A Low-Cost Prototype for an Online High-Impedance Vegetation Fault Detection Module

Douglas Pinto Sampaio Gomes
College of Engineering and Science
Victoria University
Melbourne, Australia
douglas.uf@gmail.com

Cagil Ramadan Ozansoy
College of Engineering and Science
Victoria University
Melbourne, Australia
cagil.ozansoy@vu.edu.au

***Abstract*—This paper presents a low-cost feasibility prototype for vegetation High-Impedance Fault (HIF) detection. The method relies on high-frequency sweep sampling based signal processing for classifying high-dimensional network signals. The prototype is set to perform signal processing calculations for feature extraction and classify signals with an ensemble of boosted decision trees. A card-sized single board computer was used to sample and classify analog voltage signals online. The demonstrated performance proved that at least two implementation concerns could be addressed with this low-cost prototype. These are the effects of environmental noise on detection accuracy and computational complexity (cost) in high-dimensional data processing. Evaluation of the experimental set-up was accomplished by different experiments and their respective results were used as evidence of a successful feasibility prototype. Results show that sampling only one power cycle at 50 Hz can be sufficient for an accurate HIF detection. A classification accuracy of 95.23% was achieved in hardware for split data tests compared to 97.53% accuracy in desktop analysis of the same dataset.**

Keywords—classifier, design, feasibility prototype, high impedance faults, noise, sweep sampling, vegetation conduction.

I. INTRODUCTION

HIGH Impedance Faults (HIFs) can generate small fault currents not easily distinguishable from regular increases in the load current. The subject of most papers has been on the exclusive proposition of HIF detection algorithms, mainly validated offline, with synthetic or small real data sets [1]. To authors' best knowledge, the work in [2] represents the single recent exception to this, proposing a prototype to detect HIFs based on inter harmonics features extracted from fault currents.

In [3], the authors argued for two main probable reasons for the absence of a conclusive solution: the broader treatment received by HIFs, and the low-resolution sampling in existing protection apparatus. The first is given by over-generalizing and non-discriminative representation of high-impedance surfaces that compose the fault [4-6]. The second is due to digital relays commonly having sampling rates in the order of a few thousand samples per second. Both issues were addressed in [3] with an approach dedicated to vegetation surfaces using high-resolution sampling with fault current (I_f) threshold set to 0.5 A. Features were extracted by Fourier and Wavelet transforms methods, and classification was performed by a boosted decision trees classifier with 98% overall accuracy. The developed method was proposed as a fault detection classifier explicitly developed for vegetation HIFs rather than one that works for a variety of contact surfaces. This was achieved using an extensive data set from staged vegetation faults. This work presents classification performance of the built signal-processing and decision-making module implementing the proposed method. The research also enabled evaluation of direct concerns such as the computational complexity and resilience to environmental noise. Experiments have proven that classification power could be enhanced, with less computation complexity, by implementing changes using the signal processing methodology. The prototype implements a feature extraction procedure superior to that in [3], by dropping all Periodogram features and using only the 4-level detailed components of the Multi-Resolution Analysis (MRA).

The hardware for embodying signal-processing and decision making functionalities is an accessible card-sized computer board [7]. The board is set-up to process and classify signals online as well as their sampling using an Analog-to-Digital Converter (ADC). This is undertaken in a similar manner to [8] by first converting the fault recordings to analog signals, via a desktop computer's sound card, and streaming them to the board. The signals are received by the board via an external USB sound card, which is an ADC that digitizes and streams the sequences to be processed by the board. Hardware in this HIF detection scheme are commercial, low-cost products that represent a crude and restricted implementation to attest to the method's feasibility.

The novel contributions of this work include (i) a simplified feature extraction method based on the use of 4-level detailed components of the Multi-Resolution Analysis approach (ii) development of a low-cost signal-processing and decision-making module (iii) assessment of the impact of noise on the classification accuracy (iv) introduction of a 'sweep sampling' based signal processing approach for classifying high-dimensional signals (v) feasibility justification of the classifier using real data from 'wire into vegetation' faults.

