IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 August 2022, accepted 27 September 2022, date of publication 5 October 2022, date of current version 12 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3212401

RESEARCH ARTICLE

A Hierarchical Approach for Multiple Periodicity Detection in Software Code Analysis

MINE KERPICCI[1], (Graduate Student Member, IEEE), MILOS PRVULOVIC[2], (Senior Member, IEEE), AND ALENKA ZAJIĆ[1], (Senior Member, IEEE)

[1]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[2]School of Computer Science, Georgia Institute of Technology, Atlanta, GA 30332, USA

Corresponding author: Mine Kerpicci (mkerpicci3@gatech.edu)

This work was supported by ONR under Grant N00014-19-1-2287.

ABSTRACT This paper introduces an end-to-end processing method for multiple periodicity signal detection and analysis with particular application in software analysis using analog side channels. The probabilistic distributions of signal blocks are estimated with kernel density estimation. The corresponding kernel bandwidths, which are optimally found in a data-driven manner, are used to detect change points. After separating the signal into parts with different behaviors, average magnitude difference function is leveraged iteratively to find the smallest periodic signal sections. To illustrate efficiency of the proposed method, we use EM side-channel signals collected from real-life applications to successfully detect multiple existing periodicities.

INDEX TERMS Period detection, multiple periodicities, change point detection, kernel density estimation, average magnitude difference function, side-channel.

I. INTRODUCTION

Periodic signals are found in various forms in different research fields. There are many examples that include biomedical signals such as heartbeats, meteorological recordings of weather changes or financial time series data. In all these forms, the period information provides highly valuable insight to understand the nature of the signal. This makes the signal period detection a crucial step in various applications ranging from astronomy [1], biomedical [2], [3], communication [4], to analog side-channels [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], etc.

This paper was motivated by problems found in signal processing of electromagnetic side channels to profile software activities. Electromagnetic side channels emanate from computer system during program running. Hence, they contain information about the program activities run by the device. Period detection is highly crucial for such signals since it allows one to understand the program behavior for various purposes such as software analysis [17], [18], [19], [20] and hardware Trojan detection [21].

Signal periodicity in electromagnetic side channels can tell us the number of periodic activities (such as loops), structure (are they nested or not), how long are those loops and all questions essential for recognizing code structure by analyzing analog signals. However, these signals are challenging in terms of period detection since they commonly contain multiple nested periodicities due to repetitive nature of the program execution. In other words, the signal periodicity may not be represented with one dominant period in all cases. Instead, the signal may contain multiple interlaced or nested periods. Moreover, different periodicities can be observed in various regions of signals with changing behaviors in time. To the best of our knowledge, there are no signal processing techniques that address finding periodicities in electromagnetic side channels to analyze software activities. To address these issues, we study period detection problem for quasi-periodic signals with multiple nested periodicities.

The first observation from experimental data was that signals do not have periodic structure but more like

The associate editor coordinating the review of this manuscript and approving it for publication was Liangtian Wan.

quasi-periodic structure. The reason is that within loops different code might be executing if branching is present. That has motivated us to investigate how to detect all periodicities in unknown empirically collected quasi-periodic data. To accomplish this objective, one first needs to identify behavioral change points to extract signal intervals containing periodicities. Therefore, our method is twofold. We first detect the change points of the observed signal without any model assumption. Then, we iteratively find all existing periods, i.e., from the longest period to the smallest period, in a signal with multiple periods. To demonstrate the performance of the proposed technique, we use experimentally collected side-channel signals and show that we successfully detect all existing periods.

The main contribution of this paper is that we develop the first algorithm that is finding periodicities in electromagnetic side channels with application of code structure analysis. This algorithm addresses important problem in the periodicity analysis not addressed before: detecting nested periods in quasi-periodic signals. This has been accomplished in the following manner:

1) We propose an approach to find behavioral change points of a signal including quasi-periodic signals to detect different periodicities. Since we use a nonparametric density estimation method without any model assumption, this method works with data coming from any probabilistic distribution. With this, we handle the period detection problem for signals with changing behaviors in time.
2) We describe a hierarchical method for finding all periodicities in a given analog signal with low computational complexity. With the introduced approach, we address the corrupting effects such as signal interruptions or additional regions inserted between periodic activities, which disrupt perfect periodicity.
3) We demonstrate the effectiveness of the proposed method on the side-channel signals, where multiple periodicities are common occurrences due to repetitive nature of program execution. Even though these signals are not perfectly periodic and hence challenging to work with, we successfully detect all periodic patterns.

The organization of this paper is as follows. We review related work in Section II, define our problem setting in Section III, present our hierarchical approach for multiple period detection in Section IV, demonstrate the performance of our method on side-channel processor measurements in Section V, and finally, conclude the paper in Section VI.

II. RELATED WORK

To the best of our knowledge, there are no signal processing techniques that address finding periodicities in electromagnetic side channels that analyze software activities. Hence we looked at other research areas that had similar problems.

Most of the signals used in practice contain different behavioral regions such as time series data belonging to biomedical, finance, or meteorology fields. The detection of transition/change points between different states of such signals carries high importance in terms of data modeling, analysis and prediction. Therefore, change point detection methods find use in various signal processing applications [22], [23]. Change point detection problem is addressed in previous work in a supervised setting [24], [25], [26], [27], [28]. However, they require an additional training phase and hence large number of data instances and labels, which is not suitable for many applications. Another approach to change point detection is estimating the density of the signal to observe the behavioral changes. Work in [29] and [30] use parametric models for density estimation to detect change points. They assume that the data distribution is from exponential family. That is not suitable for many applications where data distribution is not a priori known. However, none of these data sets have problem of nested periods (such as nested loops), hence they are not directly applicable to our problem.

In this paper, we exploit change point detection to separate different behavioral regions in signals as in the case of quasi-periodic signals to be able to detect all existing periodicities. To address this problem, we use estimation of the signal density to observe the behavioral changes. In particular, our kernel density estimation based change point detection method does not assume any distribution and hence it can model any distribution in a data-driven manner. Please note that we apply change point detection step to extract signal intervals with different behaviors, i.e., periodicities.

