used, but rather what is a good (and as exhaustive as possible) characterization of channels based on their noise and interference time series behaviour.

Consistent with work in cognitive networking, as well as an ulterior objective of our current work, is the ability to use as small a number of samples, together with the right features and classification algorithm (expressed as a decision tree), to allow nodes to reassess the channel behaviour and re-classify channels on an as-need basis. An awareness of energy consumption motivates us to do this collection and classification efficiently, since batteries often power nodes. Consistent with these objectives, we use the same hardware (i.e., RF transceiver and microcontroller) for the measurement and classification as we use in our deployed networks. We do not assume the use of any special equipment or special calibration protocols. Please note that all measurements reported in this study are noise measurements, i.e., without our sensor nodes introducing any transmissions of their own.

In earlier work, we investigated the noise and interference on 256 channels from 904 MHz to 928 MHz (within the ISM band) and 929 MHz to 954 MHz (outside the ISM band) in an urban environment [Boers et al. 2010]. Our use of the term *urban environment* is analogous to the non-deterministic environments described in [Oehen and Plattner 2006]. For each channel, 16 nodes placed in a four-by-four grid with 1.83 m spacing recorded their received signal strength indicator (RSSI) value at roughly 5000 Hz to produce 4096 traces. The RSSI value is a measure of the radio-frequency power input at the transceiver [Texas Instruments 2007]. By sampling each channel for around 35 s, we collected 716 800 000 RSSI values in a period of about 2.5 h.

We initially explored these traces by plotting them against time. We observed five primary classes of channels (Fig. 1): (a) quiet, (b) spikes, (c) rapid-spikes, (d) high-and-level, and (e) shifting-mean. In many cases, a channel's trace would show a mixture of these characteristics (e.g., rapid spikes over a weak shifting-mean pattern). When this occurred, our previous work classified the channel by the dominant pattern. In this work, however, we assign a score for each characteristic present in a trace.

In introducing these channel classes, we are cognisant that we measured a specific environment, on a particular day, over a limited time period. If we were to measure

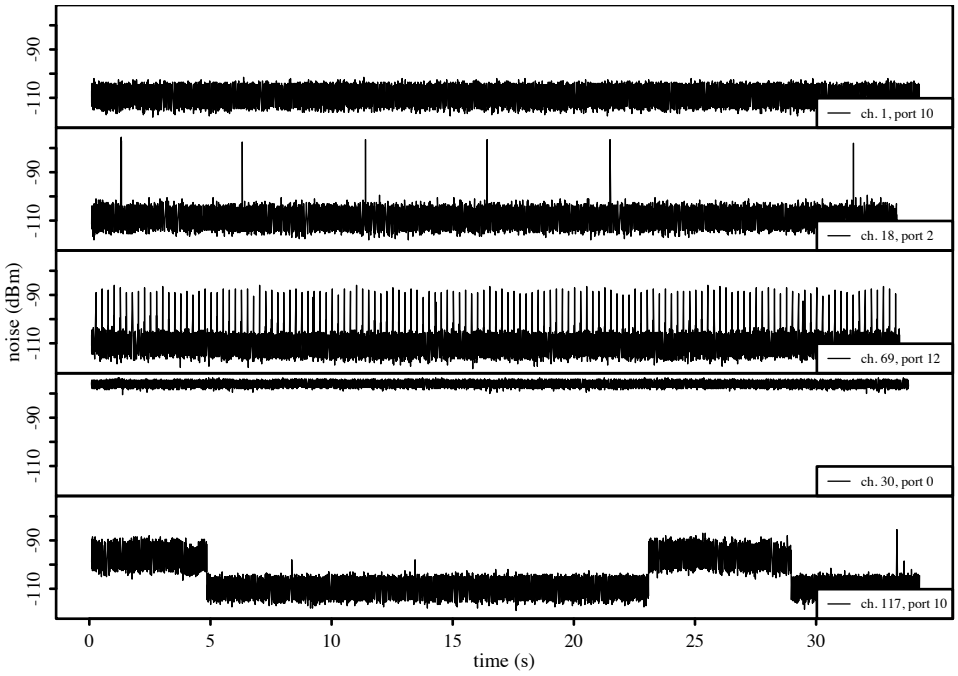


Fig. 1: The different primary classes of channels that we identified in our RSSI traces. From top to bottom, the figure shows an example of a (a) quiet, (b) spikes, (c) rapid-spikes, (d) high-and-level, and (e) shifting-mean channel.

another environment – or even the same one again – we would certainly see some different patterns. Nonetheless, these patterns provide a starting point for our research and allow us to begin addressing some interesting questions.

Again from our previous work, Table I shows the distribution of observed traces among these five different types. In the ISM band, we encountered a number of quiet traces ($n = 155$), while the vast majority ($n = 1781$) contained some form of interference. While much of this interference would have a very low impact on receptions (spikes, $n = 1146$), the remaining three types would be much more detrimental. The high-and-level and shifting-mean channel types are best avoided. In the former, the interference is constant,[1] and in the latter, it is highly unpredictable. For other channel types, such as the rapid-spikes channels, nodes could employ avoidance strategies given the periodic and consistent interference pattern.

In the non-ISM band, we saw a much greater occurrence of quiet channels ($n = 1614$). Our second-highest class, shifting-mean ($n = 273$), was often very strong with a consistent level across all deployed nodes. To investigate these traces, we searched the Assignment and Licensing System (ALS) database at Industry Canada. We found that the pattern often occurred on frequencies used by the paging services

[1]The high-and-level interference class occurs on channels with frequencies falling at $806 + 13(n)$ MHz. 13 MHz is $\frac{1}{2}$ of the transceiver's crystal frequency, and thus, originates at the nodes themselves. Our observation agrees perfectly with the anomaly described in [Chipcon 2005].

Table I: The number of traces classified as each type for both ISM and non-ISM bands.

Classification	ISM	Non-ISM	Total
quiet	155	1614	1769
spikes	1146	179	1325
rapid-spikes	523	25	548
high-and-level	32	69	101
shifting-mean	80	273	353

of various wireless providers in Edmonton, Alberta.

In this paper, we explore the classification of noise and interference encountered in our urban wireless sensor network. After reviewing related work (Section 2), we describe the hardware setup for our experiments (Section 3). In Section 4, we describe the tree classifier that we adopted. We selected the decision tree approach given its low classification overhead with the expectation that nodes will eventually perform this type of classification themselves. We investigate how nodes should sample a channel by looking at different techniques: even and uneven/Poisson sampling. Having collected samples, we then explore beyond the obvious classification features and look specifically at the usefulness of the dip statistic and Lomb periodogram for classifying shifting-mean and rapid-spikes channels, respectively. In Section 5, we explore an implementation of the periodogram suitable for mote-class devices. Finally, in Section 6, we conclude with a summary of our work.

2. RELATED WORK

Interference can originate from any number of sources in an environment. For example, Chandra [Chandra 2002] used a spectrum analyzer in a 3-story building to explore frequencies in the 900 MHz and 1.8 GHz bands. He looked at the noise generated by electronic equipment in a workshop, a photocopier, elevator, and fluorescent tubes, and he concluded that all should be modelled as interferers in the building. He did not, however, consider the detection of these interferers using WSN hardware.

Using sensor platforms, Srinivasan, Dutta, Tavakoli, and Levis [Srinivasan et al. 2006] studied packet delivery performance. They encountered large spikes (up to -35 dBm or higher) in their traces and investigated these further. With the nodes synchronized, they checked for spatial correlation among the spikes, and upon finding high correlation, concluded that the spikes originated externally to the nodes. We also noticed correlation in many of our traces, which led us to the same (external) conclusion.

Researchers working on closest-fit pattern matching (CPM) sampled noise in both indoor and outdoor environments [Lee et al. 2007; Rusak and Levis 2010]. While we sampled at 5 kHz, they sampled at relatively low rates (1 kHz or less) and made only the informal comments about interference patterns that we reiterate in the following paragraphs.

