Divide-and-Conquer based Early Classification Approach for Multivariate-Time-Series with Different Sampling-Rate

Report submitted in fulfillment of the requirements for the Exploratory Project of

Second Year IDD.

by

Mudit Bhardwaj

Under the guidance of

Dr. Hari Prabhat Gupta

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI

Varanasi 221005, India

May 2017

Dedicated to

My parents, professors, exploratory project convenor, mentor and everyone who helped and motivated us in successful completion of this report.

Declaration

I certify that

1. *The work contained in this report is original and has been done by myself and the general supervision of my supervisor.*

2. *The work has not been submitted for any project.*

3. *Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.*

4. *Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.*

Place: IIT (BHU) Varanasi
Date: 10^{th} June 2020

Mudit Bhardwaj
IDD Student
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Certificate

This is to certify that the work contained in this report entitled "**Divide-and-Conquer based Early Classification Approach for Multivariate-Time-Series with Different Sampling-Rate**" being submitted by **Mudit Bhardwaj** (**Roll No. 18074011**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is a bona fide work of our supervision.

Place: IIT (BHU) Varanasi
Date: 10$^{\text{th}}$ June 2020

Dr. Hari Prabhat Gupta
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Acknowledgments

We want to express our sincere gratitude to the people who have helped us the most throughout our project. We are grateful to our project supervisor ***Dr. Hari Prabhat Gupta*** *for providing us an opportunity to implement the paper entitled “****Divide-and-Conquer based Early Classification Approach for Multivariate-Time-Series with Different Sampling-Rate****” and his constant support for the project. We wish to thank our parents for their support and attention. We would like to thank our friends who encouraged us and helped us out in finalizing the project. At last, we also thank the Almighty for his blessings.*

Place: IIT (BHU) Varanasi

Date: 10^{th} *June 2020* **Mudit Bhardwaj**

Contents

List of Figures

List of Symbols

Symbol	***Description***
α	*Desired accuracy of classification*
Λ_i	*Sampling rate of component* C_i
D	*A labeled MTD dataset*
h_i	*Constructed classifier for* C_i *component*
m_i	*No of data points in* C_i *component*
l	*No of class labels in dataset* D
T	*No of data points in a complete MTD*
C_j^i	*ith component of jth MTD in* D
N	*Total number of MTD in dataset* D

Chapter 1

Introduction

1.1 Overview

In the Internet of Things (IoT), sensor-based devices produce the Multivariate Time Series (MTS). The classification method is useful for guessing the label of the incoming MTS class. Due to the large size and different sampling rate of the sensors in a given MTS sample, a classifier takes time to predict a class label.

Some IoT systems may require early-label prediction of the class at which the classifier begins to predict when a small number of points have been collected. In this article, we tackle the problem of early prediction of the MTS class IoT label. This methords looks at the sensors of different sampling rates to produce MTS. Each sensor produces a time series (component) of the MTS. We propose a Divide-and-

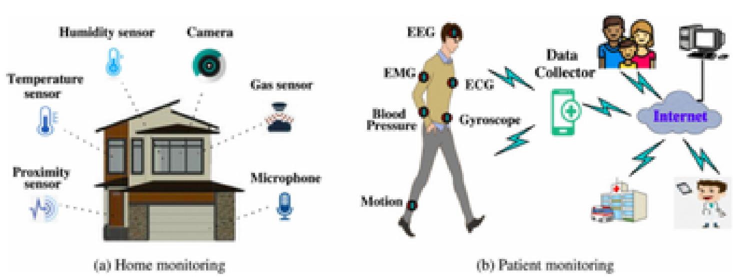


Figure 1.1 Illustration of home and patient monitoring using MTS generated by various sensors.

Conquer-based classification method for such MST. This approach creates a classifier using probabilistic isolation and integrated clustering. The ensemble classifier uses the Divide-and-Conquer method to handle different sample items during label prediction. The experimental results show that our method of divide and conquer based prediction outperforms the some of the existing approaches.