Period detection problem is extensively studied in the literature. One approach applied in these studies is using frequency domain. For example, [31] uses periodogram obtained through Fourier transform where one can detect the dominant frequencies. Similarly, Fourier transform is exploited in [32] for periodicity analysis. However, periodogram does not work well for longer periods and does not guarantee providing accurate results due to spectral leakage [33]. To avoid this, several studies propose to solve period detection problem in time domain by using time domain related properties of a signal. Among these, autocorrelation function is applied in [34] to find the period. Since low amplitude repetitions may be interpreted as less important in autocorrelation function, [33] proposes to combine periodogram and autocorrelation findings for more accurate results. This problem is also investigated as pitch detection in speech processing applications. Reference [35] uses zero-crossing rate property of speech signals in addition to the autocorrelation function for pitch detection. Moreover, [36] combines autocorrelation and average magnitude difference function (AMDF) for pitch detection. Among these numerous techniques, average magnitude difference function (AMDF) has a wide range of applications due to its low computational time. Therefore, several studies based on AMDF are developed in the literature [37], [38]. However, most of these methods and others such as [39] do not address the multiple periodicity issues and focus on finding one dominant period in the signal. Reference [40] studies multiple periodicity detection problem in both time and frequency domain where one can observe

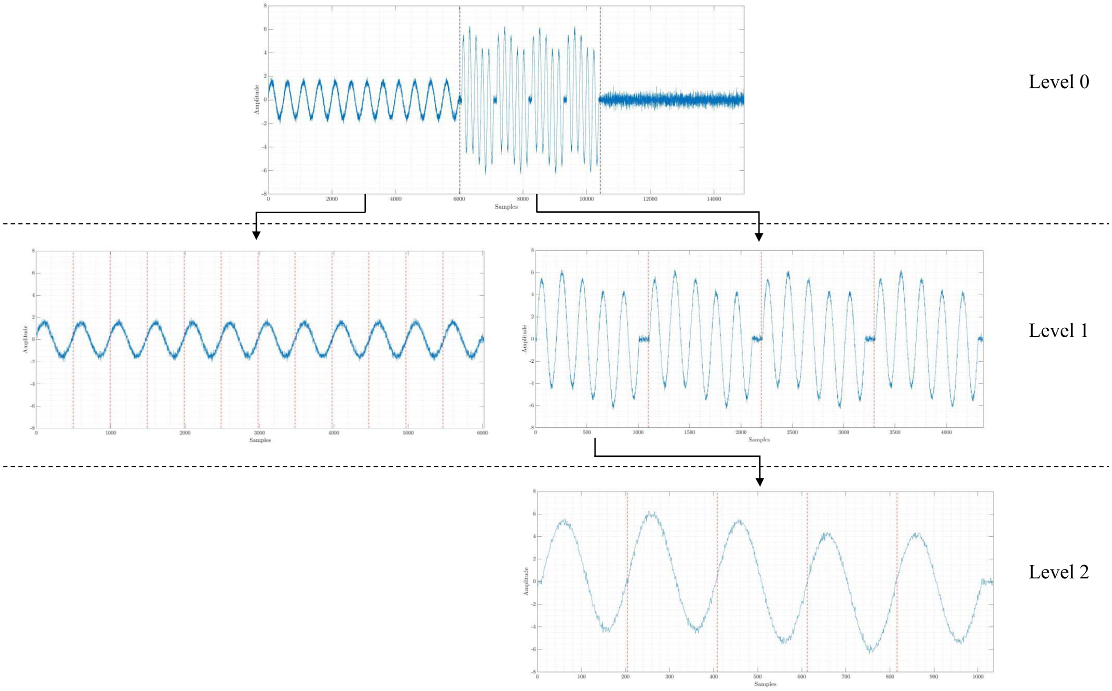


FIGURE 1. A hierarchical structure for multiple periodicity detection on an example signal with various behavioral regions.

multiple interlaced periods in addition to the dominant period. Reference [41] also analyzes multiple hidden periodicities where an observed signal is a combination of multiple signals with different periods. However, these methods do not consider quasi-periodic and almost-periodic signals where there is not perfect periodicity. In practice, the behavior of a signal may be changing in time resulting in different periodicities in different time intervals. Moreover, there may be interruptions or nonperiodic regions between periodic activities, which disrupt the perfect periodicity of the signal. As a result of such effects, the nested periods other than the dominant period become invisible from direct periodogram or autocorrelation function of the complete signal. In contrast, we introduce a hierarchical approach to find all periodicities in an observed signal where we also address these problems. In our framework, we iteratively find all existing periods where the computational complexity is only linear with the number of existing periods and the data length. Therefore, the introduced method is appropriate for several real-world applications.

III. PROBLEM DESCRIPTION

Signal periodicity in electromagnetic side channels is essential for understanding code behavior by observing analog signals. However, this is a challenging problem because these signals may contain multiple nested periodicities due to repetitive nature of the program execution. To address these issues, we study period detection problem for quasi-periodic signals with multiple nested periodicities and define the problem as described below.

We observe a sequence $\boldsymbol{x} \in \mathbb{R}^N$ of uniformly sampled quasi-periodic data instances. The observed signal $\boldsymbol{x} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_T\}$ consists of T number of patterns containing multiple periodicities where $\boldsymbol{y}_i \in \mathbb{R}^{n_i}$. Our aim is to extract the patterns in the observed sequence and detect all existing periodicities. For this, we introduce a two-step approach where we first find the data intervals showing similar behaviors based on the estimated densities. Then, we iteratively find multiple periodicities in the extracted data intervals.

In order to construct a signal representation, we divide the sequence into k blocks $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k\}$ with the same length $\boldsymbol{x}_i \in \mathbb{R}^h$ and use kernel density estimation (KDE) to obtain the distribution of each block as $f(\boldsymbol{x}_i)$. Since the signal blocks containing the same patterns have the same probability distribution, we group the blocks having similar distributions to find the patterns $\boldsymbol{y}_i = \{\boldsymbol{x}_m, \boldsymbol{x}_{m+1}, \ldots, \boldsymbol{x}_{m+t_i}\}$. This means that we search for consecutive blocks $\{\boldsymbol{x}_m, \boldsymbol{x}_{m+1}, \ldots, \boldsymbol{x}_{m+t_i}\}$ that have similar distributions, where changing from $\boldsymbol{x}_{m-1}$ to $\boldsymbol{x}_m$ and from $\boldsymbol{x}_{m+t_i}$ to $\boldsymbol{x}_{m+t_i+1}$ corresponds to the change points. For example, the synthetically constructed signal in Fig. 1 has three main regions with different behaviors. Similarly, the real signal example given in Fig. 2 consists of two main different behaviors, which are highlighted with matching frame colors. Here, there exists three change points resulting in dividing the signal into four groups.

We analyze the extracted patterns separately to find all existing periodicities. Let $\boldsymbol{y} = \{y_1, y_2, \ldots, y_n\}$ be a signal block containing multiple periodicities such that

$$y_i = y_{i+w_i},$$

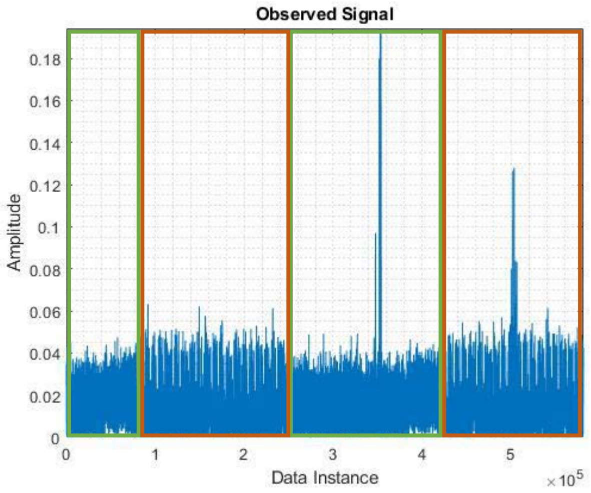


FIGURE 2. Analog side-channel signal collected from Raspberry Pi device while running ECC algorithm. Kernel density estimation based change point detection method finds two sections of the signal with similar behavior as indicated with the same color frames.

where $w_i \in \mathbb{R}$ can take one or multiple values. To find all existing periods, we introduce an iterative approach based on average magnitude difference function. At each iteration, we find the longest period and extract the periodic parts so that we continue to analyze all extracted parts separately. Hence, we eliminate the effects of possible nonperiodic regions or interruptions, which disrupt the perfect periodicity.