II. BACKGROUND AND MOTIVATION

A robust aspect of [3] was the conceptualization of a novel method using data from a large number of niche vegetation HIFs staged as part of the '*Vegetation Conduction Ignition Testing'* project. The project [9] sampled diverse vegetation species in staged HIF tests leading to a database of fault signals used herein. Prior work [3] used recordings of High Frequency (HF) voltage signals in place of current signals. This transition into a voltage-based approach discerned the proposed method from standard current-based approaches [10, 11]. The authors found no useful features to distinguish the classes in the Low-Frequency (LF) voltage signals. A

potential solution was only established by exploring the HF voltage signals (>10 kHz) [12].

HF sampling necessitates high-dimensional data processing and storing. HF sampling was considered beneficial in [9] for better characterizing fault signatures, but a 'sweep sampling' method was incorporated to deal with the high-dimensionality problem. This involved sampling a small period of signals at every second of a full power cycle at 50 Hz (20 ms). Such a non-traditional approach can generate some concerns [13-15] with respect to computational complexity and noise resilience. This prompted authors to deploy the machine learning classifier on an embedded system validating viability of implementing the required computational complexity on a low-cost device.

The methodology to empirically address these goals is inspired by [8], where authors showed how general-purpose programmable devices can aid in overcoming issues in DSP education. The effective sampling frequency of the developed prototype was 40 kHz, due to the 96 kHz frequency rate limit from commercial sound cards used in streaming and sampling signals. That is a fraction of the sampling rate used in the original recordings of the fault tests. However, the use of the sweep sampling mode meant that there was a large idle time (98% of each second) between each sweep, allowed a coping strategy. This consisted of stretching the 20 ms sweep to last close to a full second, basically reducing the effective sampling frequency to 40 kHz. As further explained in the system design section, despite adopting such a drastic procedure, the board still receives the same amount of information as would the decision-making module of a fault detection apparatus that relied on the sweep sampling mode (one sweep per second).

III. DATA SET, FEATURE EXTRACTION AND CLASSIFICATION

The sampled voltage sweeps, used as observations in the supervised learning method, could adopt either the Fault (1) or the Non-Fault (0) label. Each sweep recording was composed of 40k samples from 20 ms sampling at the 2 MS/s rate. Only the first in-fault sweep of each test was used for fault observations. This chosen sweep was the one sampled after $I_{f\text{-}RMS}$ reached 0.5 A. 566 fault observations were used in the supervised learning stage. The non-fault observations were extracted differently. As tests started at the same time as the rig was energized, no pre-fault recordings were produced. There were, yet, recordings made in different periods during the test days to characterize the background noise of the network. These recorded sweeps constitute all the non-fault observations used when learning the classifier. In total, 1132 observations, half of each class, were used. Observations were arranged to compose the dataset as in [3]. Feature extraction was formerly given as a mixture of Fourier and Wavelet transforms features. This was simplified and enhanced resulting in higher overall accuracy and reduced computational complexity, while using the same number of features. This complexity reduction was achieved by dropping all Periodogram (from Fourier) features and using only the 4-level components of the MRA approach [3]. The final eight features are from two simple calculations made in the four signals: the L_1 norm, and the 'log energy' measurement given by (1) and (2). As there are four levels in the MRA, each calculation results in four features. These four levels sum up to the final eight attributes into the classifier's algorithm input.

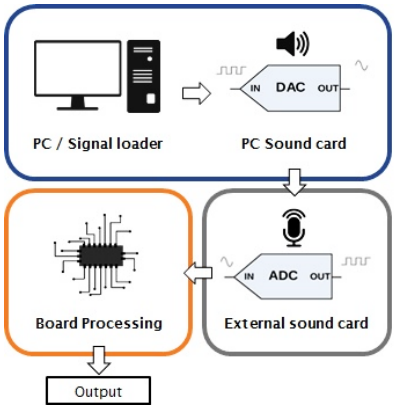

Fig. 1. Experimental set-up diagram.

$$L_1[i] = \sum_n |d_i[n]| \quad (1)$$

$$En_{log}[i] = \sum_n log(d_i[n])^2 \quad (2)$$

Where i is the decomposition level, and
$\boldsymbol{d_i[n]}$ is the detail sequence of level i.

The classification of signals was performed by the AdaBoost [16] algorithm with decision trees [17] as its weak learners. It consists of a boosting (averaging and re-weighting) algorithm for weak, greedy, classification learners. Or more simply, the average votes of an ensemble of weighted decision trees. The number of grown trees was set as 100, with the maximum number of splits now settled to be four. These are shallow trees, less prone to over-fitting with simpler decision boundaries.