1.2 Motivation of the Research Work

*Generally an application of Internet of Things (IoT) system consist of several sensors which generate an MTS. In **figure1.1** ,sensors are shown which generate MTS, which helps in monitoring home and patients.It seem to be an inappropriate and an impractical assumption that all the sensors of an application in an IoT have same sampling rate.Such kind of impractical and inappropriate assumptions limit the usefulness of such applications in IoT.*

In this work, we have considered all of these limitations and have proposed a unique methord (divide-and-conquer based) for early predictions for an MTS.In this approach, we first calculate the Minimum Required Datapoints (MRD) by arranging all the components in a non decreasing manner of sampling rate.At last, we construct a tree of all such components and used ou divide and conquer approach to predict MTS.

Chapter 2

Project Work

2.1 Preliminaries

2.1.1 Assumptions

We will be assuming that our application of IoT consist of n sensors.All of these sensors which are going to constitute our MTS, are having different sampling rate.All these sensors collect data after a certain interval of time.Let us assume the full length of our time series be T. The MTS which consist of T data points in all its time series is known as complete MTS.

2.1.2 Component

The time series which is generated by a specific sensor over a fixed period of time is referred as a component of the MTS. It can also be referred as an ordered sequence of sensory data points.If we have n sensors then our MTS generated would consist of n components. Such MTS is denoted as $C = \{C_1, C_2, \ldots, C_n\}$ *where* C_i *denotes ith component.*

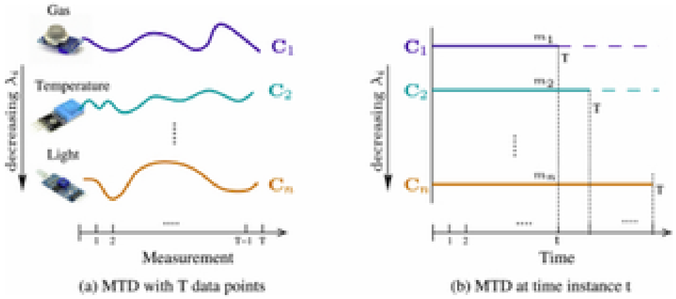


Figure 2.1 Illustration of an MTD with n components that are generated by various sensors.

2.1.3 Earliness

Let the full length of a component in the given MTD is T .The MRD is the number of minimum required data points to classify a component with α accuracy. The earliness of the component is defined as

$$Earliness = \frac{T-MRD}{T} \times 100$$

2.1.4 Accuracy

It refers to the percentage of correctly classified MTDs using class-wise MRD (with α desired level of accuracy). Let N denotes the number of correctly classified MTD in a given dataset with N MTD. Now, the accuracy can be mathematically expressed as

$$Accuracy = \frac{N'}{N} \times 100$$

2.1.5 Gaussian Process Classifier

A Gaussian Process (GP) [20] is an infinite set of space or time ordered random variables, where every finite set follows a multivariate normal distribution. The GP follows the joint distribution of those infinite random variables over continuous domain functions. It is a non-parametric classifier that models a time series as a finite set of *random variables, which are the outputs of a stochastic process. Let* $X = \{X[i] : 1 \leq t \leq T\}$ *is an MTS of dataset D with where n is number of components in X.*

2.2 EARLY CLASSIFICATION OF MTD (ECM)

I have tried to implement the following approach, which will classify the incoming MTD as early as possible with the desired accuracy(α *). This approach is known as Early Classification of MTD(ECM).*