Using the CC2420 IEEE 802.15.4 transceiver, Lee, Cerpa, and Levis [Lee et al. 2007] sampled the chip's RSSI register at 1 kHz (the register itself updates at 62.5 kHz). By storing the retrieved measurements in the device's flash memory, they could record samples for 197 s. They sampled noise on channels that both

overlapped and did not overlap IEEE 802.11b channels in Wi-Fi-enabled buildings, Wi-Fi-enabled outdoor areas, outdoor quiet areas, and controlled areas. They observed three key characteristics in their samples: (a) spikes sometimes as strong as 40 dB above the noise floor, (b) many of the spikes were periodic, and (c) over time, the noise patterns changed. They did not encounter the shifting-mean characteristic that we observed.

Using TelosB motes with CC2420 transceivers, Rusak and Levis [Rusak and Levis 2010] sampled the chip's RSSI register for packets transmitted between a pair of nodes at both 4 and 100 Hz. They sampled channels in buildings at Cornell and Stanford University and concentrated on modelling the signal strength rather than just noise. Both environments had a number of interference sources including 802.11 wireless networks, cordless phones, microwaves, and personal wireless access points. Although they mentioned the presence of interferers, they did not comment on the observed interference patterns.

Most recently, Srinivasan, Dutta, Tavakoli, and Levis [Srinivasan et al. 2010] expanded on much of their previous work. With six synchronized nodes, they sampled RSSI values at 128 Hz and explored the correlation of noise traces. They observed 802.11b interference at 45 dB above the noise floor and suggested avoiding channels that coexist with 802.11b networks.

3. HARDWARE SETUP

Using the Texas Instruments CC1100 transceiver [Texas Instruments 2009] in EMSPCC11 wireless nodes [Olsonet Communications Corp. 2008], we set up a four-by-four grid of nodes with 1.83 m spacing and elevated each 28 cm off of the floor within the Smart Condo [Boers et al. 2009] at the University of Alberta. The Smart Condo is within Telus Centre, a moderately sized office building, located across the street from a large apartment building. Within these two buildings, any number of interference sources could exist.

These devices run the PicOS operating system [Akhmetshina et al. 2003], which supports multithreaded applications within only 2 kB of RAM. A simple application runs on each node that reads the transceiver's RSSI register and writes the resulting 8-bit 2's complement value to the UART at over 5 kHz. We verified the periodicity of the measurements by toggling an LED on the board and monitoring its signal with an oscilloscope. Using FTDI TTL-232R-3V3 cables, USB extension cables, and powered 7-port USB hubs (Digitus DA-70227), we connected all of the nodes to a single computer for data collection. In making the connections, we never exceeded USB's maximum cable length of 5 m.

Before collecting RSSI traces, we verified that the connection of our motes by wires to a single data collection point did not affect their RF communication. Note that there was no reason to suspect influence, because of the careful isolation of the RF tract on the EMSPCC11, which is a common practise in professionally designed RF equipment. By disconnecting individual nodes, logging measurements to on-board memory, and (visually) inspecting RSSI traces, we were able to ascertain that neither the USB cables alone, nor their connections to the central hub, had a perceptible impact on the RF channels.

On the computer, an application opens all 16 serial ports. With samples arriving

at 80 kHz, our application cannot capture each individual sample immediately as it arrives. Instead, we make use of the TTL-232R-3V3's 256-byte receive buffer, and our application reads values in blocks rather than singly. Immediately after reading a block, the application timestamps each value (interpolating as necessary) and stores it in a CSV text file. Earlier work, e.g., [Musaloiu-E. and Terzis 2008], raises concerns about buffer overruns while sampling; we addressed these concerns with this buffer and the steps described in the next paragraph.

The whole application is very sensitive to latency, so we took a number of precautions to prevent overruns. These precautions included (a) introducing large circular per-port buffers (multiple 64 kB blocks per port), (b) reading the ports and writing measurements to those circular buffers (in a primary thread), (c) assigning this thread real-time priority (in Mac OS X 10.6, 2.5 ms of computation time per 5.0 ms), and (d) in a secondary thread, writing completed buffer blocks to disk. With these precautions, our application processed the samples with ease on a late-2006 model MacBook Pro.

With the described hardware and software in place, we proceeded to measure the RSSI on each of the node's 256 channels. Configured with a base frequency of 904 MHz, channel spacing of 199.9512 kHz, and bandwidth of 101.5625 kHz, our nodes listened on frequencies from 904 MHz to 954 MHz (i.e., both inside and outside the ISM band). For each channel/node combination, we collected exactly 175 000 samples. The whole collection process took roughly 2.5 h and the final CSV data file consumed 18.38 GB.

With the traces stored in a CSV file, we converted the 8-bit 2's complement RSSI values into signed dBm values (with $\frac{1}{2}$ dB resolution) using the procedure outlined in the CC1100 documentation [Texas Instruments 2009]. To plot the received signal strength (dBm) against time (s) for each of the 4096 traces, we thinned the data first using an implementation of the Douglas-Peucker line simplification algorithm [Douglas and Peucker 1973]. The thinned results, which had on average 2077 points per trace, maintained the visual characteristics that we needed to hand-classify traces. At the same time, plotting the thinned version was much more efficient than plotting the full 175 000-point traces.

Recall that in our initial classification of traces (Section 1), we encountered traces containing more than one characteristic. To avoid classifying them by the dominant one in this work, we instead consider each characteristic independently. From the 4096 original traces, we randomly selected 1024 traces to closely inspect in a random order. For each trace, we evaluated it for the following five characteristics:

(1) spikes: for apparently random and infrequent spikes, their approximate strength (dB) above the noise floor and number,
(2) periodic-spikes: for periodic and infrequent spikes, their approximate strength and number,
(3) rapid-spikes: for periodic and frequent spikes, their approximate strength,
(4) shifting-mean: the strength of the shift, and
(5) high-and-level: the strength of the level above where we would expect to find the noise floor.

Most of the frequent spikes occurred at 4 Hz, while most of the infrequent ones

occurred at 0.2 Hz or less, and we never encountered any difficult-to-classify cases that required us to measure the time between spikes.

Beyond the visual differences of the infrequent and frequent spikes, other reasons made it quite natural to distinguish the two types. In our observed traces, the frequent spikes occur approximately 20 times more often than the infrequent spikes. Therefore, recognizing the frequent case should require a magnitude fewer samples. Similarly, once the pattern is recognized, the frequent case lends itself much better to opportunistically timing transmissions because it will require fewer samples to recognize a spike, which will save energy and result in lower latency. Finally, the two different patterns have different implications on packet error rates. Given Smart Condo-sized packets and bit rates, a transmission can last up to 50 ms. If the interference strength exceeds the signal strength, such transmissions can quite effectively exist with the spikes pattern, which could yield a 1% packet loss rate. With the rapid-spikes pattern, however, the higher frequency now suggests a baseline loss rate of 20% – a level significant enough to warrant some way of handling the rapid spikes to avoid this drastic performance degradation.

From these five numeric values, we introduce three new Boolean class features: *spikes* (for any of the three spike cases), *rapid-spikes*, and *shifting-mean*. We set these new attributes to true when a characteristic's level exceeds 3 dB, and we later use these three Boolean values separately to train classifiers. Herein, any reference to their names is to the Boolean rather than numeric attribute.

4. CLASSIFICATION

Many different classification techniques exist, and we focused on those most feasible for low-powered nodes. We elected to use a decision tree classifier, since once built offline, it has low memory and processing requirements. Quite possibly the most well-known algorithm in the literature for building decision trees [Kotsiantis 2007] is the C4.5 classifier [Quinlan 1993], and we adopted it for our experiments. C4.5 actually refers to the collection of programs described in [Quinlan 1993]; the primary program, sharing the same name, generates a classifier in the form of a decision tree.

Weka 3.6.2 [Hall et al. 2009], data mining software written in Java, incorporates the implementation of the C4.5 classifier that we used. Algorithm 1 overviews its recursive structure, which essentially builds the tree top-down and attempts to select the best attribute for each decision node as it builds. It begins the process with all of the feature vectors in the training set, and as it goes, the tree's decisions gradually partition those vectors into smaller subsets.

The algorithm initially tests a number of base cases and ends the recursion if (a) all of the feature vectors belong to the same class or (b) not enough feature vectors exist to warrant a decision node. Both of these cases result in a leaf node with its class determined by the class majority among its feature vectors.

