Chapter 5

Early Classification Approach for Multivariate Time Series with Unseen Class

In the previous chapters, we addressed the early classification problem for the MTS with the components of different sampling rate and faulty components. These approaches assumed that the new incomplete MTS (to be classified) should belong to only seen classes. However, in some applications such as appliance monitoring, early classification of the MTS of unseen class is desirable for fault identification. In this chapter, we present an early classification approach for identifying the class label of an incomplete MTS even when it belongs to an unseen class.

5.1 Introduction

Traditional early classification approaches can predict the class label only if it is associated to some training instances. It essentially means that these approaches can not identify an unknown (or unseen) class label. However, in some industrial or domestic applications, early classification of unseen faults (essentially the class labels) is desirable

for making the machines smart and robust. It can provide sufficient time to prevent the propagation of effects on other parts of the appliance [89]. The appliances with such early classification can remarkably save the maintenance cost by giving early warning. An unseen class (fault) can be classified with the help of MTS of seen classes (faults) using a concept of Zero-Shot Learning (ZSL) [38].

With the advent of sensors, it becomes easy to diagnose the abnormal or faulty behavior of the appliances by using the sensory data. Faulty operation of an appliance causes a significant fluctuations in the sensory measurements. Such fluctuations help to distinguish a faulty operation from the normal [39, 40]. The sensors generate an MTS corresponding an operation of the appliance, which should be classified to identify the type of fault.

This chapter addresses the problem: *how to classify an incomplete MTS, associated with a seen or unseen class label, by using a labeled training dataset?* To solve this problem, this work proposes a **S**emantic-information based **E**arly **C**lassification approach for **M**TS (SECM). Unlike existing work, SECM is capable of identifying an unseen class by utilizing the semantic information of the seen classes.

5.1.1 Motivation

While addressing the limitations observed in the existing work [22, 26, 29], this work advances an early classification approach by making it able to classify an MTS of an unseen class label. The major limitations in the existing work are as follows.

- The existing early classification approaches [22, 26, 29] can predict only a seen class label while classifying an incomplete MTS. As it is not practical to have prior knowledge or training MTS instances of every class label, these approaches can not be adapted for fault classification in the industry or domestic appliances as an unknown fault may occur any time.
- In previous studies [38, 90, 91], the authors either construct a human-annotated

attribute matrix or use word vector space for predicting unseen classes, which require domain expertise. In sensory data, word vector may consist of a fixed set of attributes (sequential patterns) that can be used in identifying class labels. Such requirement can be eliminated by learning the attributes automatically from the training data.

- Correlation between the components of MTS provides a crucial information for its early classification. Some prior work [26, 31, 32] did not consider such correlation during classification.

5.1.2 Major contributions

This chapter makes following major contributions:

- This work proposes an early classification approach, SECM, to classify an incomplete MTS even if it belongs to an unseen class label. We incorporate an idea of ZSL in SECM to predict the unseen class by using semantic information of the seen classes.
- We develop an attribute learning model to obtain most discriminative attributes corresponding to the seen classes. The attributes capture the semantic information of the seen classes. SECM utilizes these attributes for followings: 1) to learn an MRL for each seen class separately using training dataset and 2) to identify an unseen class.
- SECM uses a correlation-based similarity measure to compute the distance between the MTS. As this similarity measure utilizes the correlation between the components of MTS, it helps to obtain better MRLs for a desired accuracy of the classification.

The rest of the chapter is organized as follows. Next section presents the preliminaries of the chapter. Section 5.3 proposes an early classification approach, SECM, for MTS of seen and unseen classes. In Section 5.4, we discuss a case study for fault classification in

the washing machines using SECM. Next, performance evaluation of SECM is carried out on existing datasets in Section 5.5. Finally, Section 5.6 concludes the chapter.

