Anomaly-aware Adaptive Sampling for Electrical Signal Compression

K. R. Sahasranand, Francis C Joseph, Himanshu Tyagi, *Senior Member, IEEE*, Gurunath Gurrala, *Senior Member, IEEE*, Ashish Joglekar

***Abstract*—Intelligent electronic devices for power systems often entail high frequency sampling of electric signals, enabled to capture anomalous signal behavior. However, in normal operation this oversampling is redundant and leads to excessive data being stored or transmitted. This gives rise to a new compression problem where the collected samples should be further subsampled and quantized based on the presence of an anomaly in the underlying signal. We propose an Anomaly-aware Compressive Sampler (ACS) which tests the signal for the presence of an anomaly in a block of samples, and subsamples in a hierarchical manner to retain the desired sampling rate. ACS has been designed keeping hardware constraints in mind, using integer operations, an appropriate bit-packing, a simple iterated delta filter, and a streaming data pipeline. We present a mathematical formulation of the problem and analyze the performance of ACS, establishing theoretically its ability to identify anomalies in the signal and adapt the sampling rate. ACS competes with the state-of-the-art algorithm for the better-behaved transmission system data from DOE/EPRI, and outperforms it significantly on real-time distribution system data recorded in our laboratory. Finally, ACS is lightweight and was implemented on an ARM processor.**

***Index Terms*—anomaly detection, hypothesis testing, signal compression, streaming implementation**

I. Introduction

Real-time monitoring and fault diagnosis in voltage and current data is one of the distinguishing features of smart grid. Intelligent Electronic Devices (IEDs) deployed across the electrical grid are a key enabler of the required distributed sensing apparatus. These IEDs record sampled voltage and current data, often at high rates, and communicate the recorded data to the control center to enable real-time monitoring and detection and analysis of anomalies. A desired feature of this setup is the ability to identify short time power quality disturbances and subcycle transients. These include switching transients and disturbances due to lightning strikes, harmonic distortion, voltage sags/swells, oscillatory transients at low to medium frequencies, and voltage/current imbalances [1]. In order to capture the subcycle transients, the prescribed number of samples per cycle is 256 to 512 [1], [2], namely a sampling frequency of roughly 31 kHz. Moreover, often the same IED hardware handles multiple channels corresponding to different phases and data arrives at the same high rate in each channel. The IEDs using edge computing devices are usually resource-constrained and storing data at this rate is infeasible. At the same time, it is undesirable to use expensive communication bandwidth to transmit the raw recorded data. This necessitates compression of the recorded data at the edge. While compressing a stream at aforementioned data rates using standard techniques is trivial for powerful modern computers, it is a significant overhead for limited edge hardware, which is already strained executing other edge analytics algorithms. Further, one needs to establish a data pipeline where the compression of a block of streaming data is completed in time before the next block arrives.

Note that the high sampling rate is prescribed to enable the capture of high frequency anomalous signal behavior, which itself happens infrequently. Thus, it is possible to achieve compression by identifying the presence of anomalies and adjusting the sampling frequency accordingly. We propose such a compression algorithm, *Anomaly-aware Compressive Sampler* (ACS) and use it to enable high-rate data acquisition at a cheap, hardware-constrained IED. ACS evaluates each data block received from the FIFO buffer in the data pipeline for the presence of anomaly and retains the losslessly compressed data only when there is an anomaly. If there is no anomaly, ACS allows for a configurable lossy compression while retaining the desired power quality measurement accuracy. This is achieved by dropping a part of the block thereby effectively storing a downsampled version of the signal. Thus, the algorithm *adjusts* the sampling frequency adaptively according to the largest frequency actually present in a particular block.

To enable the anomaly detection component required in ACS, we need to extract a feature from the time-series data block that is amenable to anomaly detection. Also, to compress the same block we need an "innovation filter" that extracts from each block $x = (x[0], \ldots, x[n-1])$ an innovation sequence $y = (y[0], \ldots, y[n-1])$. ACS reduces the computational cost by using the same filter to extract both the signature for anomaly detection and compression of the innovation sequence.

Recent literature in electrical signal compression suggests

This work was supported by the Bosch Research and Technology Centre, Bengaluru, India and by the Robert Bosch Centre for Cyber-Physical Systems, Indian Institute of Science, Bengaluru 560012, India (under Project E-Sense: Sensing and Analytics for Energy Aware Smart Campus).

K. R. Sahasranand is with the Department of Electrical Communication Engineering, Indian Institute of Science, Bengaluru 560012, India.

Francis C Joseph is with the Department of Electrical Engineering, Indian Institute of Science, Bengaluru 560012, India.

Himanshu Tyagi is with the Department of Electrical Communication Engineering and also with the Robert Bosch Centre for Cyber-Physical Systems, Indian Institute of Science, Bengaluru 560012, India.

Gurunath Gurrala is with the Department of Electrical Engineering and also with the Robert Bosch Centre for Cyber-Physical Systems, Indian Institute of Science, Bengaluru 560012, India.

Ashish Joglekar is with the Robert Bosch Centre for Cyber-Physical Systems and also with the AI and Robots Technology Park, Indian Institute of Science, Bengaluru 560012, India.

Email: {sahasranand, francisj, htyagi, gurunath, ashishj}@iisc.ac.in

several techniques for obtaining y from x, ranging from wavelet filters to machine learning based methods [3], [4]. However, such techniques are computationally heavy and their feasibility in a streaming implementation on a processor with low compute-power such as ARM is unclear. Instead, we take recourse to a variant of simple difference coding that has been prescribed recently for data compression for IoT [5], [6]. Using offline data analysis, we choose to use an iterated Δ filter (a second order difference filter) where we convert $x[i]$ to $y[i] = x[i] - 2x[i-1] + x[i-2]$. Note that it is a linear filter with coefficients that are powers of 2 and can be easily implemented on hardware.

The sequence of integers $y[0], \ldots, y[n-1]$ thus obtained must still be represented using a fixed resolution as mandated for signed integers by hardware. But their values are very small and we can gain further compression by bit packing ($cf.$ [6]). This is an optimization in the implementation step whereby the bit representation of the integers in the sequence are compactly stored using a minimum number of bytes. Furthermore, since the iterated Δ filter is designed to attenuate the low frequency signal in normal operation, the compression is significantly higher when there is no anomaly. ACS uses this difference in compressed lengths to detect anomalous behavior. At a high level, our approach leverages the information theoretic understanding of redundancy caused by model mismatch – our compressor designed for normal operation has higher redundancy under anomaly. A preliminary version of the proposed algorithm was presented in [7]. In [8], we have presented the hardware implementation results of the algorithm on an ARM processor. This paper deals with the theoretical analysis of the proposed algorithm, comparison with the state-of-the-art, and rigorous testing on practical data sets.

In recent years, data compression for electrical signals recorded in smart grids has received a lot of attention; see [9]–[14] for a glimpse of results with different applications. Common techniques used are transform coding, which includes wavelet-based [3] and other transform based techniques [15], [16], lossless coding following an alternative representation such as image [17] and mixed transform parametric coding, which attempts to remove the "seasonality" in the data through estimation [18]; see [19] for a survey. Algorithms for detecting events in electrical signals based on different metrics abound [20]–[22]. However, the hardware constraints in our setting is a dimension not accounted for in many of these works. In particular, the algorithm must be designed to operate in real-time on streaming data and to complete the event detection as well as the evaluation of the compression mapping for a block before the next data block arrives. This requires that the blocklength n be small, which is the length of sample buffer the algorithm must maintain. At short blocklengths, it is impossible to use DFT-based estimation methods to *separate* the high frequency anomalies and the low frequency signal. This is because DFT when evaluated on a small number of samples collected at a very high sampling frequency, has poor frequency resolution. Further, the presence of subharmonics and deviations around the fundamental frequency could result in a DFT magnitude spectrum which is close to the magnitude spectrum in the case of a high frequency component actually present, due to spectral leakage resulting from windowing, and spectral sampling. Goertzel algorithm [23] which is used to detect individual frequencies rather than the whole spectrum is not immune to spectral leakage either. The standard technique used to mitigate spectral leakage, namely the usage of windows with shorter main-lobe width, renders the frequency resolution of the DFT poorer. Besides, when the signal is nonstationary and has higher uncertainty in instantaneous frequency, DFT-based techniques are unreliable. Parametric spectral estimation methods (see [24]) such as Prony, Pisarenko harmonic decomposition, MUSIC, and ESPRIT require at least as many samples as the ordinality of the highest harmonic that we are trying to detect and are computationally heavy (and involve floating-point operations), both of which are undesirable in a streaming implementation. Our algorithm, on the other hand, extracts a signature using a simple difference filter which facilitates event detection as well as compression and operates on short blocklengths. Our approach is related also to the one used in recent work on integer compression for massive databases in [25]. In particular, our algorithm operates entirely on integers and avoids completely the need for any kind of floating-point computation. Indeed, our reconstruction procedure entails spline interpolation which involves floating point operations, but it is not carried out on the low-compute hardware. Another desirable feature that enables easy fault diagnosis and low latency is for the algorithm to facilitate selective reconstruction of a few cycles of interest rather than the entire signal of a long duration. Thus, entropic compression techniques such as LZW (as in [16]), which incur a high delay when implemented on processors with low compute-power such as ARM, are not suitable. This re-emphasizes the need for an algorithm that is "lightweight", operates block-by-block, and permits short blocklengths.