When the base cases fail, it will create a new decision node that contains a test, and it will explore each non-class attribute for that test. For a continuous numeric attribute, the decision node will test whether the attribute value is above a certain threshold. Such a test results in a Boolean outcome: true or false. For a discrete (nominal) attribute, each possible value of the attribute will result in an outcome.

Algorithm 1 Building the C4.5 decision tree

```
1: check base cases
2: initialize list U for storing useful splits
3: initialize list G for storing their information gains
4: for each a in the non-class attributes do
5:    if split(a) is useful then
6:       U.append (a)
7:       G.append (gain (a))
8:    end if
9: end for
10: let a_best be one attribute a from U such that
          gain(a_best) ≥ mean(G) and
          gratio(a_best) = max_{a∈U} gratio(a)
11: create decision node that splits on a_best
12: recurse on each subset below decision node
```

Each possible outcome of the decision yields a child and the training set's feature vector is partitioned (split) amongst the children according to the result of the test. It is possible for a child to contain no feature vectors, i.e., when a possible nominal value does not occur in the subset of feature vectors. In these cases, the child becomes a leaf with its class determined by the most frequent class of its parent. A good attribute for a decision node yields two or even a few children, where each child is reasonably pure, i.e., the feature vectors in the child are predominately of the same class. In fact, before an attribute becomes a candidate for the best split, an initial usefulness test limits the candidates to (a) non-numeric attributes with a limited number of outcomes and (b) numeric attributes. After building the tree, a reduction phase (not described here, but found in [Quinlan 1993]) collapses and prunes the tree.

To determine the best attribute for a decision node, the concept of *information* plays a fundamental role [Quinlan 1993, p. 21]. As usual, the formula $-\log_2 P$ conveys the information content of a message in bits, where P is the message's probability. For example, given four equally probably messages ($P = \frac{1}{4}$), the information content of any one of them is $-\log_2(\frac{1}{4})$ or 2 bits.

Algorithm 1's first use of *information* occurs on line 7, when it calculates the *gain* of splitting the feature vectors F on the attribute a. Gain represents "the informational value of creating a branch on the ... attribute [a]" [Witten and Frank 2005]. Prior to calculating this difference, we first need a baseline, which in this case is the informational value *prior* to creating the branch.

Let F be the feature vectors available when creating a decision node. Given a case in F, we can calculate the average amount of information needed to identify its class

$$\text{info}(F) = -\sum_{i=1}^{n} \left[\frac{\text{freq}(C_i, F)}{|F|} \times \log_2 \left(\frac{\text{freq}(C_i, F)}{|F|} \right) \right] \tag{1}$$

where $C_1, C_2, \ldots, C_n$ represent the different classes and the notation $\text{freq}(C_i, F)$ refers to the frequency of class C_i in F.

After splitting the feature vector into the children using a test X with m outcomes, the average amount becomes

$$\text{info}_X(F) = \sum_{i=1}^{m} \left[\frac{|F_i|}{|F|} \times \text{info}(F_i) \right] \tag{2}$$

where F_i is the collection of feature vectors partitioned into child i.

With these values, we can finally calculate the gain achieved by splitting on the test X

$$\text{gain}(X) = \text{info}(F) - \text{info}_X(F) \tag{3}$$

In a predecessor to C4.5 named ID3, Quinlan used only this gain as its criterion for determining the best split. Unfortunately, notice from 3 that the gain is maximized when $\text{info}_X(F)$ is minimal; if each outcome contains only a single feature vector then $\text{info}_X(F)$ will equal 0. To compensate for this problem, C4.5 uses a normalization of gain called the *gain ratio* when determining the best split. The gain ratio divides gain by the potential information generated by partitioning F into m subsets (one for each outcome)

$$\text{gratio}(X) = \frac{\text{gain}(X)}{-\sum_{i=1}^{m} \frac{|F_i|}{|F|} \times \log_2 \left(\frac{|F_i|}{|F|} \right)} \tag{4}$$

In order to use this classifier, we obtain a number of features from our traces. The most obvious metrics include the (a) mean, (b) standard deviation, (c) moving-average standard deviation, (d) skew, (e) kurtosis, (f) minimum, and (g) maximum. The moving-average standard deviation rather arbitrarily averages ten data points. We also include a number of percentiles: the 95.0, 96.0, 97.0, 98.0, 99.0, and 99.5 percentiles.

With these features alone, we fail to capture at least two key characteristics in our data. For the shifting-mean traces, we observed that their density plots are rarely unimodal (Fig. 2). For the rapid-spike traces, we observed that their high-frequency spikes tend to be very periodic. In Sections 4.2 and 4.3, we introduce new features to account for these observations.

Fig. 3 shows one of the trees produced by the C4.5 algorithm. In this tree, which classifies rapid-spikes channels, the first feature that it uses is *periodic*, an attribute that results from our frequency analysis (Section 4.3). The tree only uses a subset of all the features available to it. The remaining features either failed to meet the *useful* criteria tested in Algorithm 1 or were pruned in the post-processing.

4.1 Subsampling

We built our training set from the 175 000 samples per trace that we collected, but we cannot expect deployed nodes to collect the same number of samples. These devices tend to have small memories, and as they collect more data, the computations also consume additional CPU cycles. Moreover, the data collection requires the transceiver to actively listen to the channel, which increases the node's energy consumption. For those reasons, we investigate classification performance on subsamples of our data. Since we passively collected traces, subsampling our data is equivalent to collecting the initial samples at a lower rate. If we had used an active

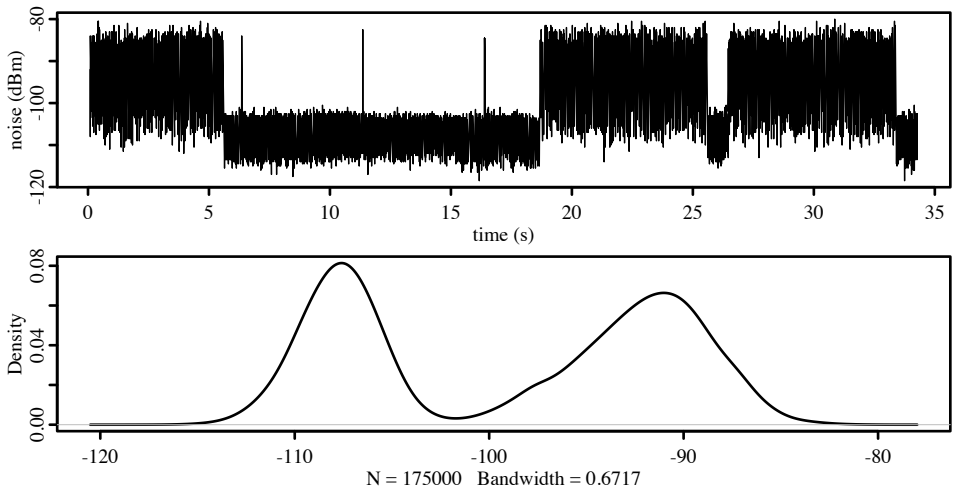


Fig. 2: The trace from channel 94, port 10 (top) and its density plot (bottom). This trace has the shifting-mean pattern and its density plot is bimodal.

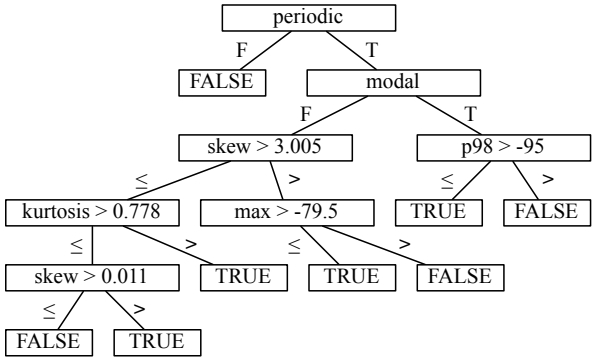


Fig. 3: Tree produced by the C4.5 algorithm to classify the rapid-spikes channels.

approach that could affect the environment, e.g., packet transmissions, then this approach would be invalid.

With periodic or even sampling, a constant time separates each sample ($\triangle t = t_{i+1} - t_i$ is constant $\forall i$). In this case, it is possible for the sampling to be synchronized with the trace's characteristic of interest – and all of the samples could miss it. For that reason, we compare periodic sampling with Poisson sampling, since the latter separates samples by random times and also maintains the distribution characteristics of the sampled distribution [Cramér 1962, pp. 327-328].