5.2 Preliminaries

This section describes the terminologies and notations used in this work. Let $\mathbf{D} = \{\langle \mathbf{C}^j, \mathcal{L}^j \rangle : 1 \leq j \leq N\}$ be a labeled training dataset where $\mathbf{X}^j$ denotes an instance of MTS and $\mathcal{L}_j$ denotes its class label. The class label $\mathcal{L}^j$ can belong to one of the l classes as $\mathcal{L}^j \in \{L_1, L_2, \cdots, L_l\}$. The set of all possible classes in $\mathbf{D}$ is called as seen classes, denoted by $\mathbf{L} = \{L_1, L_2, \cdots, L_l\}$. Further, the class labels that are not in $\mathbf{D}$ are called as unseen classes.

Definition 5.1 (Attribute) *An attribute is a most discriminative subsequence of a time series (component of MTS) which helps in distinguishing the time series of one class from others. It captures the semantic information of the class label to which the time series belongs.*

Definition 5.2 (Smart appliance) *An appliance is said to be smart if it has ability to monitor and control its operational behavior automatically by using real-time measurements of embedded sensors.*

• **Zero-Shot Learning (ZSL):** Traditional classifier learns a mapping function $f : X \rightarrow Y$ using a given labeled training dataset, where X and Y denote input feature space and output class space, respectively. The function f can predict the most likely class label $y \in Y$ for a test input feature vector $x' \notin X$, that means it is able to predict only the seen classes which are occurred in training dataset. The ZSL [38] extends the prediction capability of the traditional classifiers for unseen classes. It utilizes semantic information of the seen classes to recognize the unseen classes. Such semantic information is inferred from the input instances. The ZSL transforms input feature vectors to semantic attributes and builds an attribute space $\mathcal{A}$. Then, the objective

of the classifier is to learn a mapping function $f' : \mathcal{A} \rightarrow Y$. Figure 5.1 illustrates the traditional classifier with ZSL for time series classification.