Perhaps the work closest to our setting is [11] where a compression scheme called G-PQDR, based on dissimilarity between normal and anomalous cycles of data, is proposed. G-PQDR involves storing the wavelet coefficients of a reference frame (and those of frames "dissimilar" to the reference frame), and estimating and storing only the fundamental frequency in frames similar to the reference frame. However, we observe that on distribution systems data (as against transmission systems), techniques that rely on instantaneous frequency estimation tend not to perform well due to higher variability in measured frequency. We have implemented the scheme for comparison, and found that the performance of ACS is better compared to G-PQDR on distribution system data. Besides, G-PQDR operates on blocklengths as large as 4 cycles which results in a higher latency in real-time applications. Reference [21], which proposes universal waveshape-based disturbance detection using similarity metrics, points out some of the drawbacks of the G-PQDR scheme as well. However, on transmission system data, G-PQDR yields performance comparable to our scheme. In particular, we have evaluated the performance of both the algorithms on DOE/EPRI National Database Repository of Power System Events [26] data. Both the algorithms were able to detect most of the events for fixed parameter settings and achieve good compression.

In the theory of practical, streaming compression algo-

rithms, the mean squared error of reconstruction of a periodic signal and a bandlimited signal from values quantized after oversampling is shown to vary inversely with the oversampling rate in [27] and [28], respectively, and investigated further in [29]. Reference [30] shows that it is possible to achieve exponential decay of the quantization error as a function of the oversampling rate. However, none of these works consider the setting with uncertainty in the frequency band. Namely, available theory literature does not characterize the dependence of quantization error on the oversampling rate, when the signal could consist of either only low frequencies or contain a high frequency anomaly as well. Our proposed scheme addresses this important problem; we leave a formal analysis of the scheme in this direction as future work.

The remainder of the paper is organized as follows. The next section describes the problem formulation and proposes a solution. An elaboration of the proposed algorithm is presented in Section III. Section IV provides a theoretical analysis of the anomaly detection block of the algorithm following which experimental results are presented in Section V. Section VI concludes the paper.

II. Problem Formulation and Proposed Solution

Our goal is to compress voltage and current signals sampled at a high rate. The data is fed to the compression algorithm, block-by-block, via a FIFO buffer. The compression algorithm takes as input a block of the sampled signal and produces a bit stream as output. It is desirable to complete the "processing" of a block of data before the next block arrives to ensure that no data is lost or overwritten and to reduce the latency caused by accumulating data over a period of time. The formal description of the problem is provided below.

A. Problem description

A real-valued signal $x(t)$ is sampled at a high frequency f_s Hz and the samples are quantized to r-bit resolution by an analog-to-digital converter (ADC) to yield $x[k] = x(k/f_s)$, $k \in \mathbb{N}$, which take values in the set of integers, $\{-2^{r-1}, \ldots, -1, 0, 1, \ldots, 2^{r-1}-1\}$. Further, n consecutive samples are grouped into blocks $x^{(0)}, x^{(1)}, \ldots$ where block $x^{(i)}$ comprises the samples $x[ni], ..., x[n(i+1)-1]$. We seek to design a compression map that converts the block of data $x^{(i)}$ into a bit sequence B_i in a streaming fashion. This is depicted in Fig. 1.

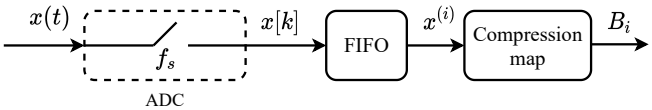


Fig. 1: Schematic of the problem setting

The signal $x(t)$ is nonstationary and hence at different times could consist of high frequencies (corresponding to anomalies) in addition to the low frequencies which are always present. While the purpose of high sampling frequency is capturing anomalies, under normal operation it generates a lot of samples, all of which need not be retained for reconstruction. One measure of reconstruction accuracy is *normalized mean-squared error* (NMSE) for recovering the sampled sequence over a finite time horizon N, given by

$$\frac{\sum_{k=1}^{N} \left(\hat{x}[k] - x[k]\right)^2}{\sum_{k=1}^{N} x[k]^2},$$

where $\hat{x}[\cdot]$ are reconstructed samples. The performance of the algorithm at a fixed NMSE level will be measured by the *compression ratio* (CR) defined as

$$\frac{rnD}{\sum_{i=1}^{D} |B_i|},$$

calculated over D blocks and $|B_i|$ denotes the length of the bit sequence B_i. The formulation above prescribes NMSE as the measure of accuracy for the compression algorithm. This is standard practice. However, fault analysis in power system requires more accurate, nearly lossless, reconstruction of the signal near the anomaly. The algorithm must be able to offer a variable NMSE performance, with lossless recovery around anomaly and lossy recovery during normal operation. This discovery of anomaly region must be in-built in the algorithm.

B. High-level approach

In the absence of an anomaly, discarding a subset of the samples collected at a high sampling frequency results in compression while not violating the Nyquist criterion. The compression algorithm *tests* every block for anomalies and, if found, compresses the block in a lossless fashion. If no anomaly is found, the block could be compressed in a lossy fashion by discarding a subset of samples. In effect, the high sampling frequency f_s Hz is *retained locally* in blocks where it is essential; the data is downsampled otherwise. This anomaly-aware adaptivity is enabled by solving a hypothesis testing problem using a small number of samples (equal to the length of the block).

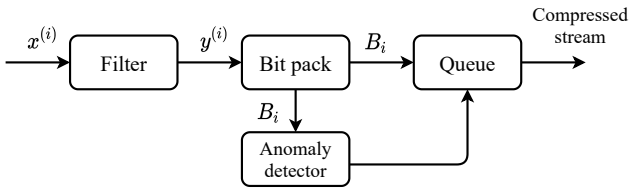
III. Anomaly-aware Compressive Sampler

We now elaborate further on ACS, our proposed compression algorithm summarized in the introduction. A schematic summary of ACS is provided in Fig. 2 and the complete description is given in Algorithm 1. The i-th block of data $x^{(i)}$ (comprising n samples) is passed through an iterated Δ filter (Algorithm 2). The output $y^{(i)} = y^{(i)}[0], \ldots, y^{(i)}[n-1]$ of the filter is passed through a bit packing procedure to obtain the bit packet B_i which is obtained as follows. The bit sequence corresponding to the first and the second sample in $y^{(i)}$ are stored separately. The smallest number of bits needed to represent each of the remaining $(n-2)$ samples in block i, $\ell^{(i)}$ and the $\ell^{(i)}$-bit representation of the $(n-2)$ samples are stored together via bit packing (Algorithm 3). This bit packet B_i is fed to an anomaly detector which labels block i as anomalous if $\ell^{(i)}$ exceeds a threshold τ, and normal otherwise. Normal blocks are compressed in a lossy fashion, but not immediately since we need to retain a few cycles of data before and after the anomaly. The bit packet B_i resulting from the input block $x^{(i)}$ is pushed into a FIFO queue of length `BUF` when no anomaly is detected. If no anomaly is detected during

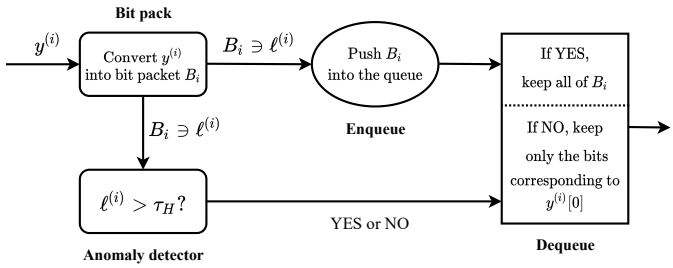
the time a packet B_i enters the queue and reaches the head of the queue, it is marked to be stored in a lossy fashion. This is achieved by storing only the part of the B_i corresponding to the first sample in block i. This, in effect, is tantamount to downsampling by a factor of n.

This is repeated at a higher level as well. If m consecutive blocks are marked to be stored in a lossy fashion, namely, retaining only the first sample, then these first samples are passed through the same iterated Δ filter and the output is used to decide whether to retain all of the m values or just the first one among them. In particular, if the number of bits per sample resulting from this filter, $\ell_B^{(i)} > \tau_B$, a fixed threshold, all m values are retained. Otherwise, all values except the first one are dropped. This process of downsampling the data first by n, and then by m if needed, results in a hierarchical subsampling. Indeed, it is possible to repeat this process beyond two levels; we restrict ourselves to two-level hierarchical subsampling.

At a high level, the iterated Δ filter and bit packing are designed to compress the samples well when only low frequencies are present. Thus, in the presence of a high frequency anomaly, the filter output deviates from the compression model and hence bit packing incurs more bits to represent the output samples. Note that the statistic used by bit packing is the smallest number of bits needed to represent *each* of the output samples and hence an anomaly is detected even if it does not exist throughout the block. Thus, the statistic works even for nonstationary signals in which anomalies or faults are rare events that do not last long. If the entire block of length n is void of anomalies, all but the first sample in the block are dropped, thereby resulting in a downsampling by a factor of n. This sampling frequency, f_s/n Hz suffices to reconstruct the signal in this block, since it does not contain high frequencies that impose a Nyquist rate of f_s Hz. If consecutive m blocks do not contain an anomaly, the entire $mn-1$ samples could be dropped for the same reason.