Suppose that we want to produce an even subsample of length n from a trace of length N. We divide the trace into n segments of equal duration

$$\left[t_{\frac{0}{n}N}, t_{\frac{1}{n}N}\right), \left[t_{\frac{1}{n}N}, t_{\frac{2}{n}N}\right), \ldots, \left[t_{\frac{n-1}{n}N}, t_{\frac{n}{n}N}\right]$$

Within the first segment, between t_0 and $t_{\frac{1}{n}N}$, we randomly select a starting point t_s and its corresponding value is the first in our subsample (x_1). For the times $t_{\frac{1}{n}N} + t_s$, $t_{\frac{2}{n}N} + t_s$, ..., $t_{\frac{n-1}{n}N} + t_s$, we then add their corresponding values to our subsample, $x_2, x_3, \ldots x_n$, respectively.

To produce a Poisson subsample, we first determine the length of the subsample by drawing a random number from the Poisson distribution with $\lambda = n$. We then draw a simple random sample without replacement of that length from the trace.

To evaluate our results, we use the ϕ coefficient, which is actually a product-moment coefficient of correlation [Kotz et al. 2006] and is also called the Matthews correlation coefficient [Matthews 1975]. It indicates the association between two variables, and as a normalized value, ranges from -1 to 1. It reflects the relationship between the correct classifications, true positives (TP) and true negatives (TN), and the incorrect classifications, false positives (FP) and false negatives (FN). We generally want the value to be close to 1, which indicates that the true negative/positive values are (much) greater than the false negative/positive values. Although other methods can be used to evaluate classifiers, e.g., the sensitivity, specificity, precision, and recall percentages, the coefficient may provide a more balanced evaluation [Baldi et al. 2000]. Unlike percentage-based measures, it uses (and can be calculated solely from) the four values from a confusion matrix

		Predicted	
		FALSE	TRUE
Actual	FALSE	true negative (TN)	false positive (FP)
	TRUE	false negative (FN)	true positive (TP)

using the formula

$$\phi = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{5}$$

It has a direct relationship with the χ^2 statistic for a 2×2 contingency table

$$\phi^2 = \frac{\chi^2}{n} \tag{6}$$

and this relationship can be used to test significance levels (with 1 degree of freedom).

In our comparisons of even versus Poisson sampling, we evaluated classifiers for three types of channels: spikes, rapid-spikes, shifting-mean. We looked at the effect of Poisson sampling on the ϕ coefficient (Fig. 4) and used even sampling as the baseline. We used ten subsample sizes of 250, 375, 500, 750, 1000, 2000, 3000, 4000, 6000, 8000, and 12 000 points for each subsampling technique for each of the three types of channels. In all these cases, we used all available attributes, including the new *periodic* and *modal* Boolean attributes that we define in Sections 4.2 and 4.3.

All but one of the resulting ϕ values had statistically significant correlation with $P = 0.001$, meaning that the predicated classification was significantly correlated

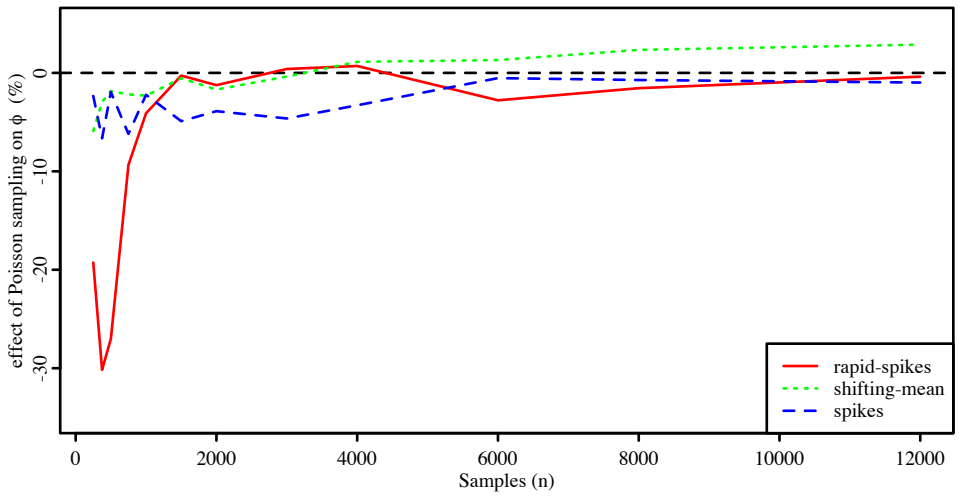


Fig. 4: The effect of using Poisson rather than even sampling on ϕ for identifying three classes of channels.

with the actual classification at this significance level. The exception, 250 samples with the rapid-spikes classifier and Poisson sampling, was still statistically significant at $P = 0.01$. For large subsample sizes ($n \geq 750$), we observed little difference between the two sampling techniques. With fewer subsamples ($n < 750$), we observed some deviation in the rapid-spikes classifier, which we discuss next.

In Section 4.3, we describe the development of periodogram response thresholds, which are suitable for an average sample. A spike's approximate duration is 5 ms, and by sampling at around 5000 Hz, a full trace may capture 25 samples per spike. It follows that with even sampling, a rate of only $5000 \times \frac{1}{25} = 200$ Hz would still capture every spike, and even lower rates would still tend to detect a *fixed fraction* of all spikes ($\frac{5 \text{ ms}}{\text{inter-sample time}}$). An individual Poisson sample, on the other hand, may capture a higher or lower fraction of the spikes, while on average capturing the same fraction. When it detects a higher fraction, a particularly high periodogram response may result, but as long as that response exceeds the threshold, any excess is in a sense wasted. On the other hand, a lower fraction resulting in a lower response will fail to meet the threshold. At higher sample sizes, both techniques capture a sufficient number of spikes to mitigate these effects. For this reason, we are not particularly surprised that the rapid-spikes classifier with Poisson sampling performed worse at small sample sizes.

4.2 Dip statistic

Given that histograms of shifting-mean traces often show more than one mode, our search for a statistic to capture this fact led us to the dip statistic [Hartigan and Hartigan 1985]. Without making any assumptions about the underlying sample distribution, the dip statistic allows us to test for unimodality. It does this by computing "the maximum difference between the empirical distribution function and the unimodal distribution function that minimizes that maximum difference" [Har-

tigan and Hartigan 1985].

For a given p-value, the dip value depends heavily on the number of samples. Hartigan and Hartigan [Hartigan and Hartigan 1985] provide values for sample sizes up 200 and Martin Maechler[2] extends that table up to sample sizes of 5000. Given that our complete traces consist of 175 000 points, we again extended the table.

To make this extension, we used code provided by Martin Maechler, which was originally based on the published Hartigan code [Hartigan 1985]. Our random uniform values of the null distribution came from the GNU Scientific Library [Galassi et al. 2009] with the `gsl_rng_ranlxd2` generator and unique seed values for each run. We calculated values for both 50 000 and 175 000 samples at 26 different significance levels.

We next augmented our training set with dip values. We also introduced a new Boolean feature named *modal* that we set to true when we rejected the null (unimodal) hypothesis, i.e., when the dip statistic exceeded the threshold for a given significance level. To determine the best significance level, we calculated the *modal* attribute for each level, and in each case, looked at the agreement between the shifting-mean and modal attributes. The previously introduced ϕ coefficient gave us an indication of the agreement, and we obtained the maximum value for it at the significance level of 1.

For each of our 1024 hand-classified traces, we computed subsamples using both even and Poisson sampling. We again used subsample sizes of 250, 375, 500, 750, 1000, 2000, 3000, 4000, 6000, 8000, and 12 000 points. In all cases, for each subsample size and technique, we subsampled each trace 10 times. When adding our Boolean *modal* attribute to the subsamples, we interpolated for the smaller sample sizes using values from our dip value table at the chosen significance level.

After classifying the subsamples, we see that the modal attribute provides considerable benefit across all the tested subsample sizes (Fig. 5). The shifting-mean channels tend to spend a reasonable amount of time at each level, and thus, even a small number of samples can capture the two modes. Without the modal attribute, the classifier was very sensitive to the sample size, and generally performed very poorly. We did not see a significant difference between even and Poisson sampling.