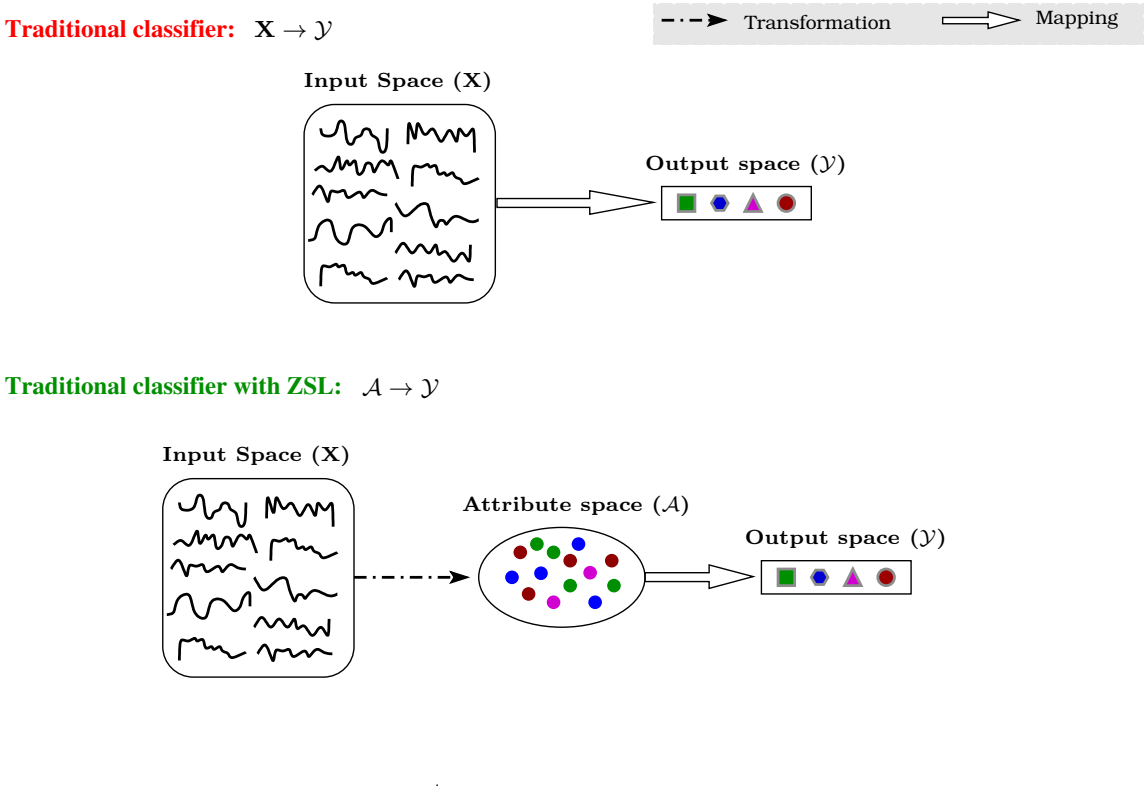


Figure 5.1: Overview of the traditional classifier with ZSL.

5.3 Semantic-Information based Early Classification Approach for MTS

This section proposes an early classification approach SECM to classify an incomplete MTS using semantic information of the target classes. Unlike traditional early classifiers [22, 25, 80], SECM is capable of classifying an MTS even if it belongs to an unseen class. SECM utilizes semantic information of the seen classes for classifying the MTS of unseen class. The approach consists of mainly four steps as shown in Figure 5.2. For a given dataset $\mathbf{D}$, SECM first learns the attributes corresponding to each class label for each component of the MTS separately. Next, a distance threshold is computed corre-

sponding to the obtained attributes. SECM later utilizes the attributes for computing class-wise MRLs. Finally, the attributes and MRLs are used to predict the class label of an incomplete MTS. Algorithm 5.1 illustrates all the steps of SECM.

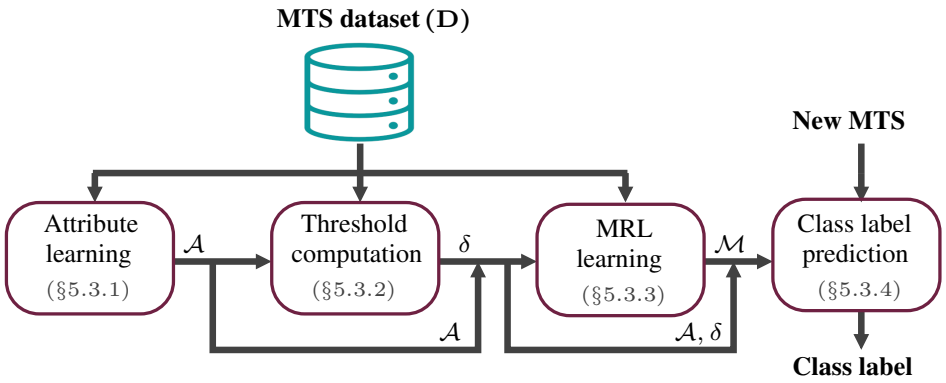


Figure 5.2: Block diagram of SECM.

5.3.1 Attribute learning

In this section, SECM uses the given MTS dataset **D** to obtain semantic information of seen classes. Such semantic information is referred to attributes of the MTS. SECM develops an attribute learning model to learn most discriminative attributes of the seen classes. It finds q attributes for each of the l seen classes from each component of the MTS. These attributes (or subsequences) are interpretable as they are the most identifiable subsequences of the time series, which capture succinct information of their class labels. SECM learns these attributes by optimizing the loss function with maximum information gain. SECM considers each component separately for attribute learning.