(a) High-level schematic of ACS



(b) A more detailed schematic of ACS

Fig. 2: Schematic summary of the proposed algorithm

Algorithm 1: Anomaly-aware Compressive Sampler (ACS)

Input: Stream of samples X, n, m, r, BUF, τ_H, τ_B
Steps: Repeat for $i = 1, 2, \ldots$,

1: Read the block $x^{(i)}$ from the stream X.
2: **Filter** :
   1) Pass $x^{(i)}$ through an iterated Δ filter (Algorithm 2) to get the sequence $y^{(i)} = y^{(i)}[0], \ldots, y^{(i)}[n-1]$.
   2) The sequence $y^{(m_i+1)}[0], \ldots, y^{(i-1)}[0], y^{(i)}[0]$ comprising the first samples $y^{(i)}[0]$ from $i - m_i$ such consecutive blocks is again passed through the same filter where m_i is the largest multiple of m smaller than i.
3: **Convert into bit packet** :
   1) The first two elements of $y^{(i)}$ are converted into a bit sequence and stored first.
   2) Using Algorithm 3, the remaining elements of $y^{(i)}$ are jointly converted into a sequence of bits B_i and the number of bits used per sample, $\ell^{(i)}$ noted.
   3) If i is a multiple of m, the number of bits $\ell_B^{(i)}$ needed to store the sequence $y^{(i-m+1)}[0], \ldots, y^{(i-1)}[0], y^{(i)}[0]$ using Algorithm 3 is also noted.
4: **Enqueue** : $\ell_B^{(i)}$ and B_i are pushed to the queue.
5: **Anomaly detection** : If $\ell^{(i)} > \tau_H$, declare that block i contains an anomaly.
6: **Dequeue** : Skip this step if $i <$ BUF. Let $b = \min\{0, i - 2 \cdot \text{BUF}\}$.
   1) If no block j is declared to contain an anomaly, for $i \geq j > b$, pop the packet P^* at the head of the queue and mark it to be stored in a lossy fashion.
   2) For m consecutive packets marked to be stored in a lossy fashion, if $\ell_B^{(i)} > \tau_B$, store the bytes B^* corresponding to the first sample $y^{(i)}[0]$ in P^* and discard the next $m-1$ packets. Otherwise, store the bytes B^* corresponding to $y^{(i)}[0]$ in all m packets.
   3) If some block j is declared to contain an anomaly, for $i \geq j > b$, store the bytes B^* in P^* losslessly.

A. Building blocks of ACS

Now we describe each block of the algorithm. All numbers are represented as signed or unsigned integers. Let $|a|$ denote the absolute value of the number a, and $[a]_r$ denote the r-bit representation of a. For bit sequences a and b, let $a \star b$ denote the concatenation of a and b.

1) Filtering: We implement a two-level linear filter. Consider m consecutive blocks $x^{(0)}, \ldots, x^{(m-1)}$, each comprising n consecutive samples. First, each block $x^{(i)}$ is passed through an iterated Δ filter as described in Algorithm 2 (Note that a slightly different operation is carried out on the first two samples since the filter is applied block-by-block and a second order difference filter requires two previous samples to operate). Then, the vector of first samples of each of the m blocks is passed through the same filter.

Since the data is obtained by oversampling a signal that comprises low frequencies most of the time, difference filters produce small values under normal operation and thus aid compression [31]. In our experiments, the iterated Δ filter produced smaller values compared to a Δ filter that takes only first order differences. For more details on delta modulation for quantizing oversampled signals, see Section 7.4 in [31].

Algorithm 2: Iterated Δ filter

Input: Signed integer array $x[0], \ldots, x[a-1]$.
Output: Signed integer array $y[0], \ldots, y[a-1]$.
Steps:

1: $y[0] \leftarrow x[0]$.
2: $y[1] \leftarrow x[1] - x[0]$.
3: For $k = 2, \ldots, a-1$,

$$y[k] \leftarrow x[k] - 2x[k-1] + x[k-2].$$

2) Convert into bit packet: We adopt the Variable Byte scheme [25], Elias coding [32], and bit packing algorithm [6] adaptively to convert the innovation sequence $y^{(i)}$ into bit packet B_i. The block is first zigzag coded [33] to handle negative integers; a sample a is mapped to $2a$ if a is non-negative, and to $(2|a|-1)$ if a is negative. The first two values of the zigzag coded block $\widetilde{y}^{(i)}$, namely $\widetilde{y}^{(i)}[0]$ and $\widetilde{y}^{(i)}[1]$ are typically larger than the rest of the values in the block, and hence stored separately using a procedure similar to the Variable Byte scheme [25] – the values are stored with the prefix 0, 10, or 11 if the value (including the prefix) can be stored using 1, 2, or 3 bytes, respectively. For the remaining samples, the algorithm chooses either Elias Gamma code [32] or bit packing (Algorithm 3), the one that yields the smallest number of output bits. Note that the number of output bits under either scheme can be calculated (and hence this choice can be made) without actually performing the encoding. This is done for each block separately and a bit indicating the scheme used is inserted into the bit representation. The above procedure is completely lossless since the first two samples may be extracted from the respective bit representations stored separately and the bit packing (as well as the Elias Gamma code) that is used to store the remaining $n-2$ samples can be inverted.

Bit packing (described in Algorithm 3) serves two purposes. One is to convert the samples into a sequence of bits in the following manner: each of the zigzag coded samples $\widetilde{y}^{(i)}[2], \ldots, \widetilde{y}^{(i)}[n-1]$ is represented using $\ell^{(i)}$ bits where $\ell^{(i)}$ is the smallest number of bits using which each of the above samples can be represented losslessly; the value of $\ell^{(i)}$ is also stored. Secondly, the number of output bits incurred by bit packing, more precisely $\ell^{(i)}$, is used as a *statistic* to test for the presence of anomalies in that block. Note that $\ell^{(i)}$ takes values between 0 and r since the original samples are of resolution r bits to begin with, and hence $\ell^{(i)}$ itself could be stored using $\log_2 r$ bits[1].

[1]If the number of bits required turns out to be $r-1$ or r, we represent the samples using r bits itself, so that $\ell^{(i)}$ takes one of only r possible values, namely $0, 1, \ldots, r-2, r$.

Algorithm 3: Bit pack

Input: Unsigned integers $w[0], \ldots, w[a-1]$, and r
Output: Bytes `fBits`
Steps:

1: Find the position ℓ of the leftmost 1 in the bitwise OR of $w[0], \ldots, w[a-1]$.
2: Set `fBits` $\leftarrow [\ell]_{\log_2 r}$.
3: For $k = 0, 1, \ldots, a-1$,

$$\texttt{fBits} \leftarrow \texttt{fBits} \star [w[k]]_\ell.$$

3) Enqueue: For every m-th block $x^{((i+1)m-1)}$, $\ell_B^{(i)}$, the position of the leftmost 1 in the bitwise OR of $\widetilde{y}^{(im)}[0], \ldots, \widetilde{y}^{((i+1)m-1)}[0]$ is calculated. $\ell_B^{(i)}$ and packet B_i are pushed into the queue.

4) Anomaly detection: If the value of $\ell^{(i)}$ computed in Algorithm 3 is greater than τ_H, anomaly is declared and the *dequeue* function is invoked with a flag `lossless` set to `true`. This indicates to the dequeue function to store the next `BUF` packets to be stored losslessly. If $\ell^{(i)} \leq \tau_H$, the flag `lossless` is set to `false` which signals to the dequeue function to store the packet at the head of the queue in a lossy fashion.

5) Dequeue: The algorithm *dequeue* is invoked with a flag `lossless` which is set to `true` if the bit packet B^* in the packet P^* popped from the head of the queue is to be stored in a lossless fashion, `false` otherwise. If m consecutive packets, $x^{(im)}, \ldots, x^{((i+1)m-1)}$, for some i, are marked to be stored in a lossy fashion, the number of bits per sample required to store only the first samples of these m packets, $\ell_B^{(i)}$ is compared with a threshold τ_B. If $\ell_B^{(i)} \leq \tau_B$, only the bytes corresponding to the first sample of the first enqueued packet among the latest m packets is stored; otherwise compressed bytes corresponding to all m packets are stored. A special character (byte) is included at the beginning of a bit sequence whenever the algorithm switches between lossless compression, compression dropping a single packet, and compression dropping m consecutive packets.

6) Reconstruction: In blocks i where all the samples are retained, the first two samples are stored separately (see Section III-A2) and are readily extracted. To obtain the remaining $n-2$ samples, the inverse of the bit packing operation is applied to B_i, which yields the zigzag-coded sequence $y^{(i)}$. The zigzag coding is inverted, followed by the application of an inverse[2] iterated Δ filter to obtain $x^{(i)}$.

Reconstruction of parts of the signal which were dropped, is performed using splines [24]. Linear splines are used to interpolate between the available first values in consecutive blocks where $mn-1$ samples are dropped and cubic splines in those parts where $n-1$ samples are dropped. This choice gave a better NMSE in our experiments. Indeed all $mn-1$ samples are dropped if the iterated Δ filter in both the levels yield small values; heuristically, this corresponds to the second

[2]All the aforementioned inversions are straightforward; hence we do not describe them in detail here.

derivative being small and hence the signal in the m blocks is well-approximated by a line.

Remark – Note that the compression algorithm involves calculating differences of integers (filtering), bitwise OR of unsigned integers (calculation of $\ell^{(i)}$), shifting of bits (zigzag coding and bit packing), and comparisons (anomaly detection). All the aforementioned operations are linear in the number of bits and hence the algorithm is computationally efficient. In particular, none of the blocks involve floating point operations (Indeed, reconstruction entails interpolation which involves floating point operations, but it is not carried out on the low-compute hardware). This makes ACS suitable for execution on low-compute hardware. Further, the algorithm uses the same *signature*, namely $\ell^{(i)}$, for compression as well as anomaly detection; this is useful in a streaming implementation where no step involves "revisiting" data that has been processed already.

IV. Analysis of the Anomaly Detection Algorithm

Here we present a theoretical analysis of the anomaly detection block of our algorithm. The identification of presence or absence of anomaly in a given block is modeled as a hypothesis testing problem, wherein we consider a simplified representation of the sampled signals. The null hypothesis $\mathcal{H}_0$ is that the observed signal is a sum of L low frequency sinusoids, and the alternative hypothesis $\mathcal{H}_1$ is that the signal is a sum of H harmonics of a high frequency, in addition to the L low frequency sinusoids. Under either hypotheses, measurement noise (modeled as a sequence of independent and identically distributed sub-Gaussian[3] random variables) and quantization noise (which is bounded) are added to the signal. Then, we proceed to evaluate the condition under which the statistic, namely the maximum of the absolute value of the filtered sequence, *separates* the two hypotheses while satisfying certain requirements on the probability of error. That is, we seek a threshold τ such that the algorithm, which declares $\mathcal{H}_0$ when the statistic is less than τ and $\mathcal{H}_1$ when the statistic is greater than τ, outputs the correct hypothesis with high probability. In particular, we identify the condition which enables us to choose a threshold τ such that under $\mathcal{H}_0$, the statistic is less than τ with high probability and under $\mathcal{H}_1$, the statistic is greater than τ with high probability.