IV. PROPOSED METHOD

In this section, we describe our hierarchical approach to analyze an observed signal. In the first level of our hierarchical structure, we search for different behavioral regions existing in the signal. For this, we use kernel density estimation based change point detection technique and extract signal intervals with different behaviors. We provide details of this step in Section IV-A. Then, we process these regions separately and apply iterative period detection method based on average magnitude difference function to find all existing periodic regions. We present this technique in Section IV-B.

We provide the representation of our approach on a synthetic signal example in Fig. 1. The original signal at Level 0 consists of three main regions where the first and the second regions have different periodicities and the third region does not have any periodic activity. We first detect separation points of these regions (black dashed lines at Level 0) with our kernel density estimation based method. Then, we search for periodicities in these separated regions and detect the dominant periods for the first and the second regions at Level 1. The nested periods are detected in the second region and hence another iteration of period detection method is applied at Level 2. We show the separation lines of periodic regions with red dashed lines in Fig. 1.

A. CHANGE POINT DETECTION

Density estimates provide highly valuable information regarding data modeling. Therefore, density estimation is used for several purposes in the literature including anomaly detection [42], classification [43], and in particular change point detection [22]. In this paper, we use density estimation technique to model the behavior of the signal and extract the intervals with different behaviors. We do this to analyze the different periodic parts separately in the next steps so that we can detect all existing periodicities in the signal.

For modeling, we divide the signal into equal-length blocks and use the kernel density estimation (KDE) [44] method at each block. Let $\mathbf{x}$ be a point and $\mathcal{S}$ be a set of points in one block. Then, KDE is obtained as

$$f_{\mathcal{S}}(\mathbf{x}) = \frac{1}{w\delta} \sum_{i:x_i \in \mathcal{S}, 1 \leq i \leq w} K\left(\frac{\mathbf{x} - x_i}{\delta}\right), \tag{1}$$

where w is the number of instances in $\mathcal{S}$, δ is the bandwidth and $K(\cdot)$ is a kernel function [45]. In this work, we use the Gaussian kernel function [44] $K(x) \triangleq \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$.

Since data blocks containing different behaviors construct different distributions, we can detect change points based on the distribution change points. This can be achieved by exploiting distribution specific properties. At this point, we use bandwidth parameter of KDE, which is also calculated optimally based on the corresponding data points [46]. In other words, we map the observed signal into a low dimensional space, where KDE bandwidths represent the distributional behaviors of the original signal. Then, we declare the points where drastic bandwidth deviations are observed as original signal change points. We provide our KDE based change point detection algorithm in Algorithm 1 as a pseudocode.

Remark: Note that since we obtain kernel densities and hence KDE bandwidths based on divided signal blocks, we obtain not the exact change points but blocks, i.e., set of data points, containing the change points. Therefore, we use average magnitude different function on detected change blocks to find the exact change points. This step is explained in detail in Section V where we also provide the experimental results with real signals.

B. HIERARCHICAL DETECTION OF MULTIPLE PERIODICITIES

The main idea of average magnitude difference function (AMDF) is similar to autocorrelation where one seeks to find highest similarity intervals by sliding window on a signal. The conventional AMDF [47] is calculated as

$$D_{\boldsymbol{x}}(\tau) = \frac{1}{N - \tau - 1} \sum_{i=0}^{N-\tau-1} |\boldsymbol{x}_w(i + \tau) - \boldsymbol{x}_w(i)|, \tag{2}$$

where $\boldsymbol{x}_w$ is the windowed quasi-periodic sequence with N data points. Therefore, $D(\tau)$ is minimized for $\tau = mT$, i.e., multiples of period T. In an ideal periodic case, $D(\tau)$ would be zero at points corresponding to exact multiples of period as $\tau = mT$. However, we search for τ values, which minimize $D(\tau)$ by sliding window. Therefore, it is applicable to noisy

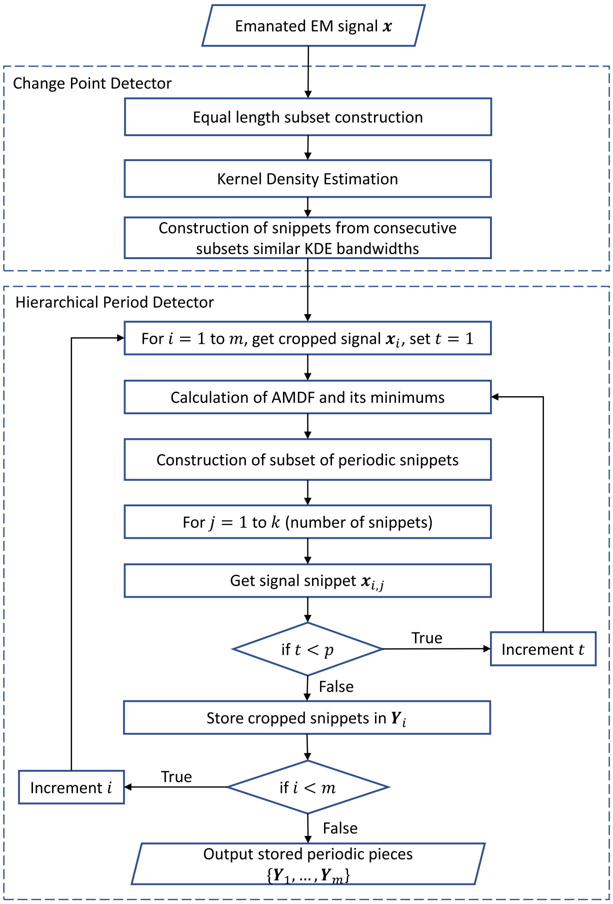

Emanated EM signal x
Change Point Detector
Equal length subset construction
Kernel Density Estimation
Construction of snippets from consecutive subsets similar KDE bandwidths
Hierarchical Period Detector
For $i = 1$ to m, get cropped signal x_i, set $t = 1$
Calculation of AMDF and its minimums
Construction of subset of periodic snippets
For $j = 1$ to k (number of snippets)
Get signal snippet $x_{i,j}$
if $t < p$
True
Increment t
False
Store cropped snippets in Y_i
Increment i
True
if $i < m$
False
Output stored periodic pieces $\{Y_1, \ldots, Y_m\}$

FIGURE 3. **Flowchart of the proposed method for multiple periodicity detection.**

quasi-periodic signals which do not have one exact dominant period but multiple periodicities.