Let $\mathcal{D} = \{\langle C^j, \mathcal{L}^j \rangle : 1 \leq j \leq N\}$ be a dataset that contains N labeled time series of a single component of the given dataset **D** and $\mathcal{L}^j \in \{L_1, L_2, \cdots, L_l\}$. SECM learns q attributes for each class labels. As there are l class labels in the given dataset, total $l \times q$ attributes should be learned for $\mathcal{D}$. Let $\mathcal{Z}$ denotes a set of $l \times q$ subsequences from set of all possible subsequences of $\mathcal{D}$. The subsequences are obtained using a sliding window of length W, which provides total $\mathcal{J} = M - W + 1$ subsequences in a time

series of length M. SECM first randomly selects $|\mathcal{Z}|$ subsequences and utilizes them to transform the dataset $\mathcal{D} \in \mathbb{R}^{N \times M}$ into a matrix $\mathcal{T} \in \mathbb{R}^{N \times |\mathcal{Z}|}$. The matrix $\mathcal{T}$ contains the minimum distance information from each time series of $\mathcal{D}$ to each of the selected subsequences in $\mathcal{Z}$. The minimum distance between a time series $C^j \in \mathcal{D}$ and a k^{th} selected subsequence $\mathcal{Z}_k \in \mathcal{Z}$, is calculated as

$$\mathcal{T}_{j,k} = d_{min}(C^j, \mathcal{Z}_k) = \frac{\sum_{a=1}^{\mathcal{J}} d(C^j, \mathcal{Z}_k, a)\mathbf{e}^{d(C^j, \mathcal{Z}_k, a)}}{\sum_{a=1}^{\mathcal{J}} \mathbf{e}^{d(C^j, \mathcal{Z}_k, a)}}, \tag{5.1}$$

$$\text{where, } d(C^j, \mathcal{Z}_k, a) = \sqrt{\frac{1}{\mathcal{L}} \sum_{b=1}^{\mathcal{L}} (C^j[a+b-1] - \mathcal{Z}_k[b])^2}. \tag{5.2}$$

Equation 5.2 computes Euclidean distance between C^j and $\mathcal{Z}_k$ as it is a computationally efficient and commonly used similarity measure for univariate time series [92]. Next, we define an attribute learning model using $\mathcal{T}$ as follows

$$\hat{\mathcal{L}}^{j,L_c} = \mathcal{B}_c + \sum_{a=1}^{|\mathcal{Z}|} \mathcal{T}_{j,a} \cdot \theta_{c,a}, \quad 1 \leq j \leq N,\ 1 \leq c \leq l, \tag{5.3}$$

where, θ denotes a weight parameter matrix of size $l \times |\mathcal{Z}|$ which needs to be learned using $\mathcal{T}$ and $\mathcal{B}$ denotes a vector of bias terms for l classes. The model then uses a softmax function [93] to get the probability distribution over the classes from $\hat{\mathcal{L}}^{j,L_c}$, which is given as

$$p(\hat{\mathcal{L}}^{j,L_c}) = softmax(\hat{\mathcal{L}}^j) = \frac{\mathbf{e}^{\hat{\mathcal{L}}^{j,L_c}}}{\sum_{c=1}^{l} \mathbf{e}^{\hat{\mathcal{L}}^{j,L_c}}}. \tag{5.4}$$

In order to learn the parameters of the model (given in Equation 5.3), we define a loss

function for j^{th} instance of $\mathcal{T}$ as

$$Loss(\mathcal{L}^j, \hat{\mathcal{L}}^j) = -\sum_{c=1}^{l} \mathcal{L}^{j,L_c} \cdot log(p(\hat{\mathcal{L}}^{j,L_c})) + \frac{\Lambda}{N}||\theta||_2^2, \tag{5.5}$$

where, $||\theta||_2^2$ is a regularization term to avoid overfitting of the model and Λ controls the influence of the regularization. As $\mathcal{L}^{j,L_c} = 1$ if L_c is the true class label and 0 otherwise, Equation 5.5 can be written as

$$Loss(\mathcal{L}^j, \hat{\mathcal{L}}^j) = -log(p(\hat{\mathcal{L}}^{j,L_c})) + \frac{\Lambda}{N}||\theta||_2^2. \tag{5.6}$$

The main objective of the model is to minimize the loss function (given in Equation 5.6) with respect to $\mathcal{Z}$ to get $l \times q$ subsequences of $\mathcal{D}$ with maximum information gain. As the loss function is convex with respect to $\mathcal{Z}$, SECM uses Stochastic Gradient Decent method to obtain $\underset{\mathcal{Z},\theta}{argmin}\ Loss(\mathcal{L}, \hat{\mathcal{L}})$.