Notation – $\mathbb{P}_{\mathcal{H}_i}[A]$ denotes the probability of an event A under hypothesis $\mathcal{H}_i$. $\mathbb{E}[X]$ denotes the expectation of random variable X. For a positive integer M, $[M]$ denotes the set $\{1, 2, \ldots, M\}$ and $\mathcal{S}(M)$ denotes the set $\{a, a+1, \ldots, a+M-1\}$ for any non-negative integer a. For a real number a, $\lfloor a \rfloor$ denotes the largest integer less than or equal to a. For $1 \leq p \leq \infty$, $\|x\|_p$ stands for the $\mathcal{L}^p$-norm of the signal x.

Consider a block of n samples $x[0], \ldots, x[n-1]$ obtained by sampling a real-valued signal at frequency f_s Hz. We observe samples $\overline{x}[k] = \mathcal{Q}\left(x[k] + z[k]\right)$ where $\mathcal{Q}(\cdot)$ is an r-bit quantizer and $z[k]$ are independent and identically distributed sub-Gaussian random variables with mean 0 for $k = 0, \ldots, n-1$, and seek to resolve the following hypothesis testing problem.

$$\mathcal{H}_0 : x[k] = \sum_{l=1}^{L} \sin\left(\omega_l k t_s + \theta_l\right),$$

$$\mathcal{H}_1 : x[k] = \sum_{l=1}^{L} \sin\left(\omega_l k t_s + \theta_l\right) + \sum_{h=1}^{H} \sin\left(h\widetilde{\omega} k t_s + \widetilde{\theta}_h\right),$$

where $t_s = 1/f_s$, and for $l \in [L]$, $\omega_l = 2\pi f_l$, $f_l \leq f_a < \widetilde{f} \leq H\widetilde{f} < f_s/2$, and $\widetilde{\omega} = 2\pi\widetilde{f}$. That is, we seek to determine if the signal has only low frequency components or it has higher frequency components as well. A test is a (randomized) mapping which takes as input $\overline{x}[k], k = 0, \ldots, n-1$, and declares $\mathcal{H}_0$ or $\mathcal{H}_1$. A test T constitutes a (δ, ϵ) test if

$$\mathbb{P}_{\mathcal{H}_0}\left[T \text{ declares } \mathcal{H}_1\right] < \delta,$$
$$\mathbb{P}_{\mathcal{H}_1}\left[T \text{ declares } \mathcal{H}_0\right] < \epsilon.$$

An iterated Δ filter (Algorithm 2) is applied to the block comprising n samples $x[0], \ldots, x[n-1]$ to obtain $y[0], \ldots, y[n-1]$. Namely, for $k = 2, \ldots, n-1$,

$$\begin{aligned} y[k] &= \overline{x}[k] - 2\overline{x}[k-1] + \overline{x}[k-2] \\ &= x[k] - 2x[k-1] + x[k-2] + w[k] \\ &\quad + q[k] - 2q[k-1] + q[k-2], \end{aligned}$$

where $q[k] = \overline{x}[k] - x[k] - z[k]$ is the quantization noise and $w[k] = z[k] - 2z[k-1] + z[k-2]$ are identically distributed with mean 0. Let $q_{\max} \triangleq \max_k |q[k]|$ and $\left|w[2]\right|, \ldots, \left|w[n-1]\right|$ be sub-Gaussian with variance parameter σ^2 and common mean $\mu = \mathbb{E}\left[\left|w[2]\right|\right] > 0$. Then, for

$$W_n \triangleq \max_{k \in \{2, \ldots, n-1\}} \left|w[k]\right|, \tag{1}$$

we have, for a constant $c > 0$ ($cf.$ [34, Theorem 1.14]),

$$\mathbb{P}\left(W_n > \mu + c\right) \leq \exp\left\{-\frac{c^2}{2\sigma^2} + \log(n-2)\right\}. \tag{2}$$

Under $\mathcal{H}_0$, for $k = 2, \ldots, n-1$, we have

$$\begin{aligned} y[k] &= x[k] - 2x[k-1] + x[k-2] + w[k] \\ &\quad + q[k] - 2q[k-1] + q[k-2] \\ &= -4\sum_{l=1}^{L} \sin^2\left(\frac{\pi f_l}{f_s}\right) \sin\left(\omega_l\left(k-1\right)t_s + \theta_l\right) + w[k] \\ &\quad + q[k] - 2q[k-1] + q[k-2]. \end{aligned} \tag{3}$$

Similarly, under $\mathcal{H}_1$, for $k = 2, \ldots, n-1$, we have

$$\begin{aligned} y[k] &= -4\sum_{h=1}^{H} \sin^2\left(\frac{h\pi\widetilde{f}}{f_s}\right) \sin\left(h\widetilde{\omega}\left(k-1\right)t_s + \widetilde{\theta}_h\right) \\ &\quad - 4\sum_{l=1}^{L} \sin^2\left(\frac{\pi f_l}{f_s}\right) \sin\left(\omega_l\left(k-1\right)t_s + \theta_l\right) + w[k] \\ &\quad + q[k] - 2q[k-1] + q[k-2]. \end{aligned} \tag{4}$$

Define

$$S_n \triangleq \max_{k \in \{2, \ldots, n-1\}} \left|y[k]\right|. \tag{5}$$

[3]This is a more general assumption on the noise than Gaussian; the moment generating function (MGF) of a sub-Gaussian random variable with variance parameter σ^2 is upper bounded by the MGF of a Gaussian random variable with variance σ^2. See [34] for more details.

The test entails choosing a $\tau > 0$ and declaring $\mathcal{H}_0$ whenever $S_n \leq \tau$ and $\mathcal{H}_1$ otherwise. Note that this is equivalent to the anomaly detection procedure described in the previous section for the following reason. An r-bit ADC yields values quantized to 2^r levels. Therefore, the number of bits per sample used by the bit packing procedure (Algorithm 3) for a block of size $n-2$ with $S_n = s \in (0,1]$ is

$$\ell = \log_2(s2^r) = r - \log_2\left(\frac{1}{s}\right).$$

We set $\ell = 0$ if $s = 0$. Thus, ℓ is a monotonically increasing function of the statistic $s \in [0,1]$ and r is a constant. Hence, it suffices to find high probability upper and lower bounds for S_n under $\mathcal{H}_0$ and $\mathcal{H}_1$, respectively. For the latter, we need the following lemma. We adapt the characterization of the peak value of a signal in terms of the maximum absolute value of the samples collected via oversampling due to Ehlich and Zeller [35], to our setting.

Lemma 1. *Consider a signal $x_A(\cdot)$ which comprises H harmonics of frequency $\widetilde{f}$ Hz such that $f_s > 2H\widetilde{f}$. Let $\widetilde{\omega} = 2\pi\widetilde{f}$. Then,*

$$x_A(t) = a_0 + \sum_{h=1}^{H} a_h \sin(h\widetilde{\omega}t) + b_h \cos(h\widetilde{\omega}t),$$

where a_0, a_h, and b_h are real-valued, for $h \in [H]$. Let $x[\cdot]$ be the discrete sequence obtained by sampling $x_A(\cdot)$ at frequency f_s Hz. Then, for any $M \geq f_s/\widetilde{f}$,

$$\max_{k\in\mathcal{S}(M)} \left|x[k]\right| \geq \left(a_0^2 + \frac{1}{2}\sum_{h=1}^{H} a_h^2 + b_h^2\right)^{1/2} \cos\left(\frac{\pi H\widetilde{f}}{f_s}\right).$$

where $\mathcal{S}(M)$ denotes the set $\{l, l+1, \ldots, l+M-1\}$ for any non negative integer l.

Proof. Let $a > 1$ denote the oversampling factor for the given signal and sampling frequency; for a signal with maximum frequency $H\widetilde{f}$ Hz and sampling frequency f_s Hz, $a = f_s/2H\widetilde{f}$. Then, by [35, Theorem 1] we have

$$\begin{aligned}\|x_A\|_\infty &\leq \frac{1}{\cos\left(\frac{\pi}{2a}\right)} \max_{k\in\mathcal{S}(\lfloor 2aH\rfloor)} \left|x[k]\right| \\ &= \frac{1}{\cos\left(\frac{\pi H\widetilde{f}}{f_s}\right)} \max_{k\in\mathcal{S}(\lfloor f_s/\widetilde{f}\rfloor)} \left|x[k]\right|.\end{aligned}$$

Note that $H\widetilde{f}/f_s < 1/2$ and $\cos(\pi H\widetilde{f}/f_s)$ is positive in $[0, \pi/2)$. That is, if $M \geq f_s/\widetilde{f}$, then

$$\max_{k\in\mathcal{S}(M)} \left|x[k]\right| \geq \|x_A\|_\infty \cos\left(\frac{\pi H\widetilde{f}}{f_s}\right). \tag{6}$$

The fundamental period of $x_A(\cdot)$ is $1/\widetilde{f}$. By Plancherel's relation which yields

$$\|x_A\|_2^2 = \widetilde{f}\int_{1/\widetilde{f}} |x_A(t)|^2 dt = a_0^2 + \frac{1}{2}\sum_{h=1}^{H} a_h^2 + b_h^2,$$

and the observation, $\|x_A\|_2 \leq \|x_A\|_\infty$, we get

$$\|x_A\|_\infty \geq \left(a_0^2 + \frac{1}{2}\sum_{h=1}^{H} a_h^2 + b_h^2\right)^{1/2},$$

which along with Equation (6) completes the proof. □

Now we state the separation result for the statistic S_n in (5) under either hypotheses. In particular, we identify the condition under which it is possible to choose a value τ such that a test T that declares $\mathcal{H}_0$ if $S_n \leq \tau$ and $\mathcal{H}_1$ otherwise, constitutes a (δ, ϵ) test.