IV. SYSTEM DESIGN

This section presents the basis regarding claims made on the feasibility prototype. The role and schema of hardware and software pieces are described as well as the experimental procedure proposed to evaluate the prototype's performance.

A. Hardware Set-Up

The set-up has three distinct parts: a desktop computer, an external USB sound card, and a single board computer as in Fig. 1. The desktop computer loads the original signals generating and streaming audio signals through the on-board audio codec, which constitute a 24-bit DAC. Audio signals represented the HF voltage sweeps from both classes of observations. The USB sound card is a 24-bit ADC that sampled the audio signals via a TRS cable, representing a data acquisition hardware. The board embodied the calculation and decision making module running the signal processing and machine learning algorithms. Streaming and sampling were set at 48 kHz with one second of sampling relating to 40k values of a voltage sweep, plus a small space of zeros (remaining 8k) in between sweeps. The small space helps to differentiate each recording and compensate for any small sample delays that the set-up might introduce. The

board has an ARM® 720 MHz processor, 256 MB of RAM, and runs a Linux distribution (Debian®) for embedded/IoT devices. Given such a simplistic arrangement, a low Signal-to-Noise Ratio (SNR) is likely. The constructed low SNR environment was critical in validating the resilience of the prototype against noise.

B. Software Set-Up

Access to the external Pulse-Code Modulation was required to sample streamed values. Processing and classifying implied exporting the codes, conceptualized in MATLAB, to another language. The code deployment required all functions in a common language to be cross-compiled from a Windows® machine to a Linux ARM architecture executable. In the main code, a buffer of 48k float values stores the sampled signals during calculations. In this manner, one second of sampling is needed, in the established sampling rate, to fill all the buffer spaces. This buffering results in the first detection delay of one second. During this buffering delay period, all calculations of the previous sweep take place. While the next sweep buffers, the present one gets processed and used to generate a state label. The ensemble/compiling tasks of source files were done in the Microsoft Visual Studio Community® environment, which also cross-compiled to the ARM architecture. These software tools enabled to deploy full machine learning models to the board.

V. EXPERIMENTS

Experiments were undertaken to evaluate the quantization process noise and validate the used classifier. The board's performance in extracting features, processing, and classifying signals was also tested. Noise quantification was undertaken by streaming, recording and calculating SNR as well as error measurements of an arbitrary number of signals. The classifier was validated by a 10-fold cross-validation approach. The third and fourth experiments tested the classification performance of the prototype with both in-sample and out-of-sample data.

A. Noise Quantification

In streaming and sampling, the DAC and ADC were sources of quantification noise. Noise evaluation of the whole set-up is more difficult, and thus performed by an empirical approach through four measurements made on an arbitrarily high number of samples. The results were derived from calculations made on 25 random observations of 40k samples each, resulting in a million streamed/sampled values. Noise quantification was made by comparing four standard signal measurements calculated from the original sampled signals. These were the SNR, L_1, L_2, and L_{inf} errors norms given by (3-6).

$$SNR_{dB} = 10\log_{10}\frac{\sum_{i=1}^{n}|x_i^1|^2}{\sum_{i=1}^{n}|x_i^1-x_i^2|^2} = 10\log_{10}\frac{\sigma_{signal}^2}{\sigma_{noise}^2} \quad (3)$$

$$L_{1err} = \frac{\|x_i^1-x_i^2\|_1}{\|x_i^1\|_1}\times 100 = 1{,}000\log_{10}\frac{\sum_{i=1}^{n}|x_i^1-x_i^2|}{\sum_{i=1}^{n}|x_i^1|} \quad (4)$$

Table I
Noise quantification results

	SNR_{dB}	$L_{1err}(\%)$	$L_{2err}(\%)$	$L_{inf}(\%)$
Mean	17.56	17.92	14.73	16.65
Min	9.97	5.03	5.83	6.01
Max	24.68	53.43	31.72	41.89

$$L_{2err} = \frac{\|x_i^1-x_i^2\|_2}{\|x_i^1\|_2}\times 100 = 1000\log_{10}\frac{\sqrt{\sum_{i=1}^{n}|x_i^1-x_i^2|^2}}{\sqrt{\sum_{i=1}^{n}|x_i^1|^2}} \quad (5)$$

$$L_{inf} = \left(1-\frac{\|x_i^2\|_\infty}{\|x_i^1\|_\infty}\right)\times 100 = \left(1-\frac{sup_i|x_i^2|}{sup_i|x_i^1|}\right)\times 100 \quad (6)$$