In our framework, we apply AMDF iteratively to extract all periods. For a sequence $\boldsymbol{x} \in \mathbb{R}^N$, let $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$ be the longest periodic parts, where the spacing between them is not necessarily the same. We first find AMDF of $\boldsymbol{x}$ as $D_{\boldsymbol{x}}(\tau)$, whose minimums corresponds to starting points of all $\boldsymbol{x}_i$ for $i = \{1, 2, \ldots, m\}$. Then, we separately analyze all $\boldsymbol{x}_i$ subsets to find smaller periods. Assume that $\boldsymbol{x}_i$ can be divided into periodic parts as $\{\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,m_i}\}$. We can find all these parts from the minimums of $D_{\boldsymbol{x}_i}(\tau)$ calculated with smaller window length. Then, we analyze each $\boldsymbol{x}_{i,j}$ until finding all periodic signal parts as in Algorithm 2.

The flowchart of the proposed method is given in Fig. 3. Here, the EM signal collected from a program running device is given as an input signal to the system. First, Change Point Detector algorithm separates the signal into equal length signal blocks to construct their kernel density estimations (KDEs). It combines consecutive signal blocks showing similar KDE bandwidths by cropping the signal from bandwidth change points. Then, the resulting signal snippets are fed into the Hierarchical Period Detector. This algorithm separately processes signal snippets with different behaviors. It calculates the AMDF and its minimums of the first snippet.

Algorithm 1 KDE Based Change Point Detector

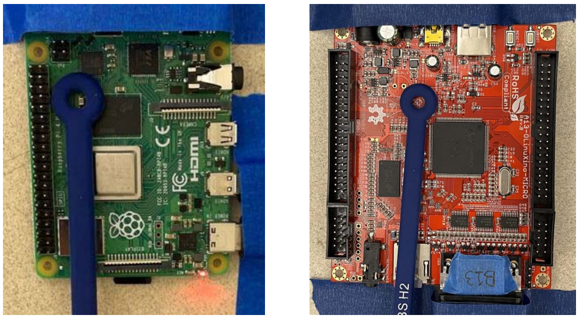
```
Observe signal $\boldsymbol{x}^{1\times N}$ containing $m$ snippets, i.e., $m-1$ change points.
Set window length $w$
Divide the signal $\boldsymbol{x}$ into $k = \frac{N}{w}$ blocks to obtain subsets as $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k\}$
for $t = 1, 2, \ldots, k$ do
    Calculate optimal KDE bandwidth $\delta_t$ for $\boldsymbol{x}_t$
    Obtain the distribution of $\boldsymbol{x}_t$ with KDE as $f_t(x) = \frac{1}{n\delta_t}\sum_{j=1}^{n} K(\frac{x-x_j}{\delta})$
end for
Calculate the average $\mu = \frac{1}{k}\sum_{j=1}^{k}\delta_t$
Declare block indices $i$ satisfying $\delta_j > \mu$ or $\delta_j < \mu$ for $j = i, i+1, \ldots, i+h$ as change points
```

Algorithm 2 Hierarchical Period Detector

```
Obtain signal snippet $\boldsymbol{x}^{1\times T}$ with Algorithm 1
Set iteration number $p$ and window lengths $w_1, \ldots, w_p$
Set an index number $g < T$
for $j = 1, 2, \ldots, g$ do
    Crop $\boldsymbol{x}$ starting from $j^{\text{th}}$ index to obtain $\tilde{\boldsymbol{x}}_j$
    Calculate AMDF $D_j$ with $\tilde{\boldsymbol{x}}_j$
    Set $L(j) = \min(D_j)$
end for
Declare periodicity starting point as $\hat{j} = \min(L)$
Crop $\boldsymbol{x}$ starting from $\hat{j}^{\text{th}}$ index to obtain $\hat{\boldsymbol{x}}$
for $t = 1, 2, \ldots, p$ do
    Get the matrix $\boldsymbol{Y}^{k\times h}$ of $k$ snippets ($\boldsymbol{Y} = \hat{\boldsymbol{x}}$ at $t = 1$ )
    for $i = 1, 2, \ldots, k$ do
        Get $i^{\text{th}}$ snippet $\boldsymbol{y}_i^{1\times h}$
        Calculate $D(\tau) = \frac{1}{n-1-\tau}\sum_{i=1}^{n-1-\tau}|y(i) - y(i+\tau)|$
        Construct the set $R(i) = j$ where $D(j) = \min\{D(i), D(i+1), \ldots, D(i+w_t)\}$
        Store $d$ snippets with initial indices in $R$ as $\boldsymbol{Y}_i^{d\times s}$
    end for
end for
```

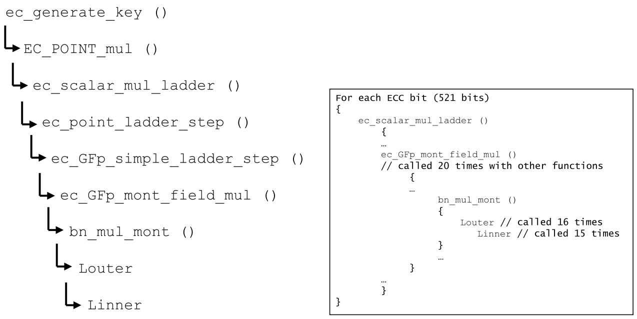
This constructs the first set of periodic pieces. Then, it calculates AMDF and the corresponding periodicities of this signal piece until the iteration number is achieved. After storing the smallest periodic pieces of the first snippet, it starts processing the second snippet with different periodic behavior. At the end of this process, the algorithm outputs all stored periodic pieces as shown in Fig. 3.

The visualization of this process can be seen in Fig. 1. At Level 0, there are three different regions to be analyzed. The method does not detect any periodicity for the third region since it contains only noise. When the first and second regions are processed, the separations of periodic regions are obtained as in Level 1. The structure stops branching for the first region since it does not contain any nested periods. It processes only the second region to detect the smaller periodic regions in Level 2 and completes the process.



(a) (b)

FIGURE 4. Measurement setups used to collect emanated EM signals for (a) Raspberry Pi 4, (b) A13-OLinuXino devices.



```
For each ECC bit (521 bits)
{
    ec_scalar_mul_ladder ()
        {
        …
        ec_GFp_mont_field_mul ()
        // called 20 times with other functions
             {
             …
                  bn_mul_mont ()
                  {
                      Louter // called 16 times
                         Linner // called 15 times
                  }
                  …
             }
        …
        }
}
```

(a) (b)

FIGURE 5. (a) ECC code flow with corresponding functions, (b) ECC code structure.

Remark: Note that the observed signal may contain nested periods as in the second region of the example in Fig. 1 (the signal region on the right at Level 1). Here, this cropped signal contains periodic activity repeating four times and each of them has smaller periodic regions repeating five times. However, there are additional nonperiodic noisy regions at the end of each four dominant repetition, which disrupts the perfect periodicity and disables one detecting these 20 periodic activities directly. We see the advantage of our method at this point. Since we do not assume perfect periodicity and we take behavioral changes into account by processing the regions separately, our method can successfully detect all existing periodicites. We provide the experimental results regarding this discussion with real life signal examples in Section V.