The gradient of x^{th} data point of a subsequence $\mathcal{Z}_k$ with respect to the time series $C^j \in \mathcal{D}$ can be computed using chain rule of derivatives, as follows

$$\frac{\partial Loss(\mathcal{L}^j, \hat{\mathcal{L}}^j)}{\partial \mathcal{Z}_{k,x}} = \frac{\partial Loss(\mathcal{L}^j, \hat{\mathcal{L}}^j)}{\partial \hat{\mathcal{L}}^j} \frac{\partial \hat{\mathcal{L}}^j}{\partial \mathcal{T}_{j,k}} \sum_{a=1}^{\mathcal{J}} \frac{\partial \mathcal{T}_{j,k}}{\partial d(C^j, \mathcal{Z}_k, a)} \frac{\partial d(C^j, \mathcal{Z}_k, a)}{\partial \mathcal{Z}_{k,x}}. \tag{5.7}$$

As each multiplicative term in Equation 5.7 is partially differentiable, it can be easily solved. Further, we can generalize Equation 5.7 for the dataset $\mathcal{D}$ to find $l \times q$ most identifiable subsequences. Figure 5.3 illustrates the attribute learning for the given dataset $\mathcal{D}$.

Learning of just $l \times q$ attributes does not ensure the balance distribution of the attributes in the classes. This work therefore uses following steps to obtain such balance distribution:

1. Learn $|\mathcal{Z}|$ attributes from the dataset $\mathcal{D}$.

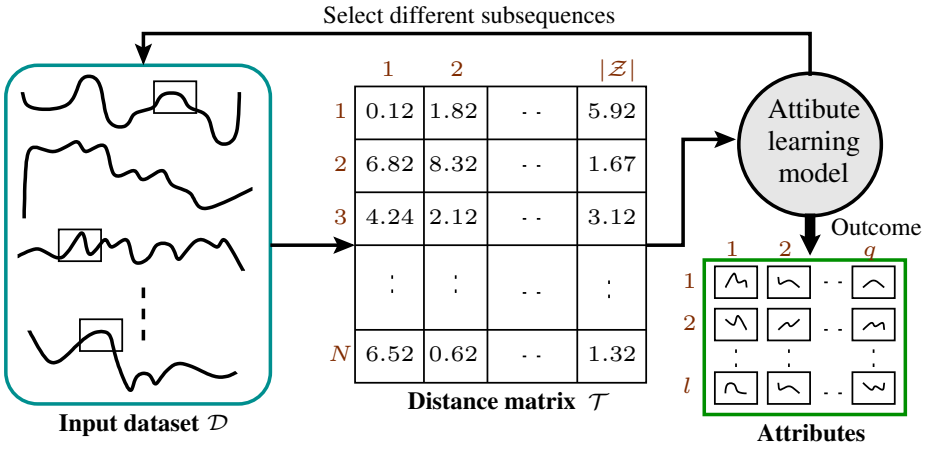


Figure 5.3: Illustration of attribute learning for given single component dataset $\mathcal{D}$.

2. If any of the l classes has less than q attributes in $\mathcal{Z}$ then increment $|\mathcal{Z}|$ by one and goto step 1.
3. As $\mathcal{Z}$ may contain more than $l \times q$ attributes, SECM chooses q attributes for each class from $\mathcal{Z}$ and thus provides final set of $l \times q$ attributes with balance distribution.