4.3 Lomb periodogram

In our plots of traces with rapid spikes, we noticed that the spikes appeared to have a very periodic nature. The first thought that occurred to us – Fourier transforms – does not apply when the spacing between samples is uneven. Given our desire to test Poisson sampling with its uneven spacing, we looked for alternative solutions.

The Lomb periodogram is a technique of computing least-squares frequency analysis on unequally spaced data [Lomb 1976; Press et al. 1992]. Given N data points,

[2]See http://www.cran.r-project.org/web/packages/diptest/.

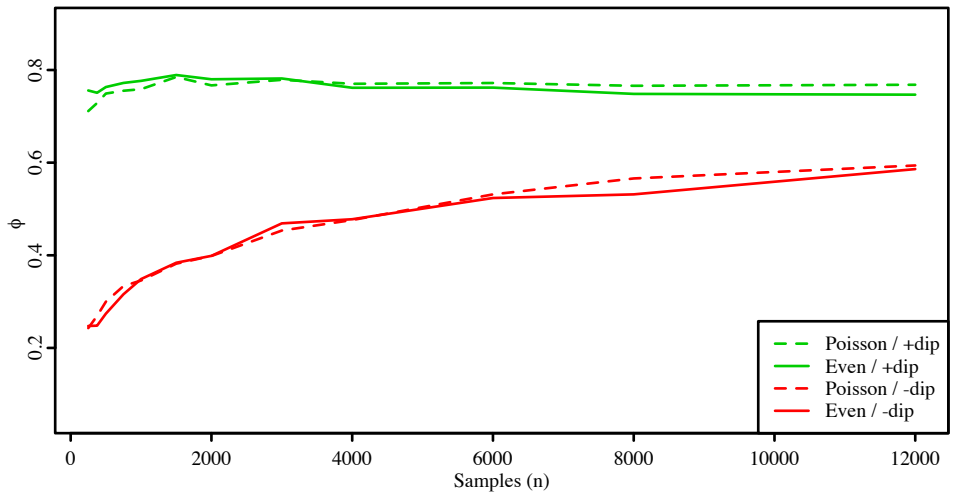


Fig. 5: The effect of sampling technique and use of the modal attribute (derived from the dip statistic) on the classification of *shifting-mean* subsamples.

the periodogram's normalized form is defined by

$$P_N(\omega) \equiv \frac{1}{2\sigma^2}\left\{\frac{\left[\sum_j (h_j - \bar{h})\cos\omega(t_j-\tau)\right]^2}{\sum_j \cos^2\omega(t_j-\tau)} + \frac{\left[\sum_j (h_j - \bar{h})\sin\omega(t_j-\tau)\right]^2}{\sum_j \sin^2\omega(t_j-\tau)}\right\} \tag{7}$$

where $\omega = 2\pi f$, f is the frequency, h_j and t_j represent the magnitude and time of sample j (respectively), $\bar{h}$ and σ^2 are the mean and variance of all the samples (respectively), and τ is defined by the relation

$$\tan(2\omega\tau) = \frac{\sum_j \sin 2\omega t_j}{\sum_j \cos 2\omega t_j} \tag{8}$$

The intuition behind (7) is as follows: for each sample in the trace, multiply it by the sine and cosine waves modulated at the chosen frequency f. One of the two waves will capture any periodicity at f regardless of shifts in time. When the values $h_j - \bar{h}$ are random distributed about 0, i.e., the mean-adjusted signal level is essentially a Gaussian random variable with mean 0, this sum will tend to 0. When the values also modulate with the frequency, however, it will tend to a larger value. The final division at the end, by the sample variance, accounts for the width of the Gaussian distribution.

In our periodograms, we noticed that in addition to a peak at the fundamental frequency, additional peaks often occurred at multiples of that frequency. According to [Newton 2007], the strong non-sinusoidal signal in our time series is responsible for these peaks, and the subsequent peaks are called harmonics.

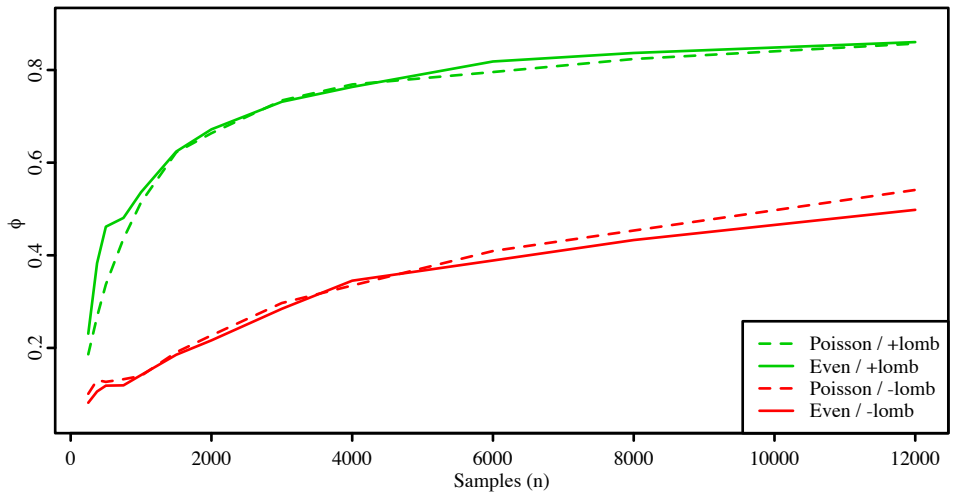


Fig. 6: The effect of sampling technique and use of the periodic attribute (derived from the Lomb periodogram) on the classification of *rapid-spikes* subsamples.

We added a *periodic* attribute to our training set to summarize the periodogram in a single Boolean value. To determine when it should be true, we considered for each trace (a) the maximum periodogram value in one of the frequency ranges 2.5 to 3.5 Hz, 2.5 to 4.5 Hz, 2.5 to 5.5 Hz, and so on up 2.5 to 20.5 Hz and (b) the threshold that, when the above maximum periodogram value exceeds it, yields a maximum value for ϕ. Using a variant of simulated annealing implemented in R's `optim` function,[3] we identified a suitable threshold parameter for setting the periodic attribute when using the maximum frequency between 2.5 and 5.5 Hz.

Like with the dip statistic, we subsampled our 1024 hand-classified traces, using both even and Poisson sampling, for the same subsample sizes and repetitions. Given that the periodogram is a sum across the (sub)sample, its magnitude depends on the number of samples in the sum. Therefore, when adding our new Boolean attribute to the subsamples, we had to scale the threshold proportionately.

We found that the periodic attribute derived from the Lomb periodogram produced significant gains in the rapid-spikes classifier performance (Fig. 6). Given small sample sizes, classifiers both with and without the periodic attribute show low performance. As the sample size increases, the performance of the classifier incorporating the *periodic* attribute increases at a higher rate than the one without the attribute. Unlike the dip statistic, a reasonable sample size appears to be necessary for a response from the periodogram. Like with the dip statistic, we did not see a significant difference between even and Poisson sampling.

We compared the decision trees both with (Fig. 3) and without (Fig. 7) the *periodic* attribute. Prior to adding the attribute, the tree size is 67 with 34 leaf nodes, while with the attribute, the size is 15 with only 8 leaf nodes. It is evident that given the *periodic* attribute, we not only improve the classification performance

[3]See `http://www.r-project.org` for more information.

but also obtain a simpler decision tree.

Given the simplification introduced by the periodic attribute, we went back to compare the shifting-mean case both before and after adding its *modal* attribute. The tree without this attribute has 29 nodes, 15 of which are leaf nodes, and adding the modal attribute reduces the tree size to 15 nodes, 8 of which are leaf nodes. Although the reduction is less than with the periodic attribute, we see another benefit to the modal attribute.

5. NODE-BASED REALIZATIONS

After experiencing improved performance with trees that incorporate the dip statistic and Lomb periodogram, we now consider classifying channels on WSN-class devices. When generating the earlier trees, the C4.5 algorithm had a wide range of attributes available to it, and the resulting trees included many of them. Computing those attributes on a wireless node, with the exception of the minimum, mean, and maximum, requires considerable memory and processing power. For in-node classification, we therefore explore smaller attribute sets to simplify the computations.