We provide our resulting iterative pattern detection algorithm in Algorithm 2 as a pseudocode. We apply this procedure to every signal interval found in Algorithm 1 separately. Here, we assume that the periodic part may not be in the beginning of the signal snippet obtained with Algorithm 1. Therefore, we first find the starting point of the periodicity, which provides minimum AMDF value (line 4-9 in Algorithm 2) and crop the signal starting from this point to obtain $\hat{\boldsymbol{x}}$. Then, at iteration $t = 1$ we apply AMDF to this cropped periodic signal $\boldsymbol{Y} = \hat{\boldsymbol{x}}$ (at $t = 1$; $k = 1$ and h equals to the length of the signal in line 12 of Algorithm 2) and extract its longest periodic snippets (line 15 and 16 in Algorithm 2). We store these k snippets with h data instances as $\boldsymbol{Y}^{k \times h}$ (line 17 in Algorithm 2). At the next iteration $t = 2$, we get this set of snippets $\boldsymbol{Y}^{k \times h}$ (line 12 in Algorithm 2), extract its periodic pieces and store them as $\boldsymbol{Y}_i^{d \times s}$ to be analyzed in the next iteration (line 17 in Algorithm 2). We continue this procedure until finding the smallest periodic pattern in the signal.

Remark: In the case of ideal exact periodic signal, the minimums of AMDF correspond to the starting points of the periodic parts. However, for generalization, we assume that the signal from beginning to end may not contain periodic parts consecutively. In such cases, even though AMDF still has local minimums, they would be greater than the minimums of actual periods. Therefore, we can eliminate these by thresholding the detected local minimums.

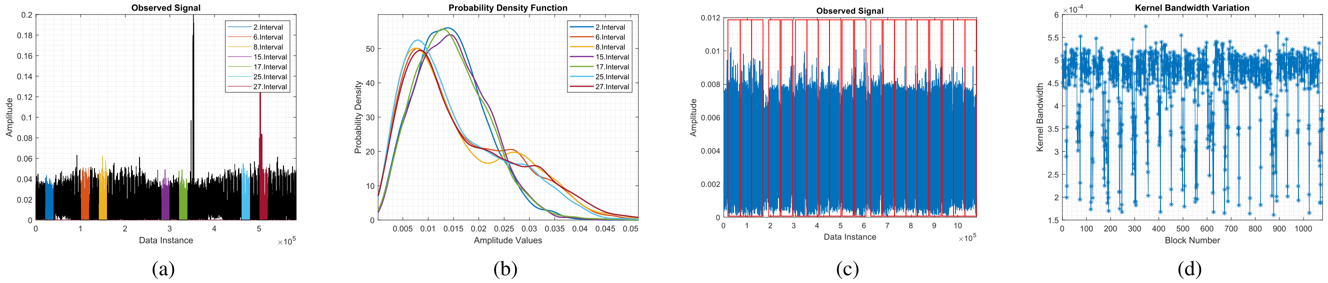
V. EXPERIMENTAL RESULTS

To illustrate how proposed method works on practical data, here we provide brief description of measurement setups that are used to collect the signals with multiple periods and present results obtained using the proposed method on these signals.

A. MEASUREMENT SETUP

In the experiments, we use two devices, i.e., Raspberry Pi 4 and A13-OLinuXino to obtain the set of electromagnetic (EM) side-channel signals as in Fig. 4. Raspberry Pi 4 single-board computer has quad-core ARM Cortex A72 processor, whose operating frequency is set to 1.2 GHz. A13-OLinuXino is a single-board embedded Linux computer with ARM Cortex A8 processor, whose operating clock frequency is 1 GHz.

We run ECC (Elliptic Curve Cryptography) implementation of OpenSSL [48] on the boards where we provide its code flow in Fig. 5a. ECC is an encryption method, which provides high security against side-channel attacks in a computationally efficient way. Since the main idea is mapping elements through a defined curve structure, it requires a sequence of operations including additions and multiplications. These iterative steps result in observable nested loops in emanated EM signals, which is investigated in this section. Here, we use the code structure in Fig. 5b. This code calls "ec_scalar_mul_ladder ()" for each bit of 521 bits. This function contains some operations along with "ec_GFp_mont_field_mul ()", which is called 20 times. Hence, the resulting signal contains 20 iterative loops (operations) separated unevenly. Note that "ec_GFp_mont_field_mul ()" also consists of some operations and "bn_mul_mont ()", which includes 16 iterations of "Louter". And finally, "Louter" iterates"Linner" 15 times. Therefore, each of 20 loops contain 16 nested periodic loops. And, each of 16 loops consist of 15 periodic loops, which correspond to the smallest periodic activity in the signal. Also, note that the additional functions in between 16 outer loops corrupt perfect periodicity, which disables one to detect 16× 15 loops directly from periodogram or autocorrelation function of complete signal.



(a) (b) (c) (d)

FIGURE 6. (a) EM signal intervals belonging to two different behavior groups (signal blocks of 2-15-17 and 6-8-25-27 are different from each other) (b) Kernel density estimation based distributions constructed with corresponding data intervals shown in Fig. 6a. (c) EM signal collected from A13-OLinuXino device where data intervals of the same processor activities are shown with 20 red frames (d) KDE bandwidths of the signal blocks in Fig. 6c.

The emanated EM signals are collected with Aaronia H2 near-field magnetic probe, which is located around the pins of the processors of the target devices as in Fig. 4. For recording of the signal collected from Raspberry Pi 4, Keysight UXA signal analyzer is used with 1.28 GHz sampling rate. The signal collected from A13-OLinuXino device is recorded with Keysight DSOS804A oscilloscope with 10 GHz sampling frequency. Note that the signal collected from Raspberry Pi 4 is filtered for interference removal before processing. However, the signals of A13-OLinuXino device are directly used without any pre-processing to show the efficiency of the proposed method.

B. MULTIPLE PERIODICITY DETECTION RESULTS

In this section, we present experimental results of each step in the proposed method. We first provide our results for change detection on the signal collected from Raspberry Pi 4 device. As seen in Fig. 2, this signal consists of two different behaviors internally, i.e., one contains repeating loops and the other is noise-like behavior, which are highlighted with different colors. Our aim is to detect the change points and separate these parts.