Where $\boldsymbol{x_i^1}$ is the original signal sequence, $\boldsymbol{\sigma^2}$ is the variance, $\boldsymbol{x_i^2}$ is the discrete signal recorded by the board, and

B. Classifier Validation

As feature extraction modifications were made in the data process step before supervised learning, validation of the classifier had to be performed again. This is needed to test the classifier's ability to generalize out of sample data and not overfit the data set in the learning process. The standard practice of 10-fold cross-validation method was performed. Doing so meant partitioning the data set in ten equal parts. All parts, except one, were used for training and the remaining one for testing. This is repeated iteratively with all parts until all samples are used for testing exactly once.

C. Whole Data Set Training

Testing the classifier on the same data used in the training has no value in generalization or assessment of classification performance. This experiment involved investigating noise effects introduced by the experimental set-up by training the classification algorithm with the whole data set and deploying it to the board. All original signals were streamed by the desktop, sampled, and classified by the board. The desktop classification result is expected to differ from that by the board, giving a quantitative measure of noise effects in the decision boundaries created by the classifier.

D. Split Set Training

The fourth experiment used the classical data set split method to examine the noise effects. The data was partitioned into two equal parts: one to train the algorithm and the other to test it. The trained algorithm was deployed to the board and tested with the same out-of-sample data as the desktop. The comparison of classification accuracies was quantitatively presented as a measure of the prototype's performance.

VI. RESULTS

The noise quantification measurements are shown in Table I. Despite having outliers, the observed signal measurements compose a skewed (towards minimum) and narrow probability distribution. The max values were given by an outlier signal that had short and rapid spikes. Inspired by empirical testing, our hypotheses is that fast transitions and saturated values were more difficult to accurately reproduce in the streaming part, and sample in the data acquisition part by the hardware.

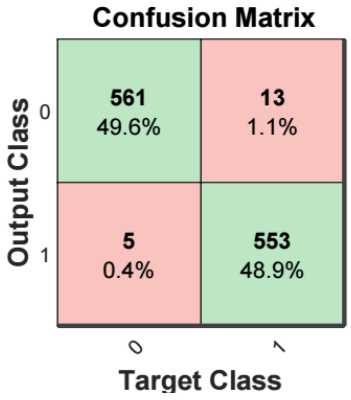

Fig. 2. Classifier's confusion matrix.

Table II
Fault Test Data

Test #	I_{init} (A)	I_{limit} (A)	Species	Moisture (%wt.)	Diameter (mm)	Fire Result
335	1.858	4	B. Marginata	61.3	30	0
504	0.023	1	R. Fruticosus	31.8	15	1
515	0.281	2	R. Fruticosus	50	-	1

A. Cross-Validation Results

The 10-fold cross-validation results are shown in Fig. 2. The results show higher overall accuracy (98.5%) when compared (98.06%) to the former classifier in [3]. It must be noted that this higher accuracy was achieved with one Discrete Wavelet Transform (DWT) instead of two transforms (DWT + FFT). This simplification was achieved by dropping all Periodogram features and using only the 4-level detailed components of the MRA. This also reduced the computational complexity.

B. Desktop Classification

Table II shows selected valid ignition tests [9] with pre-fault data. In these tests, conduction started after energisation making it possible to visualise the work of the classifier over pre-fault and fault durations. As set in Table II, current thresholds (I_{limit}) were set between 0.5 to 4 A. Tests were terminated upon I_f reaching set thresholds. 'Bush' tests staged 'wire into vegetation' faults where a live conductor fell into a bush without touching the ground.

Test 335 was a 'bush' test where the energised conductor was dropped into the 'Banksia Marginata' bush. I_f begins to flow prior to the 14th second. Marxsen [9] recommended 0.5 A fault detection sensitivity as appropriate and feasible for extreme fire risk days. The recommendation in [9] was that "if a powerline protection system can detect and respond to an earth-fault drawing 0.5 A, fire risk from 'wire into bush' faults might be cut by about 80%". The marker labelled '1' in Fig. 3 (c) indicates the time when I_f reaches 0.5 A. As shown, the classifier was able to correctly classify the fault. Test 335 is also the only test which was later tested on the board; results of which are further discussed in Section (7).