Theorem 2. *Suppose $n - 2 \geq f_s/\widetilde{f}$. Let T be the test that declares $\mathcal{H}_0$ if $S_n \leq \tau$ and $\mathcal{H}_1$ otherwise, for S_n as in* (5)*. Then, there exists a τ such that T constitutes a (δ, ϵ) test that resolves between $\mathcal{H}_0$ and $\mathcal{H}_1$ provided*

$$\begin{aligned}&8L\sin^2\left(\frac{\pi f_a}{f_s}\right) + \sigma\sqrt{2\log\frac{n-2}{\delta}} + \sigma\sqrt{2\log\frac{n-2}{\epsilon}} \\ &+ 8q_{\max} + 2\mu < 4\sqrt{\frac{H}{2}}\cos\left(\frac{\pi H\widetilde{f}}{f_s}\right)\sin^2\left(\frac{\pi\widetilde{f}}{f_s}\right),\end{aligned}$$

where $q_{\max} \triangleq \max_k |q[k]|$ and $\mu = \mathbb{E}\left[\left|w[2]\right|\right]$.

Proof. T declares $\mathcal{H}_1$ when $S_n > \tau$. Using triangle inequality, from (1), (3), and (5), under $\mathcal{H}_0$, we have

$$S_n \leq 4L\sin^2\left(\frac{\pi f_a}{f_s}\right) + W_n + 4q_{\max},$$

and hence

$$\mathbb{P}_{\mathcal{H}_0}\left[T \text{ declares } \mathcal{H}_1\right] \leq \mathbb{P}_{\mathcal{H}_0}\left[W_n > \mu + c\right],$$

where $c = \tau - 4L\sin^2\left(\frac{\pi f_a}{f_s}\right) - 4q_{\max} - \mu$. Therefore, to ensure that $\mathbb{P}_{\mathcal{H}_0}\left[T \text{ declares } \mathcal{H}_1\right] < \delta$, from (2), it suffices to have

$$\exp\left\{-\frac{c^2}{2\sigma^2} + \log(n-2)\right\} \leq \delta,$$

which is satisfied when

$$\tau \geq \sigma\sqrt{2\log\frac{n-2}{\delta}} + 4L\sin^2\left(\frac{\pi f_a}{f_s}\right) + 4q_{\max} + \mu. \tag{7}$$

Under $\mathcal{H}_1$, from (4) and (5), we have

$$\begin{aligned}\left|y[k]\right| \geq \Bigg|&4\sum_{h=1}^{H}\sin^2\left(\frac{h\pi\widetilde{f}}{f_s}\right)\sin\left(h\widetilde{\omega}\,(k-1)\,t_s + \widetilde{\theta}_h\right) \\ &+ 4\sum_{l=1}^{L}\sin^2\left(\frac{\pi f_l}{f_s}\right)\sin\left(\omega_l\,(k-1)\,t_s + \theta_l\right) - w[k] \\ &- q[k] + 2q[k-1] - q[k-2]\Bigg|.\end{aligned} \tag{8}$$

Invoking Lemma 1 with $M = n-2$, we have,

$$\begin{aligned}&\max_{k\in\mathcal{S}(n-2)}\left|\sum_{h=1}^{H}\sin^2\left(\frac{h\pi\widetilde{f}}{f_s}\right)\sin\left(h\widetilde{\omega}\,(k-1)\,t_s + \widetilde{\theta}_h\right)\right| \\ &\geq \sqrt{\frac{H}{2}}\sin^2\left(\frac{\pi\widetilde{f}}{f_s}\right)\cos\left(\frac{\pi H\widetilde{f}}{f_s}\right).\end{aligned} \tag{9}$$

By triangle inequality, (8), and (9), we have $S_n > \tau$ whenever

$$4\sqrt{\frac{H}{2}}\sin^2\left(\frac{\pi\widetilde{f}}{f_s}\right)\cos\left(\frac{\pi H\widetilde{f}}{f_s}\right) - 4L\sin^2\left(\frac{\pi f_a}{f_s}\right) - W_n - 4q_{\max} > \tau.$$

Thus, letting $A = \sqrt{\frac{H}{2}}\sin^2\left(\frac{\pi\widetilde{f}}{f_s}\right)\cos\left(\frac{\pi H\widetilde{f}}{f_s}\right)$, we have

$$\begin{aligned}\mathbb{P}_{\mathcal{H}_1}\left[T \text{ declares } \mathcal{H}_1\right] &= \mathbb{P}_{\mathcal{H}_1}\left[S_n > \tau\right] \\ &\geq \mathbb{P}_{\mathcal{H}_1}\left[W_n < \mu + c\right],\end{aligned}$$

where $c = 4A - 4L\sin^2\left(\frac{\pi f_a}{f_s}\right) - \tau - 4q_{\max} - \mu$. To ensure that $\mathbb{P}_{\mathcal{H}_1}\left[T \text{ declares } \mathcal{H}_1\right] \geq 1-\epsilon$, from (2), it suffices to have

$$\exp\left\{-\frac{c^2}{2\sigma^2} + \log(n-2)\right\} \leq \epsilon,$$

which is satisfied when

$$\tau \leq 4A - 4L\sin^2\left(\frac{\pi f_a}{f_s}\right) - \sigma\sqrt{2\log\frac{n-2}{\epsilon}} - 4q_{\max} - \mu. \tag{10}$$

From (7) and (10), T constitutes a (δ,ϵ) test if

$$\begin{aligned}&\sigma\sqrt{2\log\frac{n-2}{\delta}} + 4L\sin^2\left(\frac{\pi f_a}{f_s}\right) + 4q_{\max} + \mu \\ &< 4A - 4L\sin^2\left(\frac{\pi f_a}{f_s}\right) - \sigma\sqrt{2\log\frac{n-2}{\epsilon}} - 4q_{\max} - \mu,\end{aligned}$$

which completes the proof. □

We provide a numerical example that illustrates the usefulness of Theorem 2. For simplicity, we consider the noiseless case. Suppose there are $L = 10$ low frequency components with frequency less than or equal to $f_a = 50$ Hz. For sampling frequency $f_s = 32000$ Hz and blocklength $n = 16$, the requirement $n - 2 \geq f_s/\widetilde{f}$ implies that we can detect an anomaly of frequency $\widetilde{f} \geq 2286$ Hz and its higher harmonics. We identify the largest H, the number of harmonics of $\widetilde{f} = 2286$ Hz for which the test works. Clearly, $H\widetilde{f} < f_s/2$, which yields $H \leq 6$. The condition

$$8L\sin^2\left(\frac{\pi f_a}{f_s}\right) < 4\sqrt{\frac{H}{2}}\cos\left(\frac{\pi H\widetilde{f}}{f_s}\right)\sin^2\left(\frac{\pi\widetilde{f}}{f_s}\right)$$

in Theorem 2 is satisfied for all $H \leq 6$. Thus, all "detectable" harmonics of $\widetilde{f} = 2286$ Hz for the given sampling frequency $f_s = 32000$ Hz are detected by the test. As $\widetilde{f}$ increases, the number of harmonics allowed, H reduces. However, all anomalies of frequency $\widetilde{f} = 2286$ Hz or above can be detected using an appropriately chosen τ for some value of H ranging between 1 and $\lfloor f_s/2\widetilde{f}\rfloor$, the largest possible H constrained by the condition in Theorem 2.

Thus, we have established theoretically that the statistic S_n (and hence the number of bits per sample incurred by bit packing, ℓ which is used by the anomaly detection block) is able to separate the two hypotheses, following which the compression algorithm can switch between lossy and lossless compression. While a threshold-based test like the one proposed is standard, the remarkable fact is that the number of samples used to arrive at the decision, namely n, could be small, and hence the decision is quick. Using a larger n for a fixed sampling frequency f_s Hz, as well as using a smaller f_s for a fixed n can help resolve between hypotheses in which the low and high frequencies are closer.

Recall that in the filtering step, we use a two-level filter and in the dequeue step, we perform another test involving only the first samples of m consecutive blocks if the first test declares no anomaly in any of the m blocks. This is, in effect, a second hypothesis test of the same nature wherein if the first test declares that there are no high frequencies present, then we discard a few samples (resulting in downsampling) and perform the same test on samples effectively collected at a lower sampling frequency. Therefore, here the high frequency component under $\mathcal{H}_1$ is closer to the low frequencies. The efficacy of this test follows from the theorem we have shown above, applied with different hypotheses. Thus, the two-level hypothesis test enables the hierarchical subsampling, namely, downsampling the data first by n, and then by m if needed.

V. Experimental Results

A. Experiment on real data

The data is obtained from an IED connected to measure our laboratory's input power supply (see [8] for more details on the IED set up). The sampling frequency was set at 32000 Hz. The measured dataset comprises 5039 files of 1 second duration for 3 voltage channels, V_A, V_B, and V_C, and 2 current channels, I_C and I_N (the remaining two current channels are not connected to the load as the lab input is a single phase supply). The maximum, minimum, and average values of CR and NMSE were calculated for all the 5039 files by varying the thresholds, τ_H and τ_B. The results presented in this subsection have been in part, reported in [8] wherein the hardware platform enabling smart grid measurements is discussed, with a brief mention of the applications, the compression algorithm being one of them. Here, we place these results in the context of the theoretical analysis provided in the previous section.