Recall that C4.5 builds trees from the top working down, and at each node, it selects the attribute that yields the purest children. In both the rapid-spikes and shifting-mean trees, our new attributes, periodic and modal, are the top attributes in their respective tree. For that reason, we first consider the performance of using just these attributes in a single-node decision tree.

We used C4.5 to generate a single-node decision tree based on the dip statistic alone, and it positively associated our modal attribute with the shifting-mean class as expected. Earlier, in Fig. 5, we show that the dip statistic, when combined with other attributes, greatly improves performance for all our sample sizes. In this case, however, the dip statistic alone performs poorly compared with the full C4.5 tree (Fig. 8, dip only), and this attribute should not be used alone to predict shifting-mean channels.

Given the poor performance of dip statistic alone, we also consider combining it with other easily-generated attributes, i.e., minimum, mean, and maximum (Fig. 8, dip+easy). In this case, we see improved performance at all sample sizes over the dip only case, but the performance still significantly lags behind using the full attribute set. Rather than compute all of the attributes used by the full shifting-mean classifier at this time (i.e., kurtosis, modal, moving-average standard deviation, periodic, and the 96 and 99.5 percentiles), we instead investigate the rapid-spikes classifier.

Like with the dip statistic, we generated a single-node decision tree based on the Lomb periodogram alone (Fig. 9). In this case, however, we observe that the single-decision tree performs very well for subsamples of at least 2000 observations. Apparently, this single attribute is responsible for most of the rapid-spike classifier's performance at large sample sizes. In the next section, we focus on approximating the Lomb periodogram so that we can implement it within WSN-class devices. This classifier is also particularly interesting given its role in identifying transmission opportunities in this class of channels.

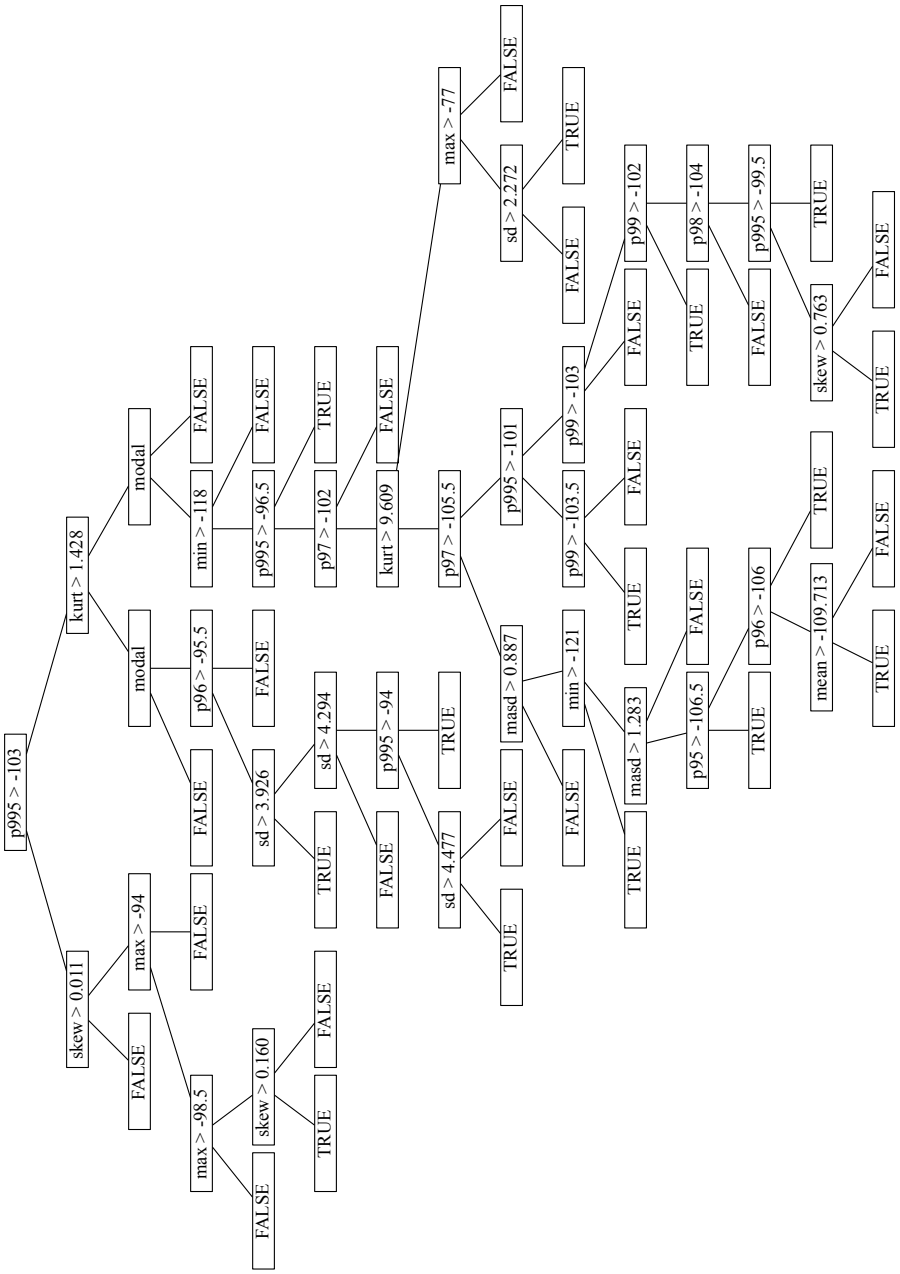


Fig. 7: When we exclude the *periodic* attribute, the C4.5 algorithm produces this tree to classify the rapid-spikes channels. For each decision node, the left child is the false case. The Boolean value in the leaf nodes indicates membership in the rapid-spikes class.

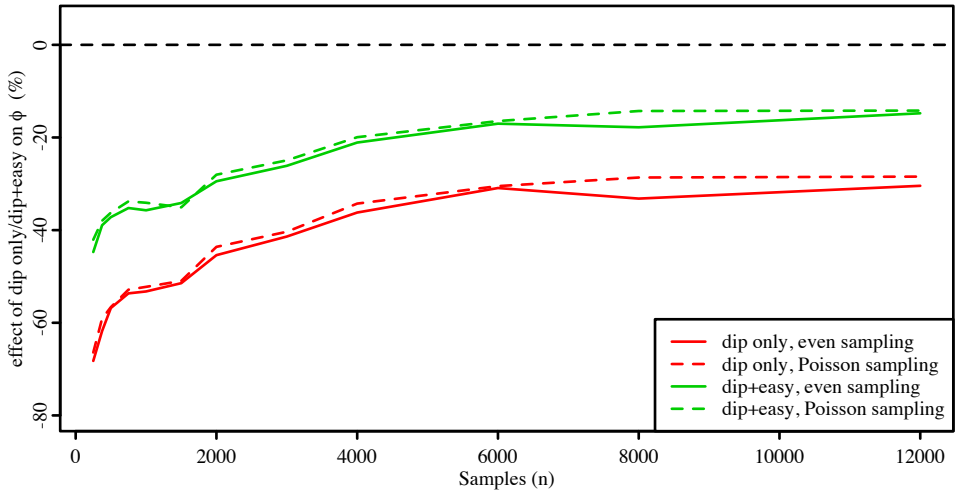


Fig. 8: The effect of using (a) only the dip statistic (dip only) and (b) the dip statistic plus other easily-computed attributes (dip+easy) on ϕ for identifying *shifting-mean* channels.