Fig. 4(c) shows another bush test on the same species 'R. Fruticosus' where the I_{limit} was limited to 1 A. As shown, the classifier was capable of correctly classifying the fault well before I_f reaches 0.5 A. The classifier achieves the same for Test 515 (see Fig. 5) for the same species with a higher moisture content.

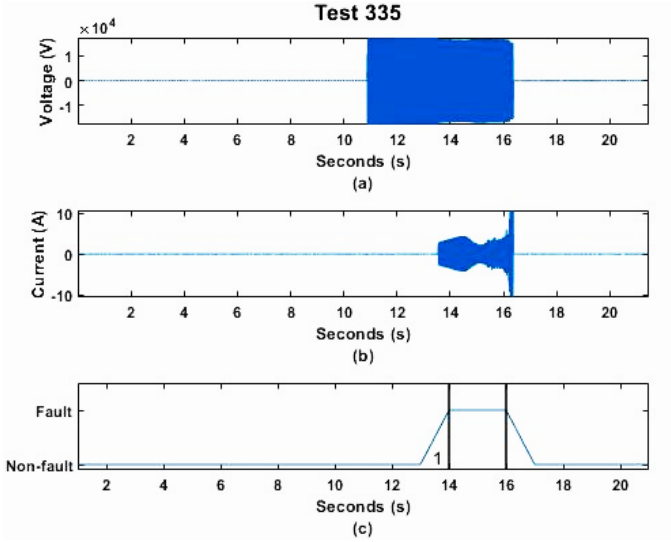

Fig. 3. Test 335 Classification (a) Voltage (b) current (c) classifier output.

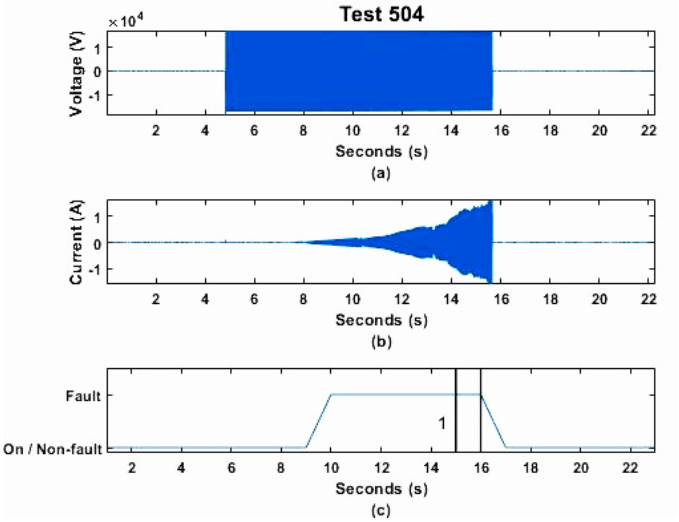

Fig. 4. Test 504 Classification (a) Voltage (b) current (c) classifier output.

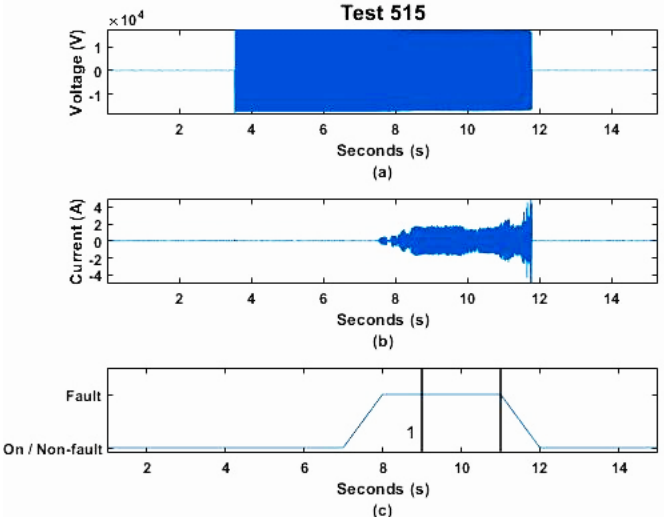

Fig. 5. Test 515 Classification (a) Voltage (b) current (c) classifier output.