As SECM learns $l \times q$ attributes for the dataset $\mathcal{D}$ which consists the time series of only one component of the MTS, we obtained total $n \times l \times q$ attributes for the whole dataset $\mathbf{D}$. Let $\mathcal{A}$ be a three dimensional matrix of size $n \times l \times q$, where each element is a triplet consisting the followings: index of the time series to which the attribute belongs (denoted by j') and start and end points of the attribute. An element of $\mathcal{A}$ can therefore be denoted by $\mathcal{A}_{i,c,s} =< j', start, end >$, where $1 \leq i \leq n$, $1 \leq c \leq l$, and $1 \leq s \leq q$. The attribute learning for whole dataset $\mathbf{D}$ is shown at Lines $1-9$ in Algorithm 5.1.

5.3.2 Threshold computation for attributes

In this section, SECM learns a distance threshold for the obtained attributes $\mathcal{A}$. Let δ be three dimensional matrix of size $n \times l \times q$, where an element $\delta_{i,c,s}$ denotes a distance threshold corresponding to an attribute $\mathcal{A}_{i,c,s}$ for $1 \leq i \leq n$, $1 \leq c \leq l$, and $1 \leq s \leq q$.

SECM computes $\delta_{i,c,s}$ as the minimum distance of an attribute $\mathcal{A}_{i,c,s}$ from all the time series of i^{th} component with class label L_c. Let N_c denotes the number of time series that belong to L_c class label. Now, $\delta_{i,c,s}$ can be mathematically expressed as

$$\delta_{i,c,s} = \min_{j \in \{1,2,\cdots,N_c\}} \left\{ \frac{1}{W} \sum_{a=start}^{end} \left| \mathbf{C}_i^j[a] - \mathbf{C}_i^{j'}[a] \right| \right\}, \tag{5.8}$$

where, $\mathbf{C}_i^{j'}$ is the time series to which the attribute belongs. The distance threshold matrix δ helps to find best matching attributes of the seen classes during the classification of an MTS of unseen class.

5.3.3 MRL learning for early classification

As this work focuses on the early classification, SECM needs to learn MRL of the time series which would be sufficient to provide a desired accuracy ($\boldsymbol{\alpha}$). Since Euclidean-based distance can not capture the correlation among the components of MTS, this work proposes a correlation-based similarity measure which helps to gain better sense of distance than simple Euclidean-based distance. SECM first transforms each MTS of the given dataset into a correlation matrix, by exploiting the relationship between the components of MTS. Such correlation matrices are then used to find similarity between the MTS. Finally, SECM formulates a stopping condition based on the obtained accuracies of classification using full length and MRL of MTS.

Let C_a and C_b are two components of an MTS of the given dataset $\mathbf{D}$, where $a \neq b$ and number of data points in both the components is M. The correlation coefficient between C_a and C_b is defined as

$$corr(C_a, C_b) = \frac{\sum_{t=1}^{M} (C_a[t] - \mu_{C_a})(C_b[t] - \mu_{C_b})}{(M-1)\sigma_{C_a}\sigma_{C_b}}, \tag{5.9}$$

where, μ and σ denote mean and variance of the component. As the MTS consists n components, the size of corresponding correlation matrix is $n \times n$, where each element

is a correlation coefficient computed using Equation 5.9. Let $\boldsymbol{\rho}$ denotes a set of N correlation matrices corresponding to N MTS of the dataset $\mathbf{D}$, which is given as $\boldsymbol{\rho} = \{\rho^j | 1 \leq j \leq N\}$ where $\rho^j \in \mathbb{R}^{n \times n}$. Now, we define a similarity measure between two MTS $\mathbf{C}^a$ and $\mathbf{C}^b$ as follows

$$Sim(\mathbf{C}^a, \mathbf{C}^b) = \sqrt{\frac{1}{n \times n} \sum_{i=1}^{n} \sum_{i'=1}^{n} (\rho^a[i, i'] - \rho^b[i, i'])^2}. \tag{5.10}$$

SECM constructs a similarity matrix $\mathcal{S}$ of size $N \times N$ which consists of similarity scores (computed using Equation 5.10) between each pair of MTS.