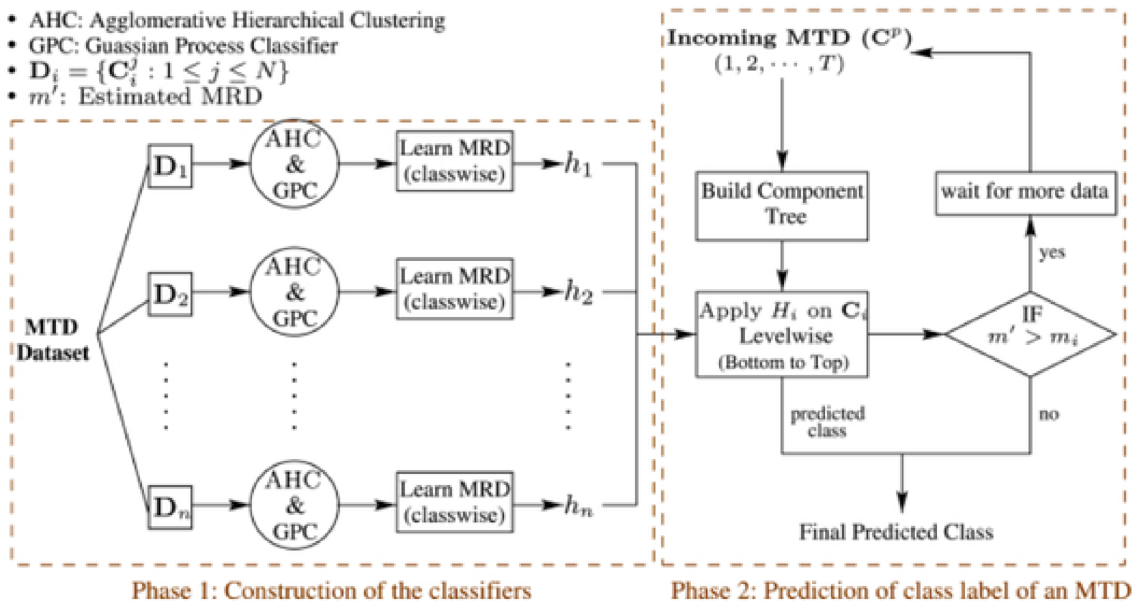


Figure 2.2 Overview of the proposed ECM approach. Phase 1 illustrates the construction of the classifiers and phase 2 depicts the prediction of an incoming MTD.

It consist of two steps:

1. *The first phase builds a set of classifiers* $h = \{h_1, h_2, \ldots, h_n\}$ *using the n components of the MTD of the given labeled dataset D. The classifiers maintain* α *accuracy.*

2. *In the second step, the ensemble classifier h which we built in the previous step, uses the Divide-and-Conquer technique to classify an incoming MTD as early*

as possible.

2.2.1 Construction of Classifiers

This is the first step of our Early Classification of MTD approach. During this step we are going to construct a set of classifiers $\boldsymbol{h}$ *by calculating the Minimum Required Datapoints (MRD) of the MTD.*

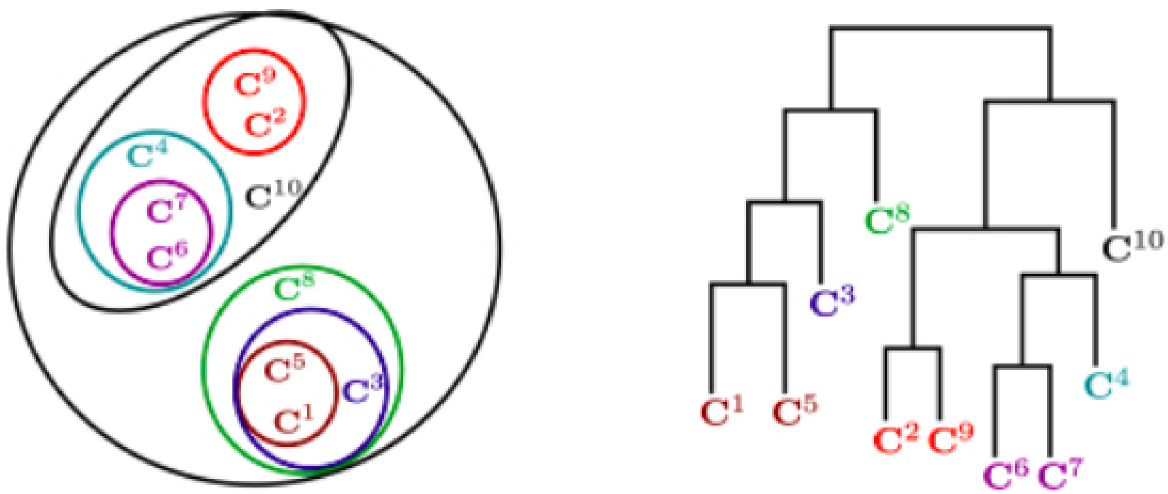
Estimate Minimum Required Data Points

In this step, we will calculate the class wise MRD seperately for each component of the MTD in our dataset D.For calculating MRD, it uses a probalistic classifier GP. In order to achieve earliness, full length of MTD must not be used to predict class.ECM first creates a dataset D_i *by seperating ith component from all the MTD.Then this dataset* D_i *is used to determine MRD for ith component. The GP classifier calculates the class probabilities for the time series* C_i^j *of the dataset* D_i*. For each component, minimum value of* t *is calculated such that we are able to acheive the desired accuracy* α*.*