Table III
Board vs. Desktop classification results

	Desk. Acc.	Board acc.
Whole data*	100%	96.47%
Split Data	97.53%	95.23%

*No learning/generalization value

C. Classification Accuracy versus Noise

Table III shows the results of third and fourth experiments highlighting algorithm's resilience to environmental noise with only a slight change in accuracy. The word '*desktop*' is used to describe results obtained in MATLAB, and '*board*' to depict results from experiments in the prototype. In the whole data experiment, the desktop computer's 100% result was expected with all testing undertaken with the *in-sample* data. While ideal streaming and sampling would have resulted in the same result for the board, the board suffered a slight decrease in accuracy due to noise. 40 out of the 1132 tests were misclassified by the board, equating to a classification accuracy of 96.47%. Results from the split data experiment are also worth noting. The marginally reduced desktop accuracy (97.53%), when compared to its cross-validation result (98.5%) is notable. This is probably due to only half of the data points being used to train the algorithm here, while the cross-validation counted with a 9/10 ratio at each test. In most adequate cases, more data points result in better generalization ability and, consequently, higher classification accuracy. The second point is related to the reduced, but yet high, accuracy presented by the board in a 17 dB SNR environment. 14 out of 566 tests (half of the data set) were misclassified by the desktop, while the board only mistakenly labelled 27 of the same signals. This display of accurate classification in a noisy environment and the ability to work with high dimensional data in sweeps is one of the main contributions of this paper and evidence of the prototype's feasibility. Moreover, this evidence does not corroborate with previously mentioned practical concerns and dismissing of the use of wavelets and high dimensional data for HIF detection.

D. Impact of Noise Level on the Accuracy

A noise versus accuracy experiment was made to solidify this claim and the algorithm's robustness to noise as in Fig. 6. The methodology involved measuring the accuracy (by cross-validation) against different noise levels (in dB) when artificial white noise was added to the recorded HF signals. A noteworthy feature of Fig. 6 is pointed out by the marker at 18 dB. By adding the same noise level as the one measured in the prototype environment (18 being the closest integer to 17.56), the desktop accuracy (96.64%) was close to the one presented by the board classification in the split test scenario (95.23%). This shows that the white noise added in the MATLAB environment has similar effects as the noise in the real hardware set-up. The slight difference is likely to be due to the use of cross-validation in this test. The same effect was presented when comparing the cross-validation and split test scenario accuracy given by the desktop (98.5% to 97.53%).

VII. BOARD CLASSIFICATION OF TEST 335

Another feature was added to further increase reliability and visibility. The board was set to not only process and classify the sweeps online, but also to plot the classified signals in an LCD cape. Fig. 7 shows the LCD display plot of the classified sweeps for Test 335. The plot is updated at every second after the sampling buffer gets flushed and values of the present sampled sweep are classified. After classification, signals were down-sampled to 512 values (for illustration purposes) and displayed in red, if a fault is detected, or in green, if it is labelled as a non-fault sweep. Fig. 8(a-b) shows the images, exemplified using the graph utility *Gnuplot*, screening sweeps classified in both classes, non-fault and fault, respectively.

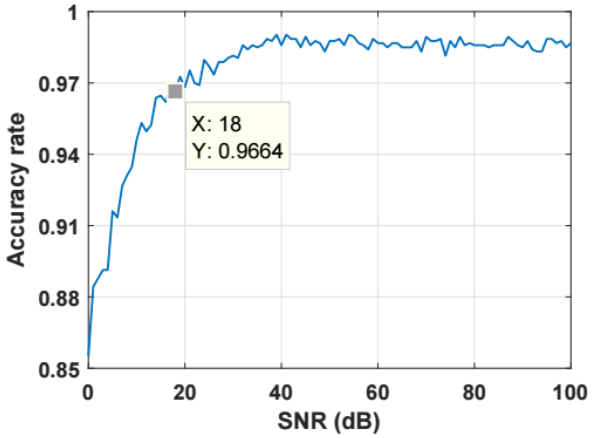

Fig. 6. Accuracy versus noise plot.