1) CHANGE POINT DETECTION

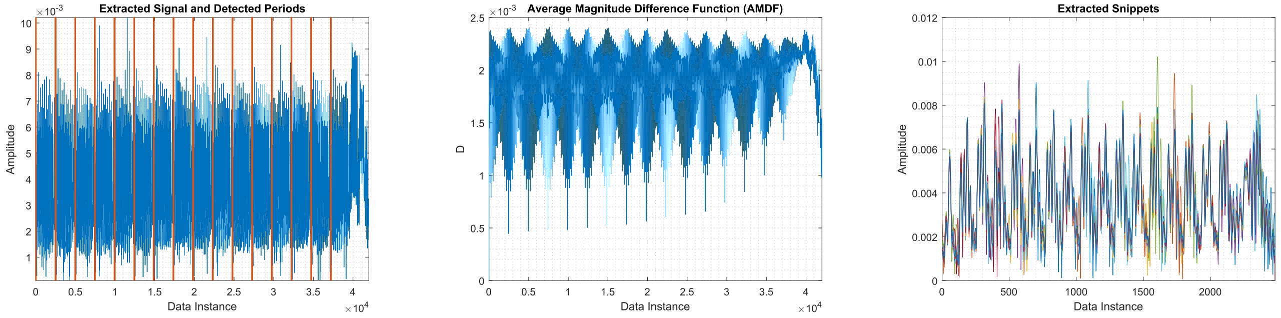
Our method first divides the signal into blocks with equal lengths. We choose block length as 2×10^4 data instances for the signal in Fig. 2. For representation, we provide different coloring of this same signal and highlight the data intervals corresponding to blocks of 2, 6, 8, 15, 17, 25 and 27 as in Fig. 6a. Note that 2., 15. and 17. blocks share the same behavior as in the first and third frames in Fig. 2. On the other hand, 6., 8., 25. and 27. blocks are similar to each other as belonging to the second and fourth frames in Fig. 2. When we construct distributions of these data blocks with kernel density estimation (KDE), we obtain Fig. 6b. As seen from the resulting distributions, the data instances having similar behaviors construct similar distributions. Hence, kernel density estimation can be used to track the behavioral changes.

We also use the signal collected from A13-OLinuXino device to detect change points and periods. This signal contains 20 unevenly spaced loops corresponding to the same activities running on the processor of the device. Hence it is periodic in a sense that the 20 intervals shown with red frames in Fig. 6c correspond to the same loops and their lengths are equal to each other but they do not have the same spacing between themselves (some of them are consecutive while some of them are a distance apart) and the noise is also different for each data instance. In this example, we interpret this variant of periodicity as a change point detection problem. For this, we specify the block length as 1000 and construct distribution of each signal block. The kernel bandwidths of the blocks are given in Fig. 6d. Here, we observe drastic bandwidth changes at the beginning and end blocks of all 20 loops. We detect these change blocks by comparing the bandwidths with their average value.

2) PERIODICITY DETECTION

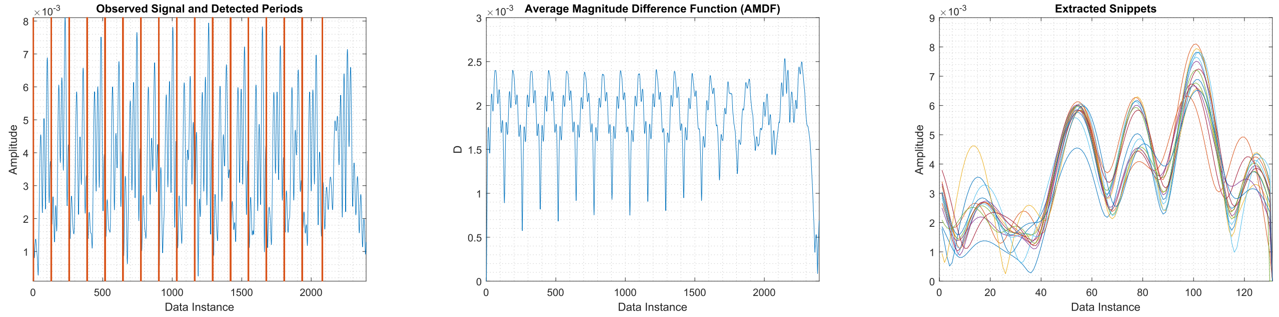
In the next step, we analyze each of detected 20 signal intervals (main loops) separately. Note that each of them contains 16 smaller loops (outer loops) where each of 16 also contains 15 even smaller periodic pieces (inner loops), which results in a multi-periodic signal.

For better visualization, we provide 16 outer loops and their separation points in Fig. 7a where their inner loops can also be seen in Fig. 8a. Here, the aim is to detect all 16 outer loops and 240 (15 × 16) inner loops with the algorithm. However, the challenge is that there are additional signal activities (as seen at the end of inner loops in Fig. 8a) between 16 outer loops, which disrupt the perfect periodicity in terms of inner loops. Therefore, even though a dominant frequency is found, the signal cannot be divided into its loop intervals correctly with that value. This disables one to obtain the signature periodic behavior of an observed signal. To observe this, we apply periodogram and autocorrelation [34] as period detection methods to our signals to detect all existing loops for comparison. Note that these methods are only applied to the signals whose behaviors do not change in time since they do not address quasi-periodicity. Autocorrelation function provides correlation values between original signal and its delayed versions. The lag values at the peaks correspond to the signal periods. Periodogram provides power spectral density estimate of a signal where one can extract the dominant



(a) (b) (c)

FIGURE 7. **(a) Periodic signal part, which is detected with change point detection step, containing 16 loops, and detected starting points corresponding to periodic loops, (b) Obtained average magnitude difference function where local minimums correspond to the periodic parts of Fig. 7a, (c) All 16 detected loops plotted on top of each other.**



(a) (b) (c)

FIGURE 8. **(a) Periodic signal part, which is detected with the first iteration of hierarchical period detector, containing 15 loops, and detected starting points, (b) Obtained average magnitude difference function where local minimums correspond to the periodic parts of Fig. 8a, (c) All 15 detected loops plotted on top of each other.**

frequencies from the peak locations. Based on the application of these methods to our collected signal with 20 unevenly spaced loops (each containing 16 outer and 15 inner loops), we observed that periodogram detects only frequency component corresponding to the inner loops along with many false alarms and fails to detect outer and main loops. Also, autocorrelation function detects lag values corresponding to outer and inner loops while it fails to detect 20 main loops similarly. However, even the detected values do not directly correspond to actual loop separations because of the additional nonperiodic signals between the outer loops. As a result, if we divide the signal into smallest periodic pieces with these detected values, the resulting signal does not represent the actual periodic behavior of the signal. Therefore, they are not directly applicable to many real life applications requiring accurate pattern detection.

On the other hand, we compensate the previously discussed issues with our hierarchical structure. As explained in Section V-B1, we first search for behavioral similarities and change points. With this step, we detect 20 unevenly spaced iterations. Then, we analyze them separately to find smaller periodicities. We also search for the starting point of periodic behaviors for accurate separation. Therefore, our method accurately detects all periodic activities even when there are inserted nonperiodic activities between periodic loops while autocorrelation, periodogram and similar methods suffer from this situation.

For representation, we take one of the 20 loops detected in the previous part as seen in Fig. 7a. In the first iteration of our method, we obtain the AMDF in Fig. 7b. We declare the local minimums as starting points of the loops and obtain the division shown in Fig. 7a. When we plot them on top of each other, we obtain the highly overlapping snippets in Fig. 7c, all of which will be analyzed separately in the next iteration.