Construct Classifier

In this step we are going to use our estimated MRD to build a set of classifiers. The classifier h_i *is built with the help of the calculated MRD for the dataset* D_i*. Basically our main objective is to build a classifier that can classify an MTD as early as possible with at least* α *accuracy.In order to ensure* α *accuracy ,ECM uses the Agglomerative clustering approach to build natural clusters of the time series.*

The Agglomerative clustering is a bottom-up hierarchical approach to build the clusters.This methord starts with a single element in each clusters and then at each steps, it merges two clusters that have the most similarity.Similarity between clusters is calculated with the help of Euclidean distance between MTD.



(a) Cluster formed at different steps

(b) Corresponding dendrogram

Figure 2.3 Illustration of the Agglomerative hierarchical clustering at different steps.

At last, the classifier calculate the class-wise MRD such that

$h_i = \{MRD_{i,k} : 1 \leq k \leq l\}$

Algorithm1:Construction of classifiers

Below is the pseudo code for the algorithm used for the construction of the classifiers.

ALGORITHM 1: Construction of the classifiers

Input: Dataset D consists of N labeled MTD with n time series and l class labels, the length of M
T and λ_i, where $1 \leq i \leq n$;

Output: A set of classifiers $\mathbf{h} = \{h_1, h_2, \ldots, h_n\}$;

/* Construct classifier h_1 using full length of time series in dataset D_1, using 1-NN*/

for *component* $i \leftarrow 2$ *to* n **do**

  for *label* $k \leftarrow 1$ *to* l **do**

    /* N_k is total number time series belonging to L_k */

    for $j \leftarrow 1$ *to* N_k **do**

      /* L_k is the label of $\mathrm{C}_i^j \in \mathrm{D}$.

      /* Compute class posterior probabilities of C_i^j using GP by Equation (8).*/

      $\mathrm{p_{i,j,T}} = \{p_{(i,j,T,1)}, p_{(i,j,T,2)}, \ldots, p_{(i,j,T,l)}\}$.

      $\delta_T = max_1\{\mathrm{p_{i,j,T}}\} - max_2\{\mathrm{p_{i,j,T}}\}$.

      for $t = 1$ *to* T **do**

        $\mathrm{p_{i,j,t}} = \{p_{(i,j,t,1)}, p_{(i,j,t,2)}, \ldots, p_{(i,j,t,l)}\}$.

        $\delta_t = max_1\{\mathrm{p_{i,j,t}}\} - max_2\{\mathrm{p_{i,j,t}}\}$.

        if $(\delta_T \leq \delta_t)$ *and* $(\boldsymbol{\alpha} \times max_1\{\mathrm{p_{i,j,T}}\} \leq max_1\{\mathrm{p_{i,j,t}}\})$ **then**

          $\mathrm{MRD}(\mathrm{C}_i^j) = t$.

    Obtain the natural group $\chi_{(i,L_k)}$ using Agglomerative hierarchical clustering.

    Find $MRD_{i,k}$ as the maximum of the MRD of the time series of G_{i,L_k}.

/* The classifier for ith component */

$h_i = \{MRD_{i,1}, MRD_{i,2}, \ldots, MRD_{i,l}\}$.

/* The classifiers for the dataset D */

$\mathbf{h} = \{h_1, h_2, \ldots, h_n\}$.

return h.

Figure 2.4 Pseudo code for construction of classifier.

2.2.2 Prediction of Class Label of an MTD

In this step, we are going to use the precalculated classifiers to classify an incoming MTD as soon as possible.The main focus of this part is to devise a Devide and conquer approach algorithm that can handle different components with different sampling rate and predict class label of C_i^j.