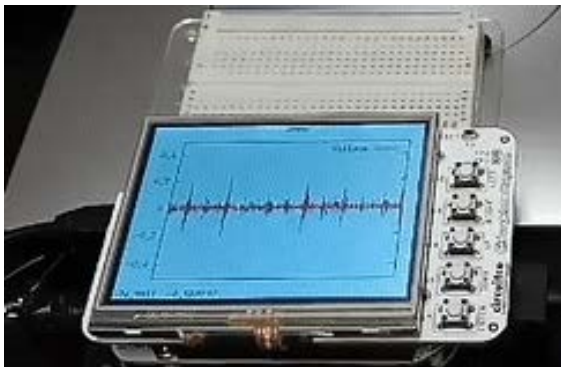

Fig. 7. Implementation of an LCD cape to plot the classified signals.

VIII. CONCLUSION

A feasibility prototype of a HIF detection module has been proposed herein explicitly for vegetation HIFs. The prototype sampled analog signals, which reflected stretched high-frequency sweeps from fault signals, processing, and classifying them. Sweep sampling based signal processing, was proposed herein, as a solution to high computational complexity resulting from HF sampling as that translates to more samples in a given period. The approach was built on the assumption that a small sample of system signals would still infer key conclusions regarding its states. Validation results have supported the research hypothesis that sweeps have enough predictor information to accurately classify faults. A low computational overhead was achieved by implementing a feature extraction procedure that used only 4-level detailed components of a MRA approach without any Periodogram features. This has reduced the computational complexity and increased the overall classification accuracy. Responsible for the detection, the boosted decision trees (AdaBoost) algorithm with enhanced features presented itself as a powerful classifier.

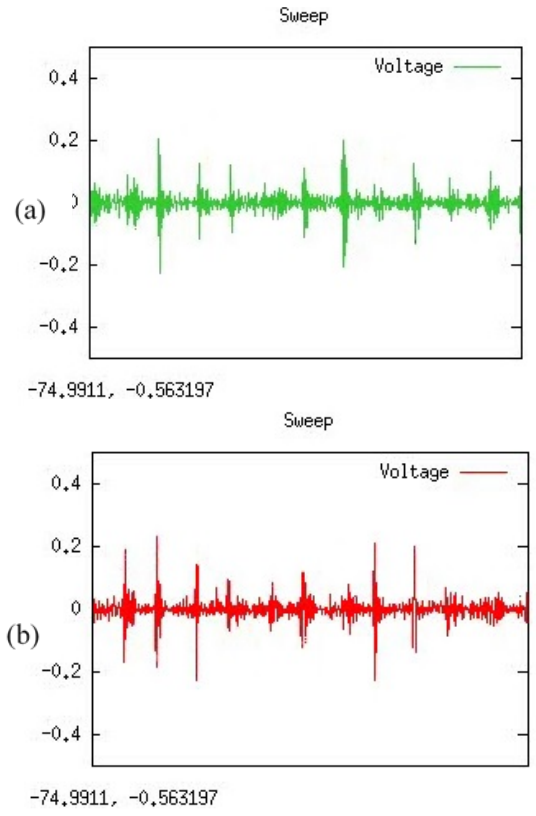

Fig. 8. (a) A Non-Fault sweep. (b) A Fault sweep.

A low SNR environment was constructed in validating the resilience of the prototype against noise. The classification accuracy given by the hardware set-up with high environmental noise suffered a slight decrease when compared to the result from the MATLAB environment. In the 'split data' experiment, the board suffered a slight decrease in accuracy, 95.23%, compared to the 97.53% desktop accuracy due to noise. By adding the same noise level as the one measured in the prototype environment, a reduction in the desktop accuracy was accounted. This demonstrates that white noise added in the MATLAB environment would have similar effects on the accuracy as would noise present in a real hardware set-up.

The research findings can be used as evidence of addressing possible concerns regarding the computational complexity and robustness to environmental noise. The results attested for the feasibility of a low-cost signal processing and classification module that could help mitigate bushfires ignited by high-impedance faults in power lines. In the presented cases, the classifier was able to correctly classify the fault before the fault current reached the set 0.5 A threshold. This validates that the proposed scheme is sensitive to very small fault currents and can potentially lead to an 80% reduction in the fire risk.

In the present prototype, one second is spent buffering the signal, with almost another full second to classify it. Such a system would require at most three seconds for a classification result. Despite being a relatively long time to detect a fault, a three-second delay represents a detection delay sufficiently low enough to greatly mitigate the vegetation fire ignition risks from powerlines. The reliability can be improved by increasing the security (false positives) of the classifier. This can be achieved by simply claiming a detection after subsequent positives, i.e. only two consecutive fault sweeps attest for a fault detection. Further works will continue to focus on possible improvements in signal representation and machine learning classifiers, and certainly on the possibility of developing a full-size prototype.