Similar to the previous steps, we take one of the detected small loops in Fig. 7c for representation and search for 15 smallest loops existing in this signal. As seen from Fig. 7c, the periodic parts do not start in the beginning of the signal. For these cases, our method also applies optimization to find the starting point of the periodic loops. For this, we crop the signal beginning from several points and calculate the minimum AMDF for each case. Then, the best cropping point giving minimum AMDF corresponds to capturing maximum periodicity. In this case, we find the starting point as 87th data instance and get the cropped snippet given in Fig. 8a. Then, we apply AMDF iteration and obtain Fig. 8b. Here, the local minimums give all 15 periodic parts as in Fig. 8a. When we draw them on top of each other, we get the result in Fig. 8c.

As seen from this figure, these loops are not exactly periodic such that they do not completely overlap and even the ones that are most similar have timing differences due to sampling. However, even in such cases, our iterative technique is able to find all periodicities with an efficient end-to-end processing.

VI. CONCLUSION

This paper presented the first signal processing technique that address finding periodicities in electromagnetic side channels that analyze software activities. It introduces an end-to-end processing method for multiple nested periodicity signal detection and analysis. The probabilistic distributions of signal blocks are estimated with kernel density estimation. The corresponding kernel bandwidths, which are optimally found in a data-driven manner, are used to detect change points. After separating the signal into parts with different behaviors, average magnitude difference function is leveraged iteratively to find the smallest periodic signal sections. To illustrate efficiency of the proposed method, we use EM side-channel signals collected from real-life applications to successfully detect multiple existing periodicities.

ACKNOWLEDGMENT

The views and finding in this paper are those of the authors and do not reflect the views of ONR.

REFERENCES

[1] P. Huijse, P. A. Estevez, P. Protopapas, P. Zegers, and J. C. Principe, "An information theoretic algorithm for finding periodicities in stellar light curves," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5135–5145, Oct. 2012.

[2] J. Hlavnicka, R. Cmejla, J. Klempir, E. Ruzicka, and J. Rusz, "Acoustic tracking of pitch, modal, and subharmonic vibrations of vocal folds in Parkinson's disease and parkinsonism," *IEEE Access*, vol. 7, pp. 150339–150354, 2019.

[3] F. Erden and A. E. Cetin, "Period estimation of an almost periodic signal using persistent homology with application to respiratory rate measurement," *IEEE Signal Process. Lett.*, vol. 24, no. 7, pp. 958–962, Jul. 2017.

[4] I. V. L. Clarkson, "Approximate maximum-likelihood period estimation from sparse, noisy timing data," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1779–1787, May 2008.

[5] M. Backes, M. Dürmuth, S. Gerling, M. Pinkal, and C. Sporleder, "Acoustic side-channel attacks on printers," in *Proc. 19th USENIX Secur. Symp.*, Washington, DC, USA, Aug. 2010, pp. 307–322.

[6] J. Quisquater and D. Samyde, "Electromagnetic analysis (EMA): Measures and counter-measures for smart cards," in *Proc. Int. Conf. Res. Smart Cards, E-Smart*, Cannes, France, Sep. 2001, pp. 200–210.

[7] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, Paris, France, May 2001, pp. 251–261.

[8] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM side-channel(s)," in *Proc. 4th Int. Workshop Cryptograph. Hardw. Embedded Syst.*, Redwood Shores, CA, USA, Aug. 2002, pp. 29–45.

[9] E. De Mulder, S. B. Örs, B. Preneel, and I. Verbauwhede, "Differential power and electromagnetic attacks on a FPGA implementation of elliptic curve cryptosystems," *Comput. Elect. Eng.*, vol. 33, nos. 5–6, pp. 367–382, 2007.

[10] D. Genkin, L. Pachmanov, I. Pipman, and E. Tromer, "Stealing keys from PCs using a radio: Cheap electromagnetic attacks on windowed exponentiation," in *Proc. 17th Int. Workshop Cryptograph. Hardw. Embedded Syst.*, Saint-Malo, France, Sep. 2015, pp. 207–228.

[11] D. Genkin, A. Shamir, and E. Tromer, "RSA key extraction via low-bandwidth acoustic cryptanalysis," in *Proc. Annu. Cryptol. Conf.*, Santa Barbara, CA, USA, Aug. 2014, pp. 444–461.

[12] A. Zajic and M. Prvulovic, "Experimental demonstration of electromagnetic information leakage from modern processor-memory systems," *IEEE Trans. Electromagn. Compat.*, vol. 56, no. 4, pp. 885–893, Aug. 2014.

[13] B. B. Yilmaz, A. Zajic, and M. Prvulovic, "Modelling jitter in wireless channel created by processor-memory activity," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2037–2041.

[14] M. Alam, H. A. Khan, M. Dey, N. Sinha, R. L. Callan, A. G. Zajic, and M. Prvulovic, "One&Done: A single-decryption EM-based attack on OpenSSL's constant-time blinded RSA," in *Proc. 27th USENIX Secur. Symp.*, Baltimore, MD, USA, Aug. 2018, pp. 585–602.

[15] R. Callan, A. Zajic, and M. Prvulovic, "A practical methodology for measuring the side-channel signal available to the attacker for instruction-level events," in *Proc. 47th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Dec. 2014, pp. 242–254.

[16] R. Callan, N. Popovic, A. Daruna, E. Pollmann, A. Zajic, and M. Prvulovic, "Comparison of electromagnetic side-channel energy available to the attacker from different computer systems," in *Proc. IEEE Int. Symp. Electromagn. Compat. (EMC)*, Aug. 2015, pp. 219–223.

[17] R. Callan, F. Behrang, A. Zajic, M. Prvulovic, and A. Orso, "Zero-overhead profiling via EM emanations," in *Proc. 25th Int. Symp. Softw. Test. Anal.*, Jul. 2016, pp. 401–412.

[18] M. Dey, A. Nazari, A. Zajic, and M. Prvulovic, "EMPROF: Memory profiling via EM-emanation in IoT and hand-held devices," in *Proc. 51st Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Oct. 2018, pp. 881–893.

[19] H. A. Khan, M. Alam, A. Zajic, and M. Prvulovic, "Detailed tracking of program control flow using analog side-channel signals: A promise for IoT malware detection and a threat for many cryptographic implementations," *Proc. SPIE*, vol. 10630, May 2018, Art. no. 1063005.

[20] R. Rutledge, S. Park, H. Khan, A. Orso, M. Prvulovic, and A. Zajic, "Zero-overhead path prediction with progressive symbolic execution," in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng. (ICSE)*, May 2019, pp. 234–245.

[21] L. N. Nguyen, C.-L. Cheng, M. Prvulovic, and A. Zajic, "Creating a backscattering side channel to enable detection of dormant hardware trojans," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 7, pp. 1561–1574, Jul. 2019.

[22] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 339–367, May 2017.