Divide-and-Conquer approach for an MTD*: This approach breaks the given problem into smaller(simpler) problems.After that the solution of the small problems are merged to obtain the final solution.This process is carried out in the following way:*

- ***Divide:*** *This part of the algorithm is responsible for dividing the large problem into smaller problem.It splits the group of time series of n sensors into smaller groups.For doing this we use the gap between the sampling rate of the consecutive components.If the gap between the sampling rate of consecutive components is same then we split the group into two equal halves, but if it is different then we split the gap in which the difference between sampling rate is maximum. The splitting process continues until only one components remains and it becomes the leaf node.*

- ***Conquer:****The main motive of this step is to apply the built classifier at different levels of the tree, starting from the bottom to top of the tree.Let us assume at a particular level z, x and y are two groups(left and right). Both at x and y we will predict a class. In order to predict the class, we will use minimum MRD i.e t data points of that component which have highest sampling rate among all components in node x.Then we predict a class using t data points. If there are other components in that node then we will calculate the no of datapoints available for that component till when component1 has received t datapoints.If the MRD of the next component is greated than the present available points* *then we will wait for more time and then predict class label and so on.*

Let L_x and L_y be the predicted class labels of the two nodes x and y at level z.Then the two groups are combined as follows:

- *If $L_x = L_y$, then combine x and y and assign L_X as new predicted class to the new combined group.*
- *If $L_x = L_y$, then find MRD denoted as t' required for y to predict label L_X in group y. If $t' \leq m_i$, then predict the label L_y, otherwise wait for more data points i.e. $t' - m_i$ and again predict in node y and this will be the final label.*

This process is repeated until level 0 is reached.Then the class predicted most number of times will be the answer.

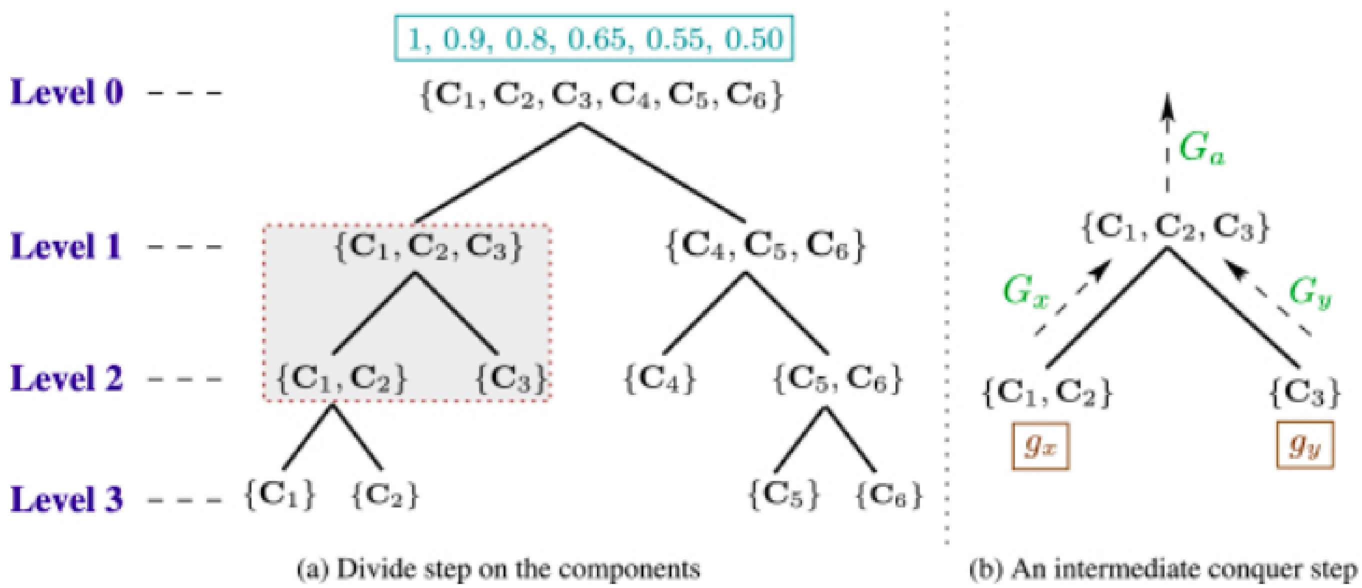


Figure 2.5 Illustration of an example of the Divide-and-Conquer method for the MTD with six components.

Algorithm1:Construction of classifiers

Below is the pseudo code for the algorithm used for the divide and conquer based prediction of MTD.