Table I provides CR and NMSE for two cases corresponding to a medium and a high compression with $n = 16$, $m = 4$, and $\texttt{BUF} = 40$. Recall that n is the blocklength, m is the number of blocks whose first samples are used in the second-level filtering, and BUF is the number of blocks retained before and after a block that is declared anomalous. Note that $\texttt{BUF} = 40$ corresponds to one cycle of data for a fundamental frequency of 50 Hz sampled at $f_s = 32000$ Hz and blocklength $n = 16$. For the medium compression case, we set $\tau_H = 11$ for both voltage and current channels and $\tau_B = 7$ for the voltage channels and $\tau_B = 6$ for the current channels. Under the high compression setting, we set $\tau_H = 15$ and $\tau_B = 14$ for all the channels. In view of the theoretical analysis (and the numerical example that follows Theorem 2), for $f_s = 32000$, using just $n = 16$ samples, the algorithm can detect the presence of an anomaly comprising frequency 2286 Hz (or higher frequencies) and its higher harmonics for an appropriate choice of thresholds. While the theory is able to guarantee detection of H harmonics of a fixed frequency $\widetilde{f}$, in practice, the algorithm detects all deviations of sufficiently high frequency.

TABLE I: Maximum, minimum, and average CR and NMSE for ACS, with two different settings of τ_H and τ_B, and $n = 16$, $m = 4$, and BUF $= 40$ [8]

Ch.	CR			NMSE		
	max.	min.	avg.	max.	min.	avg.
Medium Compression						
V_A	16.20	05.58	14.53	0.006222	0.000099	0.000438
V_B	15.70	05.64	14.14	0.004215	0.000083	0.000403
V_C	15.64	05.73	14.05	0.005118	0.000145	0.000485
I_C	15.19	03.13	11.50	0.031000	0.003036	0.011898
I_N	15.17	05.03	13.16	0.030456	0.003435	0.011852
High Compression						
V_A	47.20	17.10	45.75	0.011313	0.002498	0.003303
V_B	46.18	14.33	43.49	0.011204	0.002266	0.003008
V_C	45.52	13.76	42.62	0.014279	0.002453	0.003263
I_C	45.51	15.25	44.25	0.077090	0.012552	0.040798
I_N	45.58	15.29	44.25	0.078481	0.012431	0.038999

It was observed that an average CR of around 1.42 for current channels and 1.64 for voltage channels can be obtained if all data has to be retained losslessly. From Table I, it can be seen that an average CR of upto around 45 can be configured for waveform recording based on the accuracy requirements for the specified values of n, m, and BUF. A snapshot of voltage and current channels corresponding to the high compression case is provided in Fig. 3. Note that the real values used in plotting are obtained by multiplying the integer values (output by the ADC) with the gain associated with the ADC channel. In Fig. 4, a heat map of block wise CR for a snapshot of voltage and current channel respectively for the medium compression setting in Table I is provided. It may be noted that the algorithm achieves better compression in regions where the signal is "smooth". In fact, by adjusting the parameter values (τ_H and τ_B), we can obtain reconstruction of different accuracies. For instance, the reconstruction in Fig. 4 is "more accurate" compared to that in Fig. 3. This is because, in Fig. 4, the setting is that of medium compression (see Table I). Thus, it is possible to get a whole range of reconstruction accuracies (and compression ratios) by adjusting the parameters appropriately; a specific setting may be chosen depending on the application requirement. This is a unique feature of ACS that enables it to cater to different applications.

B. Comparison with an existing scheme

The scheme proposed in [11], named G-PQDR was implemented and executed on the same data. The scheme is based on detection of *novelty frames*, which entails comparing the energy of the signal in the high frequency spectrum with that of a reference signal and also the difference in the magnitudes of the fundamental frequency using a fixed threshold. Compression is achieved mainly by discarding frames "similar" to the reference frames and storing only the fundamental frequency information for those frames. For the novelty frames, a wavelet transform is calculated and

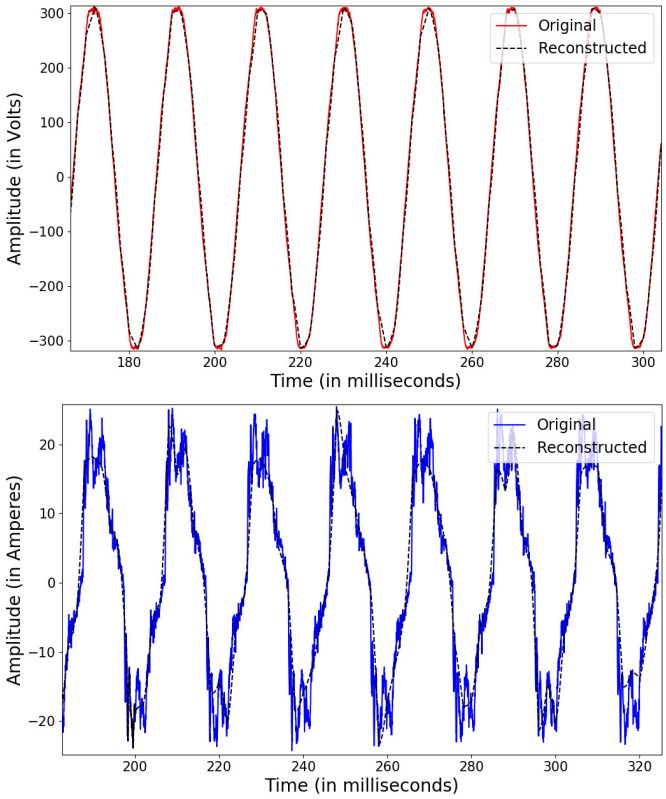


Fig. 3: Original signal and reconstruction using ACS for voltage channel V_A (top) and current channel I_C (bottom) for the high compression setting in Table I [8]

coefficients less than a threshold are discarded (set to zero). The fundamental frequencies of the cycles in the non-novelty frame is calculated using a Phase Locked Loop (PLL) based method. These values are used to modulate the harmonics of the reference frame calculated using a Sliding Window Recursive Discrete Fourier Transform. However, the examples given in [11] are transmission systems and the problem at hand is that of compression of signals in a distribution system. There is higher uncertainty in the estimated instantaneous frequency in the latter case and hence for a chosen threshold, either almost all frames are marked as novelty (resulting in a low CR) or all frames are marked as non-novelty (resulting in a high NMSE due to poor reconstruction resulting from frequency estimation errors). This behavior is more prominent for current channels than voltage channels.

The bit mapping proposed in G-PQDR is a fixed length code which is not very useful when the wavelet coefficients are set to 0 based on a threshold. In our implementation of G-PQDR, we replace this with an Elias code which is a better variable length code. To avoid the first frame of every file being treated as a novelty (thereby causing a low CR), the 5039 files were passed to the algorithm as a stream (and hence only average CR is available). For this implementation, for different values of thresholds, CR and NMSE were obtained as shown in Table II. A snapshot of the reconstructed signal for V_A and I_C is shown in Fig. 5. It can be seen that performance of G-PQDR is affected heavily by the frequency variations compared to its

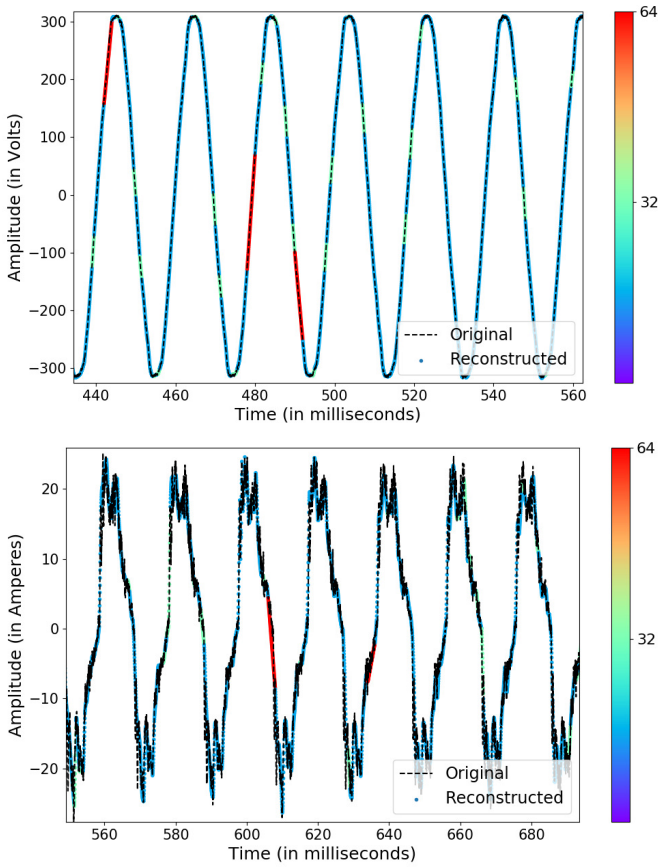


Fig. 4: Heat map of block wise CR for ACS for V_A (top) and I_C (bottom) for the medium compression setting in Table I [8]

performance on transmission system data as exhibited in [11]. Comparing the values in Table II with those in Table I, it is clear that ACS offers a superior performance for the distribution system data. For instance, for the current channel I_N, for CR ≈ 15, G-PQDR incurs an NMSE ≈ 0.27 while for ACS, the NMSE is approximately 0.07. For the voltage channel V_A, for an NMSE ≈ 0.006, G-PQDR yields a CR ≈ 2.5 whereas even the minimum CR is roughly 5.5 for ACS. Besides, it is noteworthy that ACS achieves this performance at blocklengths as small as $n = 16$, while compression by G-PQDR entails accumulating 4 cycles worth of data, equal to 2560 samples. The merit of our algorithm is that it is oblivious to the varying fundamental frequency and sampling frequency, and identifies *local* anomalies using small blocklength of data to switch between lossy and lossless compression in this scale adaptively.