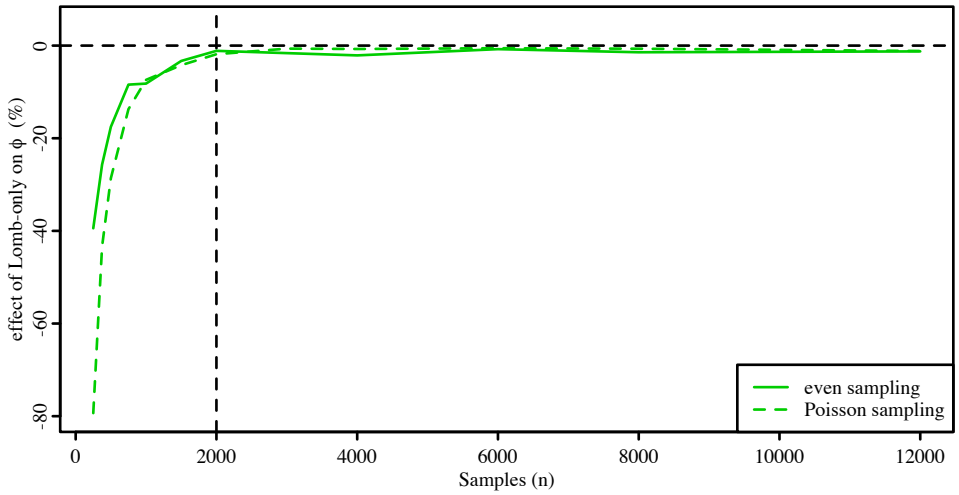


Fig. 9: The effect of using only the Lomb periodogram on ϕ for identifying *rapid-spikes* channels. Note that after 2000 samples, the Lomb periodogram alone predicts the channels almost as well as with the full attribute set.

5.1 `lombest`: an estimation of the Lomb periodogram

For a given frequency, the Lomb periodogram is defined by 7 and 8. In the exact case, the calculation

(1) requires recording all of the samples, which is necessary to compute the mean and variance used in 7 and τ used in 8,

(2) makes extensive use of the sine and cosine trigonometric functions,

(3) uses floating-point values throughout, and

(4) includes many floating-point multiplications and divisions.

In the last case, note that one integer-float multiplication or division on the TI MSP430 uses over 400 instruction cycles[4], while a register add or shift operation requires only one cycle [Texas Instruments 2006]. Given the limited memory and lack of hardware support for floating-point arithmetic and trigonometric functions in many microcontrollers, we need to consider approximations.

We aim for constant memory usage regardless of the sample size, and instead, we allow the usage to scale linearly with the number of analyzed frequencies. This setup allows us to collect the potentially large number of samples that may be required, and at the same time, it encourages us to reduce the memory footprint by searching only frequencies of interest. To make the construction of the algorithm feasible within mote-class devices, we

— calculate and evolve an integer estimate of the mean,

— disregard the τ parameter, and

— quantize the sine and cosine waves with discontinuities selected to allow bit-shifting.

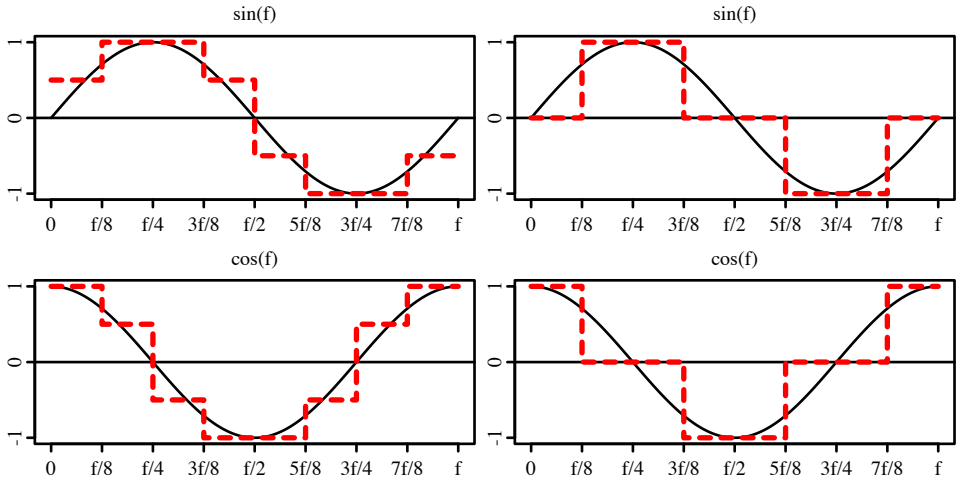
The following paragraphs describe each compromise in detail.

Note the use of the noise's mean $\bar{h}$ and variance σ^2 in 7, which in the exact case, requires the logging of all samples. Instead of taking this approach, we opt to (a) use an integer estimate of the mean that we calculate in advance using a small sample and (b) calculate the variance while collecting samples. To calculate the mean, we evaluated different sample sizes, and as expected, its value converges quickly as the sample size increases. At around 200 samples, we obtain a close approximation. Given a trace, we therefore use the first 10% for mean calculations and the remaining 90% for periodogram calculations. For example, at 2000 samples, the threshold for reasonable Lomb-only classification, the mean estimation uses 200 samples. To calculate the variance, we use floating-point numbers with one multiplication for each sample, and we perform the final divisions after accumulating this sum over the trace. For these multiplications and divisions, we could apply one of the techniques identified in [Texas Instruments 2006].

The mean plays a very important role in the algorithm, which assumes a random distribution of non-periodic samples around the mean. We are cautious in using a mean derived from a sample; therefore, we gradually adjust the mean as we sample for the actual periodogram. More specifically, for every sample that exceeds the mean, we increase the mean by 1 (and vice versa for samples below the mean).

The calculation of τ is another prerequisite for the exact periodogram. Lomb introduced this parameter to facilitate the statistical description of the least-squares spectrum [Lomb 1976]. For our approximation, however, we do not need such rigour, and intuitively, the periodogram changes little with shifts in time. We confirmed our suspicion by comparing several periodograms both before and after the removal of τ: we observed very little difference between the two cases.

[4]This cycle count assumes the C library included with IAR Embedded Workbench, version 3.41A.



(a) 4-level approximation (b) 3-level approximation

Fig. 10: Alternative approximations of the sine and cosine functions used when calculating the Lomb periodogram.

The sines and cosines in the exact formulation oscillate at the frequency of interest. When calculating the periodogram value for a particular frequency, we first identify each sample's location in the sine and cosine waves, and multiply the sample's magnitude by the corresponding values in those waves. We make a number of approximations here, both with identifying the sample's location in the wave and with the values of the waves.

To identify a location in a wave, we keep an array of timestamps, one for each frequency of interest, that indicates when the next wave begins. For each new sample, we update this array, which requires an integer comparison for each frequency and possibly a few additions. By knowing when the next wave begins for a given frequency, we can perform subtractions and additions to identify the offset into the current wave. This technique allows us to avoid the alternative – the modulo operation – which would be more expensive.

By quantizing the wave and making the discontinuities occur at opportune locations (bitshifts of the frequency), we can easily compute the offset into a wave and avoid divisions. We experimented with two rather crude (but effective) approximations of the sine and cosine functions (Fig. 10).

In the 4-level approximation, we use seven comparisons to determine the correct offset. Given an offset, we add/subtract the whole or (bit-shifted) half mean-adjusted sample to a sum depending on the sine or cosine amplitude at that offset. We record two integer sums for each denominator ($\sum_j \cos^2 \omega(t_j)$ and $\sum_j \sin^2 \omega(t_j)$): one for whole numbers and one for quarters. Using this approximation requires 6 short (2-byte) integers per frequency of interest.

In the 3-level approximation, we use four comparisons to determine the correct offset rather than seven. Given an offset, we need only add/subtract whole mean-

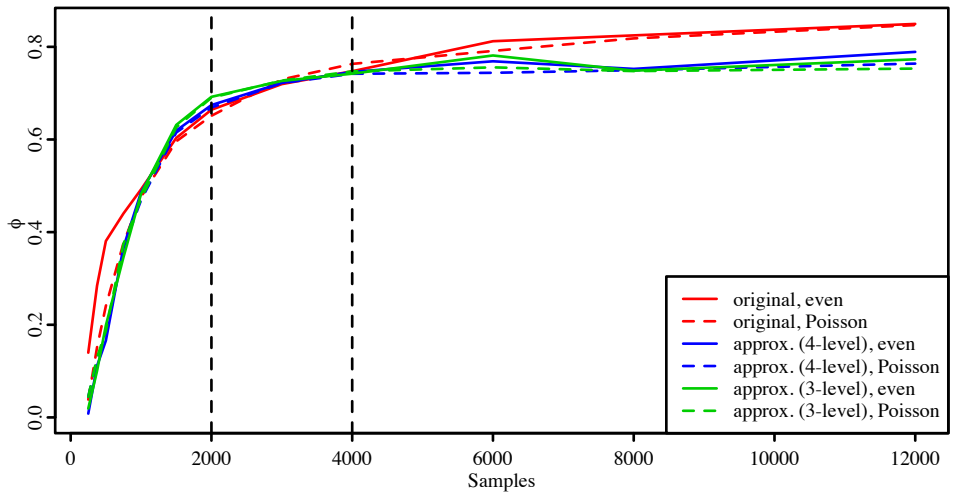


Fig. 11: The performance of classifying *rapid-spikes* channels when using only the Lomb periodogram to produce the attribute. This graph compares using the original Lomb periodogram against approximations based on our 3- and 4-level trigonometric approximations. At 2000 samples, our Lomb-only rapid-spikes classifier performs nearly as well as the full classifier. After 4000 samples, the performance when using the approximation-based attribute remains steady.

adjusted samples, and we now only encounter zero and one in the denominator. In this case, we only require 4 short integer sums per frequency.