ALGORITHM 1: Construction of the classifiers

Input: Dataset D consists of N labeled MTD with n time series and l class labels, the lengt
T and λ_i, where $1 \leq i \leq n$;

Output: A set of classifiers $\mathbf{h} = \{h_1, h_2, \ldots, h_n\}$;

/* Construct classifier h_1 using full length of time series in dataset $\mathbf{D}_1$, using 1-NN*/

for *component* $i \leftarrow 2$ *to* n **do**
  for *label* $k \leftarrow 1$ *to* l **do**
    /* N_k is total number time series belonging to L_k */
    for $j \leftarrow 1$ *to* N_k **do**
      /* L_k is the label of $\mathrm{C}_i^j \in \mathrm{D}$.
      /* Compute class posterior probabilities of C_i^j using GP by Equation (8).*/
      $\mathbf{p}_{\mathbf{i,j,T}} = \{p_{(i,j,T,1)}, p_{(i,j,T,2)}, \ldots, p_{(i,j,T,l)}\}$.
      $\delta_T = max_1\{\mathbf{p}_{\mathbf{i,j,T}}\} - max_2\{\mathbf{p}_{\mathbf{i,j,T}}\}$.
      for $t = 1$ *to* T **do**
        $\mathbf{p}_{\mathbf{i,j,t}} = \{p_{(i,j,t,1)}, p_{(i,j,t,2)}, \ldots, p_{(i,j,t,l)}\}$.
        $\delta_t = max_1\{\mathbf{p}_{\mathbf{i,j,t}}\} - max_2\{\mathbf{p}_{\mathbf{i,j,t}}\}$.
        if $(\delta_T \leq \delta_t)$ *and* $(\boldsymbol{\alpha} \times max_1\{\mathbf{p}_{\mathbf{i,j,T}}\} \leq max_1\{\mathbf{p}_{\mathbf{i,j,t}}\})$ **then**
          $\mathrm{MRD}(\mathrm{C}_i^j) = t$.
    Obtain the natural group $\chi_{(i,L_k)}$ using Agglomerative hierarchical clustering.
    Find $MRD_{i,k}$ as the maximum of the MRD of the time series of G_{i,L_k}.

/* The classifier for ith component */
$h_i = \{MRD_{i,1}, MRD_{i,2}, \ldots, MRD_{i,l}\}$.
/* The classifiers for the dataset D */
$\mathbf{h} = \{h_1, h_2, \ldots, h_n\}$.
return h.

Figure 2.6 Pseudo code for divide-and-conquer based classification of MTD.

Chapter 3

Conclusions and Discussion

In this report, we have discussed a Divide-and-Conquer based classification algorithm which can be used to classify an MST of a given dataset.In this approach we have used the components of varying sampling rate.Our approach first calculate class-wise Minimum Required Data Points(MRD) using a probalistic GP classifier and agglomerative clustering.Then afterwards, it construct a set of classifier using the calculate MRDs.Finally we make a tree of components,and then using divide and conquer we predict the class label of an MTD. Our accuracy and earliness of the experiments performed on Daily Sports and Activities dataset illustrates that our approach outperforms the existing approaches of time series prediction.The proposed approach can be further extended to achieve better earliness using deep-learning models by optimizing the correlation among the components of MTD. Further, as the proposed approach is heavily dependent on the user given desired accuracy, a method can be developed to determine such accuracy based on application domain. Such a method will help to improve the effectiveness of the approach.

Following are the results when we use ECM on dataset Daily sports and Activities:

- ***Accuracy:*** *92.30%*
- ***Precision:*** *15.09%*

Chapter 4

Bibliography

1. *S. Aminikhanghahi, T. Wang, and D. J. Cook. 2019. Real-time change point detection with application to smart home time series data. IEEE Trans. Knowl. Data Eng. 31, 5 (2019), 1010−1023.*

2. *A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. 2017. The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Discov. 31, 3 (2017), 606−660.*

3. *T. G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. 10, 7 (1998), 1895−1923.*

4. *G. He, Y. Duan, R. Peng, X. Jing, T. Qian, and L. Wang. 2015. Early classification on multivariate time series. Neurocomputing 149 (2015), 777−787.*

5. *Y. Kuo, C. Li, J. Jhang, and S. Lin. 2018. Design of a wireless sensor network-based IoT platform for wide area and heterogeneous applications. IEEE Sensors J. 18, 12 (2018), 5187−5197.*

6. *C. E. Rasmussen and C. Williams. 2006. Gaussian Processes for Machine Learning. MIT Press, Cambridge, MA.*