C. Event detection

The efficacy of ACS in identifying events was tested against data in DOE/EPRI National Database Repository of Power System Events [26]. For data with a fixed sampling frequency, the value of $\ell^{(i)}$ calculated in Algorithm 3 is compared against a fixed threshold τ_H. Recall that the detector uses this as a test for events. Fig. 6 shows the voltage and current signals sampled at $f_s \approx 7679$ Hz; the blocks where $\ell^{(i)} > \tau_H$ are labeled using circles. A blocklength of $n = 16$ was used and

TABLE II: CR and NMSE for G-PQDR for various thresholds for novelty detection for voltage (top) and current (bottom) channels

V_A		V_B		V_C	
CR	NMSE	CR	NMSE	CR	NMSE
2.53	0.006648	2.53	0.006095	2.53	0.006655
2.62	0.011207	2.61	0.010335	2.61	0.010481
2.73	0.017596	2.70	0.014677	2.71	0.014314
3.08	0.032191	3.02	0.027255	3.03	0.026784
4.22	0.056809	4.01	0.050305	4.03	0.047745
7.15	0.100265	6.71	0.089983	6.69	0.088740

I_C		I_N	
CR	NMSE	CR	NMSE
02.58	0.060520	02.58	0.059435
03.17	0.091964	03.16	0.090285
03.97	0.125218	03.93	0.120505
07.52	0.200162	07.40	0.196725
14.92	0.269896	15.38	0.277052
24.93	0.345035	26.91	0.354505

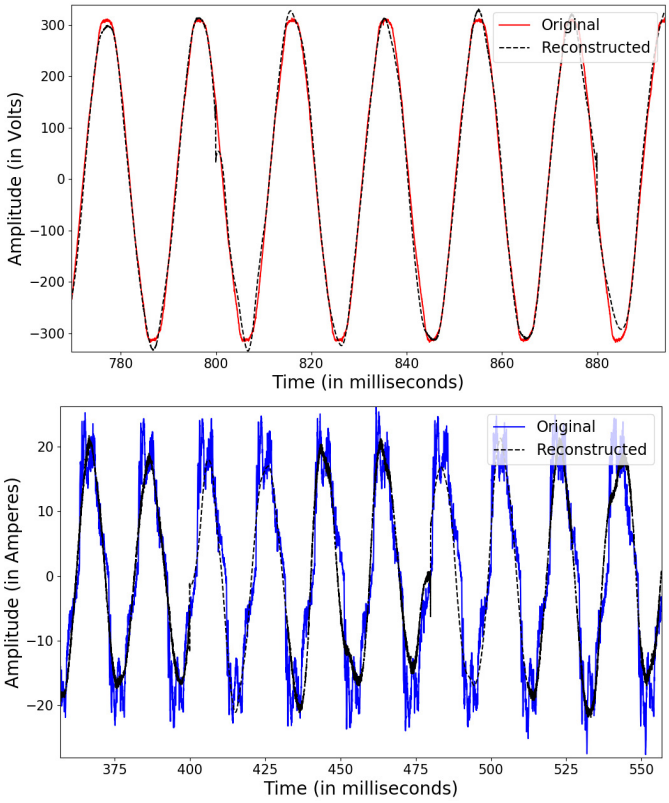


Fig. 5: Original signal and reconstruction using G-PQDR for voltage V_A (top) and current I_C (bottom) channels for the compression setting in the first rows of the tables in Table II

τ_H was set to 10 for this sampling frequency. Similar results were obtained for other event files as well and the figures are made available online at [36]. It can be seen that the algorithm detects most of the events and thus, will switch to appropriate compression when needed. Note that the algorithm is configured to store `BUF` samples before and after an event is detected losslessly; here, we only highlight the blocks in which the detector is triggered.

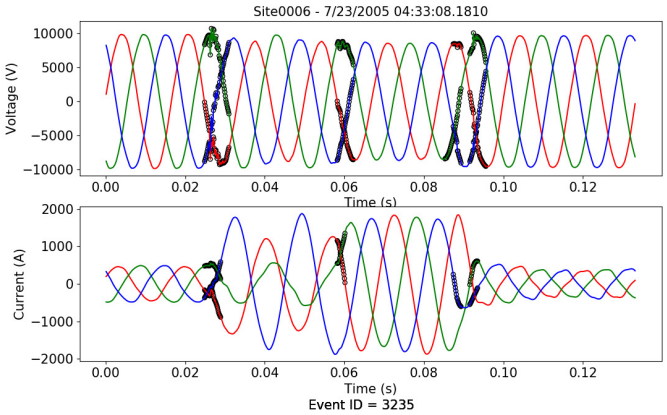


Fig. 6: Event with ID 3235 ($f_s \approx 7679$) in the DOE/EPRI disturbance library [26] – circles mark the blocks i for which $\ell^{(i)} > \tau_H$; that is, the blocks where ACS detects events.

As noted in Section III-A5, a special character (byte) is included as prefix to the compressed stream whenever the algorithm switches between lossy and lossless compression. In our implementation, we have used the byte `11011111` to denote the start of lossy compression, `11101111` to denote the start of compression dropping a single packet, and `11110111` to denote the start of compression dropping m consecutive packets. None of these bit sequences occur as the first byte of the compressed stream of a block since the first sample is stored using a variable byte scheme (see Section III-A2) and these sequences do not occur when the ADC resolution is $r = 16$. Thus, by parsing the compressed stream it is possible to determine if an event (as characterized by Theorem 2) occurred or not without processing the compressed data further. It may be noted that in the presence of an anomaly, the data is stored losslessly and hence could be reconstructed as such.

G-PQDR also detects these events. In order to compare the CR and NMSE of the two schemes, a reasonable number of cycles of data is necessary and we have executed both G-PQDR and ACS on a subset of the available data (30 files with roughly 120 cycles each) with $f_s \approx 15385$ with parameters configured to detect the events. For ACS, we have used $n = 16$, $m = 4$, and $\texttt{BUF} = 16$ so that $n \cdot \texttt{BUF}$ equals roughly one cycle of data. CR and NMSE were obtained as shown in Table III. The performance of both the algorithms are comparable in the sense that the one with lower CR has lower NMSE as well. Although the data is from a transmission system, in this case, the reconstruction by G-PQDR suffers from a high NMSE due to the fact that the fundamental frequency is 60 Hz and $f_s = 15384.6153845672$ (this corresponds to a sampling period of 65 microseconds; see [26]), whereby the number of samples per cycle is not an integer. In fact, this is a limitation of using G-PQDR in practical systems which could have sampling frequencies not necessarily integer multiples of the fundamental.

TABLE III: CR and NMSE for ACS and G-PQDR for 30 files with $f_s \approx 15385$ in the DOE/EPRI disturbance library [26]; the parameters were set to detect most of the events.

Ch.	ACS		G-PQDR	
	CR	NMSE	CR	NMSE
I_A	4.79	0.000000	16.01	0.046433
I_B	4.75	0.000001	14.52	0.030888
I_C	4.75	0.000000	11.56	0.027281
I_N	4.75	0.000007	12.67	0.150681
V_A	45.28	0.045208	33.34	0.009346
V_B	44.40	0.04544	36.93	0.010131
V_C	40.87	0.04197	31.13	0.010780

VI. Concluding Remarks

In this work, we propose a hierarchical anomaly-aware adaptive compression algorithm that outperforms the state-of-the-art compression algorithm that relies on real-time frequency estimation and requires accumulating a few cycles of data. Distribution system data exhibits higher variation in fundamental frequency and hence techniques that rely on instantaneous frequency estimation are unsuitable. Our scheme operates on short blocks of streaming data and uses the same statistic to solve a hypothesis testing problem to decide if an anomaly is present in a particular block, and to compress the block subsequently. We establish theoretically that the test identifies the presence of high frequencies even for short blocklengths, the separation between the low and high frequencies dependent on the blocklength. The efficacy of the algorithm in detecting events has been validated on publicly available DOE/EPRI data as well. The algorithm is suitable for compressing nonstationary signals sampled at high frequency when higher reconstruction accuracy is prescribed only around rarely-occurring anomalies. Besides, it employs only integer operations and is well-suited for streaming implementation on low-compute hardware. The proposed anomaly-aware adaptivity in sampling frequency as a means to compressing nonstationary signals is a new paradigm, which may be implemented with different constituent blocks corresponding to filtering, bit packing, and anomaly detection (possibly with heavier computation if applicable) as well.

Acknowledgements

The authors thank Prof. Rajesh Sundaresan, IISc for useful discussions. Thanks are also due to the reviewers and the editor whose comments improved the presentation.

References

[1] T. Designs. (2016) AFE using high-speed ADC and differential amplifier for digital fault recorder or TWFL. [Online]. Available: http://www.ti.com/lit/ug/tiduat7a/tiduat7a.pdf

[2] Eaton. Next-generation power quality meters. [Online]. Available: http://www.eaton.com/ecm/groups/public/pub/electrical/documents/content/1138850333755.pdf

[3] J. Ning, J. Wang, W. Gao, and C. Liu, "A wavelet-based data compression technique for smart grid," *IEEE Trans. on Smart Grid*, vol. 2, no. 1, pp. 212–218, 2011.

[4] H. Jiang, J. J. Zhang, and D. W. Gao, "Fault localization in smart grid using wavelet analysis and unsupervised learning," in *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Nov 2012, pp. 386–390.

[5] T. Pelkonen, S. Franklin, J. Teller, P. Cavallaro, Q. Huang, J. Meza, and K. Veeraraghavan, "Gorilla: A fast, scalable, in-memory time series database," *Proceedings of the VLDB Endowment*, vol. 8, no. 12, pp. 1816–1827, 2015.

[6] D. Blalock, S. Madden, and J. Guttag, "Sprintz: Time series compression for the internet of things," *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, p. 93, 2018.

[7] K. R. Sahasranand, B. Rout, A. Joglekar, G. Gurrala, and H. Tyagi, "Fault-aware compression for high sampling rate data acquisition in smart grids," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, ser. e-Energy '19. ACM, 2019, pp. 422–424.