ACKNOWLEDGEMENTS

Authors commend the Powerline Bushfire Safety Program for funding the staged faults and making research data available. This work was supported in part by the Department of Economic Development, Jobs, Transport and Resources.

REFERENCES

[1] M. Sedighizadeh, A. Rezazadeh, and N. I. Elkalashy, "Approaches in High Impedance Fault Detection - A Chronological Review," *Advances in Electrical and Computer Engineering*, vol. 10, no. 3, pp. 114-128, 2010.

[2] J. R. Macedo, D. Carvalho, J. W. Resende, F. C. Castro, and C. A. Bissochi, "Proposition of an interharmonic-based methodology for high-impedance fault detection in distribution systems," *IET Generation, Transmission & Distribution*, vol. 9, no. 16, pp. 2593-2601, 2015.

[3] D. P. S. Gomes, C. Ozansoy, and A. Ulhaq, "High-Sensitivity Vegetation High Impedance Fault Detection based on Signal's High-Frequency Contents," *IEEE Transactions on Power Delivery*, vol. PP, no. 99, pp. 1-1, 2018.

[4] J. Chen, T. Phung, T. Blackburn, E. Ambikairajah, and D. Zhang, "Detection of high impedance faults using current transformers for sensing and identification based on features extracted using wavelet transform," *IET Generation, Transmission & Distribution*, vol. 10, no. 12, pp. 2990-2998, 2016.

[5] M. Kavi, Y. Mishra, and M. D. Vilathgamuwa, "High-impedance fault detection and classification in power system distribution networks using morphological fault detector algorithm," *IET Generation, Transmission & Distribution*, vol. 12, no. 15, pp. 3699-3710, 2018.

[6] É. M. Lima, C. M. d. S. Junqueira, N. S. D. Brito, B. A. d. Souza, R. d. A. Coelho, and H. G. M. S. d. Medeiros, "High impedance fault detection method based on the short-time Fourier transform," *IET Generation, Transmission & Distribution*, vol. 12, no. 11, pp. 2577-2584, 2018.

[7] Gerald Coley, "BeagleBone Black System Reference Manual," Texas Instruments, 2013.

[8] G. Pasolini, A. Bazzi, and F. Zabini, "A Raspberry Pi-Based Platform for Signal Processing Education [SP Education]," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 151-158, 2017.

[9] T. Marxsen, "Vegetation Conduction Ignition Test Report - Final," Marxsen Consulting Pty Ltd.2015, Accessed on: Nov-2016.

[10] A. H. Etemadi and M. Sanaye-Pasand, "High-impedance fault detection using multi-resolution signal decomposition and adaptive neural fuzzy inference system," *IET Generation, Transmission & Distribution*, vol. 2, no. 1, pp. 110-118, 2008.

[11] S. R. Samantaray, B. K. Panigrahi, and P. K. Dash, "High impedance fault detection in power distribution networks using time-frequency transform and probabilistic neural network," *IET Generation, Transmission & Distribution*, vol. 2, no. 2, pp. 261-270, 2008.

[12] D. P. S. Gomes, C. Ozansoy, and A. Ulhaq, "Vegetation High-Impedance Faults' High-Frequency Signatures via Sparse Coding," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 7, pp. 5233-5242, 2020.

[13] A. Soheili, J. Sadeh, and R. Bakhshi, "Modified FFT based high impedance fault detection technique considering distribution non-linear loads: Simulation and experimental data analysis," *International Journal of Electrical Power & Energy Systems*, vol. 94, pp. 124-140, 2018/01/01/ 2018.

[14] K. Sekar and N. K. Mohanty, "Data mining-based high impedance fault detection using mathematical morphology," *Computers & Electrical Engineering*, vol. 69, pp. 129-141, 2018.

[15] P. E. Farias, A. P. de Morais, J. P. Rossini, and G. Cardoso, "Non-linear high impedance fault distance estimation in power distribution systems: A continually online-trained neural network approach," *Electric Power Systems Research*, vol. 157, pp. 20-28, 2018.

[16] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *icml*, 1996, vol. 96, pp. 148-156.

[17] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81-106, 1986.