Given the matrices $\mathcal{A}$, δ, and $\mathcal{S}$, for the dataset $\mathbf{D}$, the following steps illustrate the procedure for learning the MRL:

1. Find a representative MTS for each class label using δ. Let $\mathbf{C}^{j'}$ denotes a representative MTS for class label L_c. The index j' is obtained as

$$j' = \underset{j \in \{1,2,\cdots,N_c\}}{\operatorname{argmin}} \{\delta_{i,c,s}\}, \tag{5.11}$$

   where, $1 \leq i \leq n$ and $1 \leq s \leq q$.

2. Find a start index using attribute matrix $\mathcal{A}$. The start index helps SECM to know the right time for learning the MRL by considering the sufficient number of data points. It is computed as $st\_ind = \min\{start\}$ where $start$ is a vector of start indices (*i.e.,* $< \cdot, start, \cdot >$) of all the attributes in $\mathcal{A}$.

3. This step computes the classification accuracy of the SECM on the training dataset $\mathbf{D}$ by using the similarity matrix $\mathcal{S}$. The SECM first compares each MTS with every class representative MTS based on their similarity scores in $\mathcal{S}$, to find its nearest representative MTS. Next, the class label of the nearest representative MTS is assigned to the MTS. Later, the classification accuracy for each class is computed, which is denoted by $\mathbf{A}_{M,c}$ for class label L_c where M is the

number of data points of the MTS that are used in computation of $\mathcal{S}$.

4. For $t = st\_ind$ to M:
   - Compute accuracy $\mathbf{A}_{t,c}$ using first t data points of the MTS as discussed in Step 3.
   - If condition $\boldsymbol{\alpha} \times \mathbf{A}_{M,c} \leq \mathbf{A}_{t,c}$ satisfies then value of t becomes the MRL for class label L_c, where $1 \leq c \leq l$.

The above four steps provides an MRL corresponding to each of l class labels. SECM builds a set of these obtained MRLs as $\mathcal{M} = \{MRL_1, MRL_2, \cdots, MRL_l\}$. Algorithm 5.1 shows the steps of MRL leaning at Lines $11 - 15$.

5.3.4 Class label prediction using ZSL

In this work, we employ ZSL method to predict the class label of an incomplete MTS. The ZSL uses attribute matrix $\mathcal{A}$ of the seen classes while classifying the MTS. SECM waits for arrival of sufficient data points (*i.e.,* MRL) in MTS before starting the prediction. Once the MRL is obtained, the ZSL finds the distances between a component of MTS and its corresponding attributes in $\mathcal{A}$. Such distances are then compared with the precomputed thresholds (*i.e.,* δ) of the attributes to obtain a single nearest attribute for each component of the MTS. Finally, if the nearest attributes for all the components belong to the same class label then it is assigned to the incomplete MTS, otherwise an unseen class label is assigned to the MTS.

Let $\mathbf{C}^p$ be an incomplete MTS with n components, which is to be classified. SECM starts prediction as soon as t' data points are arrived in $\mathbf{C}^p$, where $t' = \min\{\mathcal{M}\}$. Let $\mathcal{L}^p$ is the class label for which the MRL is minimum. The ZSL computes distance between a component $\mathbf{C}_i^p$ of $\mathbf{C}^p$ and an attribute $\mathcal{A}_{i,c,s} =< j', start, end >$, as follows

$$dist(\mathbf{C}_i^p, \mathcal{A}_{i,c,s}) = \min_{a \in \{1,2,\cdots,t'-W+1\}} \{d(\mathbf{C}_i^p, \mathcal{Z}_{j'}, a)\}, \tag{5.12}$$

where, $\mathcal{Z}_{j'} = \mathbf{C}_i^{j'}[start : end]$ and $d(\mathbf{C}_i^p, \mathcal{Z}_{j'}, a)$ can computed using Equation 5.2. Now,