[23] P. Delacourt and C. J. Wellekens, "DISTBIC: A speaker-based segmentation for audio data indexing," *Speech Commun.*, vol. 32, nos. 1–2, pp. 111–126, Sep. 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167639300000273

[24] F. Li, G. C. Runger, and E. Tuv, "Supervised learning for change-point detection," *Int. J. Prod. Res.*, vol. 44, no. 14, pp. 2853–2868, Jul. 2006.

[25] E. Bakstein, J. Schneider, T. Sieger, D. Novak, J. Wild, and R. Jech, "Supervised segmentation of microelectrode recording artifacts using power spectral density," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2015, pp. 1524–1527.

[26] V. Gupta, "Speaker change point detection using deep neural nets," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 4420–4424.

[27] A. Gupta, V. S. Masampally, V. Jadhav, A. Deodhar, and V. Runkana, "Supervised operational change point detection using ensemble long-short term memory in a multicomponent industrial system," in *Proc. IEEE 19th World Symp. Appl. Mach. Intell. Informat. (SAMI)*, Jan. 2021, pp. 000135–000141.

[28] V. Kartik, D. S. Satish, and C. C. Sekhar, "Speaker change detection using support vector machines," in *Proc. ISCA Tutorial Res. Workshop (ITRW) Non-Linear Speech Process.*, 2005, pp. 1–7.

[29] K. Frick, A. Munk, and H. Sieling, "Multiscale change point inference," *J. Roy. Stat. Soc., B, Stat. Methodol.*, vol. 76, no. 3, pp. 495–580, Jun. 2014.

[30] P. Fearnhead, "Exact and efficient Bayesian inference for multiple change-point problems," *Statist. Comput.*, vol. 16, pp. 203–213, Jun. 2006.

[31] S. Gonzalez and M. Brookes, "PEFAC—A pitch estimation algorithm robust to high levels of noise," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 2, pp. 518–530, Feb. 2014.

[32] A. Drutsa, G. Gusev, and P. Serdyukov, "Periodicity in user engagement with a search engine and its application to online controlled experiments," *ACM Trans. Web*, vol. 11, no. 2, pp. 1–35, May 2017.

[33] M. Vlachos, P. Yu, and V. Castelli, "On periodicity detection and structural periodic similarity," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2005, pp. 449–460.

[34] S. S. Upadhya, "Pitch detection in time and frequency domain," in *Proc. Int. Conf. Commun., Inf. Comput. Technol. (ICCICT)*, Oct. 2012, pp. 1–5.

[35] R. G. Amado and J. V. Filho, "Pitch detection algorithms based on zero-cross rate and autocorrelation function for musical notes," in *Proc. Int. Conf. Audio, Lang. Image Process.*, Jul. 2008, pp. 449–454.

[36] L. Hui, B.-Q. Dai, and L. Wei, "A pitch detection algorithm based on AMDF and ACF," in *Proc. IEEE Int. Conf. Acoust. Speed Signal Process.*, vol. 1, May 2006, p. 1.

[37] S. Kumar, S. K. Singh, and S. Bhattacharya, "Performance evaluation of a ACF-AMDF based pitch detection scheme in real-time," *Int. J. Speech Technol.*, vol. 18, no. 4, pp. 521–527, Dec. 2015.

[38] W. Zhang, G. Xu, and Y. Wang, "Pitch estimation based on circular AMDF," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2002, pp. 1–4.

[39] W. Fan, Y. X. Li, K. L. Tsui, and Q. Zhou, "A noise resistant correlation method for period detection of noisy signals," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 2700–2710, Jul. 2018.

[40] Q. Wen, K. He, L. Sun, Y. Zhang, M. Ke, and H. Xu, "RobustPeriod: Robust time-frequency mining for multiple periodicity detection," in *Proc. Int. Conf. Manage. Data*, Jun. 2021, pp. 2328–2337.

[41] S. V. Tenneti and P. P. Vaidyanathan, "Nested periodic matrices and dictionaries: New signal representations for period estimation," *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3736–3750, Jul. 2015.

[42] M. Kerpicci, H. Ozkan, and S. S. Kozat, "Online anomaly detection with bandwidth optimized hierarchical kernel density estimators," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 9, pp. 4253–4266, Sep. 2021.

[43] P. Mantero, G. Moser, and S. B. Serpico, "Partially supervised classification of remote sensing images through SVM-based probability density estimation," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 559–570, Mar. 2005.

[44] S. J. Sheather, "Density estimation," *Stat. Sci.*, vol. 19, pp. 588–597, Nov. 2004.

[45] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, vol. 26. Boca Raton, FL, USA: CRC Press, 1986.

[46] A. W. Bowman and A. Azzalini, *Applied Smoothing Techniques for Data Analysis: The Kernel Approach With S-Plus Illustrations*, vol. 18. Oxford, U.K.: Oxford Univ. Press, 1997.

[47] M. Ross, H. Shaffer, A. Cohen, R. Freudberg, and H. Manley, "Average magnitude difference function pitch extractor," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-22, no. 5, pp. 353–362, Oct. 1974.

[48] OpenSSL Software Foundation. *OpenSSL: Cryptography and SSL/TLS Toolkit*. Accessed: Jan. 2021. [Online]. Available: https://www.openssl.org

MINE KERPICCI (Graduate Student Member, IEEE) received the B.S. degree from Middle East Technical University, in 2017, and the M.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2019. She is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Her research interests include signal processing, machine learning, and optimization.

MILOS PRVULOVIC (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Belgrade, in 1998, and the M.Sc. and Ph.D. degrees in computer science from the University of Illinois at Urbana–Champaign, in 2001 and 2003, respectively. He is a Professor at the School of Computer Science, Georgia Institute of Technology, where he joined in 2003. His research interests include computer architecture, especially hardware support for software monitoring, debugging, and security. He was a past recipient of the NSF CAREER Award. He is a Senior Member of the ACM and the IEEE Computer Society.

ALENKA ZAJIĆ (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the School of Electrical Engineering, University of Belgrade, in 2001 and 2003, respectively, and the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, in 2008. She is currently a Ken Byers Professor at the School of Electrical and Computer Engineering, Georgia Institute of Technology. Prior to that, she was a Visiting Faculty Member at the School of Computer Science, Georgia Institute of Technology, a Postdoctoral Fellow at the Naval Research Laboratory, and a Design Engineer at Skyworks Solutions Inc. Her research interests include electromagnetic, wireless communications, signal processing, and computer engineering. She was a recipient of the 2017 NSF CAREER Award, the 2012 Neal Shepherd Memorial Best Propagation Paper Award, the Best Student Paper Award at the IEEE International Conference on Communications and Electronics 2014, the Best Paper Award at the International Conference on Telecommunications 2008, the Best Student Paper Award at the 2007 Wireless Communications and Networking Conference, and the Dan Noble Fellowship in 2004, which was awarded by Motorola Inc. and the IEEE Vehicular Technology Society for quality impact in the area of vehicular technology. She is currently the Editor of IEEE Transactions on Wireless Communications.