[8] A. Joglekar, G. Gurrala, P. Kumar, F. C. Joseph, T. Kiran, K. R. Sahasranand, and H. Tyagi, "Open-source heterogeneous constrained edge-computing platform for smart grid measurements," *IEEE Trans. on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[9] S. Werner and J. Lundén, "Smart load tracking and reporting for real-time metering in electric power grids," *IEEE Trans. on Smart Grid*, vol. 7, no. 3, pp. 1723–1731, 2016.

[10] J. C. S. de Souza, T. M. L. Assis, and B. C. Pal, "Data compression in smart distribution systems via singular value decomposition," *IEEE Trans. on Smart Grid*, vol. 8, no. 1, pp. 275–284, 2017.

[11] L. Silva, E. Kapisch, C. Martins, L. Filho, A. Cerqueira, C. Duque, and P. Ribeiro, "Gapless power-quality disturbance recorder," *IEEE Trans. on Power Delivery*, vol. 32, no. 2, pp. 862–871, 2017.

[12] F. Zhang, L. Cheng, X. Li, Y. Sun, W. Gao, and W. Zhao, "Application of a real-time data compression and adapted protocol technique for WAMS," *IEEE Trans. Power Systems*, vol. 30, no. 2, pp. 653–662, 2015.

[13] P. H. Gadde, M. Biswal, S. Brahma, and H. Cao, "Efficient compression of PMU data in WAMS," *IEEE Trans. on Smart Grid*, vol. 7, no. 5, pp. 2406–2413, 2016.

[14] L. S. Pinto, M. V. Assunção, D. A. Ribeiro, D. D. Ferreira, B. N. Huallpa, L. R. Silva, and C. A. Duque, "Compression method of power quality disturbances based on Independent Component Analysis and Fast Fourier Transform," *Electric Power Systems Research*, vol. 187, p. 106428, 2020.

[15] V. Loia, S. Tomasiello, and A. Vaccaro, "Fuzzy transform based compression of electric signal waveforms for smart grids," *IEEE Trans. on Systems, Man, and Cybernetics: Sys.*, vol. 47, no. 1, pp. 121–132, 2017.

[16] E. Kapisch, L. Silva, C. Martins, A. Barbosa, C. Duque, A. Tavil, L. de Souza *et al.*, "An implementation of a power system smart waveform recorder using FPGA and ARM cores," *Measurement*, vol. 90, pp. 372–381, 2016.

[17] Ö. N. Gerek and D. G. Ece, "Compression of power quality event data using 2D representation," *Electric Power Systems Research*, vol. 78, no. 6, pp. 1047–1052, 2008.

[18] M. V. Ribeiro, S. H. Park, J. M. T. Romano, and S. K. Mitra, "A novel MDL-based compression method for power quality applications," *IEEE Trans. on Power Delivery*, vol. 22, no. 1, pp. 27–36, 2007.

[19] M. P. Tcheou, L. Lovisolo, M. V. Ribeiro, E. A. da Silva, M. A. Rodrigues, J. M. Romano, and P. S. Diniz, "The compression of electric signal waveforms for smart grids: State of the art and future trends," *IEEE Trans. on Smart Grid*, vol. 5, no. 1, pp. 291–302, 2014.

[20] E. O. Schweitzer III, M. V. Mynam, A. Guzman-Casillas, B. Z. Kasztenny, V. Skendzic, T. J. Lee, and D. E. Whitehead, "Fault detection in electric power delivery systems using underreach, directional, and traveling wave elements," Oct. 18 2016, US Patent 9,470,748.

[21] A. F. Bastos and S. Santoso, "Universal waveshape-based disturbance detection in power quality data using similarity metrics," *IEEE Trans. on Power Delivery*, vol. 35, no. 4, pp. 1779–1787, 2019.

[22] Y. Xiong, Y. Jing, and T. Chen, "Abnormality detection based on the Kullback–Leibler divergence for generalized Gaussian data," *Control Engineering Practice*, vol. 85, pp. 257–270, 2019.

[23] A. V. Oppenheim, J. R. Buck, and R. W. Schafer, *Discrete-time signal processing. Vol. 2*. Upper Saddle River, NJ: Prentice Hall, 2001.

[24] P. F. Ribeiro, C. A. Duque, P. M. Ribeiro, and A. S. Cerqueira, *Power systems signal processing for smart grids*. John Wiley & Sons, 2013.

[25] D. Lemire and L. Boytsov, "Decoding billions of integers per second through vectorization," *Software: Practice and Experience*, vol. 45, no. 1, pp. 1–29, 2015.

[26] E. P. R. I. Inc. DOE/EPRI National Database Repository of Power System Events. [Online]. Available: https://pqmon.epri.com/

[27] N. T. Thao and M. Vetterli, "Reduction of the MSE in R-times oversampled A/D conversion O $(1/R)$ to O $(1/R^2)$," *IEEE Trans. on Sig. Proc.*, vol. 42, no. 1, pp. 200–203, 1994.

[28] Z. Cvetkovic and M. Vetterli, "On simple oversampled A/D conversion in $L^2(\mathbb{R})$," *IEEE Trans. on Info. Theory*, vol. 47, no. 1, pp. 146–154, 2001.

[29] Z. Cvetkovic, I. Daubechies, and B. F. Logan, "Single-bit oversampled A/D conversion with exponential accuracy in the bit rate," *IEEE Trans. on Info. Theory*, vol. 53, no. 11, pp. 3979–3989, 2007.

[30] P. T. Boufounos, "Universal rate-efficient scalar quantization," *IEEE Trans. on Info. Theory*, vol. 58, no. 3, pp. 1861–1872, 2011.

[31] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer Science & Business Media, 2012, vol. 159.

[32] P. Elias, "Universal codeword sets and representations of the integers," *IEEE Trans. on Info. theory*, vol. 21, no. 2, pp. 194–203, 1975.

[33] Google, "Protocol buffers encoding," 2001. [Online]. Available: https://developers.google.com/protocol-buffers/docs/encoding\#types

[34] P. Rigollet and J.-C. Hütter, "High dimensional statistics," *Lecture notes for course 18S997*, 2015.

[35] G. Wunder and H. Boche, "Peak value estimation of bandlimited signals from their samples, noise enhancement, and a local characterization in the neighborhood of an extremum," *IEEE Trans. on Sig. Proc.*, vol. 51, no. 3, pp. 771–780, 2003.

[36] K.R.Sahasranand. Event detection on DOE/EPRI data. [Online]. Available: https://github.com/sanandkr/EPRIEventDetection

K. R. Sahasranand received the B.Tech degree in Computer Science and Engineering from Amrita University in 2009 and worked as a Scientist/Engineer at Indian Space Research Organisation, 2010-'11. He completed his Master's and PhD in Electrical Communication Engineering from Indian Institute of Science, Bangalore in 2015 and 2022, respectively. His research interests include information theory, detection and estimation, and signal processing.

Francis C Joseph received his B.E degree in Electrical and Electronics Engineering from BMSCE, Bangalore in 2009 and M.E degree in Electrical Engineering from Indian Institute of Science, Bangalore in 2011. He worked as a Senior Engineer in the R&D Department of PRDC Pvt Ltd. He is currently pursuing Ph.D. degree with the Department of Electrical Engineering, Indian Institute of Science, Bangalore, India. His research interests include power system stability, application of HPC in power systems and time domain simulations.

Himanshu Tyagi received the B.Tech. degree in Electrical Engineering and the M.Tech. degree in Communication and Information Technology from the Indian Institute of Technology, Delhi, India, in 2007, and the Ph.D. degree from the University of Maryland, College Park, MD, USA, in 2013. From 2013 to 2014, he was a Postdoctoral Researcher with the Information Theory and Applications (ITA) Center, University of California at San Diego, San Diego, CA, USA. Since January 2015, he has been a Faculty Member with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore. He received the Indian National Science Academy (INSA) Medal for Young Scientist 2020. His research interests include information theory and its application in cryptography, statistics, machine learning, and computer science. Also, he is interested in communication and automation for city-scale systems.

Gurunath Gurrala received the B.Tech. degree from S.V.H. College of Engineering, Machilipatnam, India, in 2001, the M.Tech. degree from J.N.T.U. College of Engineering, Anantapur, India, in 2003, and the Ph.D. degree from the Indian Institute of Science, Bangalore, India, in 2010. He was an Assistant Professor with the SSN College of Engineering, Ongole, India, during 2001 - 2002 and with Anil Neerukonda Institute of Technology and Sciences (ANITS), Visakhapatnam, India, during 2003-2005. He was a Research Engineer with GE Global Research, Bangalore, India, during 2010 - 2012. He was a Postdoctoral Fellow with Texas A&M University, USA, during 2012-2013 and at Oak Ridge National Lab, USA, during 2014-2015. He is currently an Associate Professor with the Department of Electrical Engineering, Indian Institute of Science. He received SERB-STAR AWARD 2020, IEEE PES Outstanding Engineer Award 2018, INAE Young Engineer Award 2015. His papers received best paper award in ICPS 2019, IEEE General meeting 2015, best poster awards in IEEE industrial society annual meeting 2015 and EEE PES T&D Conference and Exposition 2016. Three of his master students received POSOCO Power System Awards. He is senior member of IEEE and INAE young associate. His research interests include smart grids, power system stability, grid integration of renewables, microgrid control, high-performance computing applications to power systems, nonlinear, and adaptive control of power systems.

Ashish Joglekar received his PhD in EMI/EMC from the Department of Electronics Systems Engineering, Indian Institute of Science in the year 2015. Currently he is a visiting faculty at the Robert Bosch center for Cyber Physical systems IISc, and a Senior Member of Technical Staff in hardware design at the AI and Robots Technology Park IISc. His research interests include design of analog sensor front ends, design for Electromagnetic Compatibility, active EMI filters, Switched mode Power converter design, Motor control.