In Fig. 11, we compare the classification accuracy of the original and 4- and 3-level trigonometric approximations. Recall that the Lomb-only classifier begins to perform comparably to that of the full attribute set at 2000 samples. When using between two and four thousand samples, the performance of our sine and cosine approximations closely matches the original. After four thousand samples, the performance of the original periodogram continues to slightly increase while the approximations remain constant. The 4- and 3-level approximations perform similarly across all sample sizes. Overall, the approximations perform very well when compared with the exact periodogram, with the 3-level being preferred given its lower requirements.

6. CONCLUSION

In this paper, we described our work towards enabling WSN devices to better understand the interference in their environment. We first collected noise traces simultaneously from a four-by-four grid of nodes at 5000 Hz – a much higher sampling frequency than previously described in the literature.

In selecting a classifier, we considered the energy and space constraints of a WSN node. We focused on decision trees, since after training, the classification of new cases is simple.

Given high-frequency interference traces, we investigated different subsampling techniques. We compared even with uneven/Poisson sampling and did not see the expected benefits of the latter, and in hindsight, we are not surprised. Different

even samples of a given size would capture a consistent fraction of spikes, and thus produce consistent periodogram responses. With Poisson sampling, however, samples would capture varying fractions of spikes. In cases that captured fewer spikes, a lower periodogram response could fail to meet the predefined threshold.

We investigated the classification of samples according to distinct channel patterns: shifting-mean and rapid-spikes. Each of these can be exploited by cognitive radio avoidance strategies: in the former, changing channels and in the latter, carefully timing transmissions. We found two features that aided in recognizing these patterns. For the shifting-mean channels, we exploited the sample's multi-modal distribution by using the dip statistic. For the rapid-spikes channels, we exploited their periodic nature by calculating the Lomb periodogram. In both cases, the new attributes improved the classification performance for all subsample sizes. We also observed that both of these new attributes reduced the size of their respective decision trees, particularly in the *periodic* case.

We then explored using one-node decisions trees to classify the shifting-mean and rapid-spikes channels. The dip statistic alone was poor predictor of shifting-mean channels. We managed to improve its performance by adding other easily calculated attributes (minimum, mean, and maximum), but it still performed significantly worse than the full attribute case. The Lomb periodogram, when used alone, performed quite well, particularly when collecting more than 2000 samples. Finally, we showed a simplification of the Lomb periodogram that could be implemented within the limited resources of a mote-class device.

ACKNOWLEDGMENTS

The authors thank the Natural Sciences and Engineering Research Council (NSERC), Alberta Advanced Education and Technology, and the Killam Trusts for financial support.

REFERENCES

Akhmetshina, E., Gburzynski, P., and Vizeacoumar, F. 2003. PicOS: A tiny operating system for extremely small embedded platforms. In *Embedded Systems and Applications*, H. R. Arabnia and L. T. Yang, Eds. CSREA Press, 116–122.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C., and Nielsen, H. 2000. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics 16,* 5, 412.

Boers, N. M., Chodos, D., Huang, J., Stroulia, E., Gburzynski, P., and Nikolaidis, I. 2009. The Smart Condo: Visualizing independent living environments in a virtual world. In *PervasiveHealth '09: Proceedings from the 3rd International Conference on Pervasive Computing Technologies for Healthcare*. London, UK.

Boers, N. M., Nikolaidis, I., and Gburzynski, P. 2010. Patterns in the RSSI traces from an indoor urban environment. In *CAMAD '10: IEEE 14th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks*. Coconut Creek, FL.

Chandra, A. 2002. Measurements of radio impulsive noise from various sources in an indoor environment at 900 MHz and 1800 MHz. In *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*. Vol. 2. 639–643.

Chipcon. 2005. Note on CC1100 device discrepancies. Technical documentation, application note.

Cramér, H. 1962. *Mathematical Methods of Statistics*. Asia Publishing House, Bombay.

Do, J., Akos, D., and Enge, P. 2004. L and S bands spectrum survey in the San Francisco Bay area. In *PLANS 2004: Position Location and Navigation Symposium*. 566–572.

Douglas, D. and Peucker, T. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization 10,* 2, 112–122.

Galassi, M., Davies, J., Theiler, J., Gough, B., Jungman, G., Alken, P., Booth, M., and Rossi, F. 2009. *GNU Scientific Library Reference Manual*, 3 ed. Network Theory Ltd.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. 2009. The WEKA data mining software: An update. *SIGKDD Explorations 11,* 1.

Hartigan, J. A. and Hartigan, P. M. 1985. The dip test of unimodality. *The Annals of Statistics 13,* 1, 70–84.

Hartigan, P. M. 1985. Algorithm as 217: Computation of the dip statistic to test for unimodality. *Journal of the Royal Statistical Society. Series C (Applied Statistics) 34,* 3, 320–325.

Kotsiantis, S. B. 2007. Supervised machine learning: A review of classification techniques. *Informatica: An International Journal of Computing and Informatics 31,* 3 (Oct.), 249–268.

Kotz, S., Johnson, N. L., and Read, C. B., Eds. 2006. *Encyclopedia of Statistical Sciences*, 2 ed. Wiley-Interscience.

Lee, H., Cerpa, A., and Levis, P. 2007. Improving wireless simulation through noise modeling. In *IPSN '07: Proceedings of the 6th International Conference on Information Processing in Sensor Networks*. ACM, New York, NY, USA, 21–30.

Lomb, N. R. 1976. Least-squares frequency analysis of unequally spaced data. *Astrophysics and Space Science 39*, 447–462.

Matthews, B. W. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure 405,* 2, 442 – 451.

Musaloiu-E., R. and Terzis, A. 2008. Minimising the effect of WiFi interference in 802.15.4 wireless sensor networks. *International Journal of Sensor Networks 3,* 1, 43 – 54.

Newton, H. J. 2007. Lecture 2: The periodogram. Lecture notes, STAT 685, Texas A&M University.

Oehen, P. and Plattner, B. 2006. DSNAnalyzer: Backend for the deployment support network. M.S. thesis, Swiss Federal Institute of Technology (ETH) Zurich.

Olsonet Communications Corp. 2008. Platform for R&D in sensor networking. `http://www.olsonet.com/Documents/emspcc11.pdf`.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. 1992. *Numerical Recipes in C: The Art of Scientific Computing*, 2 ed. Cambridge University Press, Cambridge, MA.

Quinlan, J. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Rusak, T. and Levis, P. 2010. Physically-based models of low-power wireless links using signal power simulation. *Computer Networks 54,* 4, 658 – 673.

Srinivasan, K., Dutta, P., Tavakoli, A., and Levis, P. 2006. Understanding the causes of packet delivery success and failure in dense wireless sensor networks. In *SenSys '06: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*. ACM, New York, NY, 419–420.

Srinivasan, K., Dutta, P., Tavakoli, A., and Levis, P. 2010. An empirical study of low-power wireless. *ACM Transactions on Sensor Networks 6,* 2, 1–49.

Texas Instruments. 2006. Efficient multiplication and division using MSP430. Application Report SLAA329.

Texas Instruments. 2007. RSSI interpretation and timing. Design Note DN505, SWRA114B.

Texas Instruments. 2009. Data sheet for CC1100: Low-power sub-1 GHz RF transceiver.

Witten, I. H. and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Yucek, T. and Arslan, H. 2009. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Communications Surveys Tutorials 11,* 1 (First quarter), 116–130.