Algorithm 5.1: SECM

Input: A dataset $\mathbf{D}$ consists of N labeled MTS instances. The MTS has n components with M data points in each. An incomplete $\mathbf{C}^p \notin \mathbf{D}$ for which label is to be predicted;

Output: Attribute matrix ($\mathcal{A}$), threshold matrix (δ), set of MRLs ($\mathcal{M}$), and predicted class label $\mathcal{L}^p$;

/* Attribute learning */

1 **for** $i \leftarrow 1$ *to* n **do**

  /* $\mathcal{D}$ consists all time series of a single component */

2 Obtain a set $\mathcal{Z}$ of $(l \times q)$ subsequences with length W from $\mathcal{D}$.

3 **for** $j \leftarrow 1$ *to* N **do**

    /* for each subsequence $\mathcal{Z}_k \in \mathcal{Z}$ */

4   Compute $\mathcal{T}_{j,k}$ using Equation 5.1.

5 **for** $j \leftarrow 1$ *to* N **do**

    /* for each class label L_c */

6   Compute $\hat{\mathcal{L}}^{j,L_c}$ from $\mathcal{T}$ using Equation 5.3.

7   Calculate loss $Loss(\mathcal{L}^j, \hat{\mathcal{L}}^j)$ using Equation 5.6.

8 Learn model parameters θ and $\mathcal{Z}$ by $\underset{\mathcal{Z},\theta}{argmin}\ Loss(\mathcal{L}, \hat{\mathcal{L}})$ for $\mathcal{D}$.

9 Construct attribute matrix $\mathcal{A}$ using obtained $\mathcal{Z}$ for whole dataset $\mathbf{D}$.

10 Compute a threshold matrix δ for $\mathcal{A}$ using Equation 5.8.

/* MRL learning */

11 **for** $a \leftarrow 1$ *to* N **do**

12 **for** $b \leftarrow 1$ *to* N **do**

13   Obtain similarity score $Sim(\mathbf{C}^a, \mathbf{C}^b)$ using Equation 5.10.

14 Construct a similarity matrix $\mathcal{S}$ using obtained scores for dataset $\mathbf{D}$.

/* Using $\mathcal{A}, \delta$, and $\mathcal{S}$ as shown in Steps $1-4$ of Section 5.3.3 */

15 Obtain set of MRLs as $\mathcal{M} = \{MRL_1, MRL_2, \cdots, MRL_l\}$.

/* Class label prediction on an incomplete MTS $\mathbf{C}^p$ using ZSL */

16 $t' = \min\{\mathcal{M}\}$

/* $\mathcal{L}^p$ is class label for minimum MRL in $\mathcal{M}$ */

/* Once t' data points are arrived in $\mathbf{C}^p$ */

17 **for** $i \leftarrow 1$ *to* n **do**

18 Find nearest attribute $Nearest_i$ for $\mathbf{C}^p_i$ using Equation 5.13.

19 **if** *Class label corresponding to* $Nearest_i$ *is not* $\mathcal{L}^p$ **then**

20   Wait for more data points in $\mathbf{C}^p$ and **Break**.

21 Repeat Lines $16-21$ until y_q is same for all nearest attributes.

22 **if** $\mathbf{C}^p$ *is not completed yet* **then**

23 Assign $\mathcal{L}^p$ as predicted class label to $\mathbf{C}^p$.

24 **else**

25 Assign an unseen class label to $\mathbf{C}^p$.

the nearest attribute for the component $\mathbf{C}_i^p$ is obtained as

$$Nearest_i = \underset{\mathcal{A}_{i,c,s}}{\operatorname{argmin}} \ \{dist(\mathbf{C}_i^p, \mathcal{A}_{i,c,s}) < \delta_{i,c,s}\}, \tag{5.13}$$

where, $1 \leq c \leq l$ and $1 \leq s \leq q$. Next, if class label of $Nearest_i$ for $1 \leq i \leq n$ is same and equal to $\mathcal{L}^p$ then predicted class label for $\mathbf{C}^p$ is $\mathcal{L}^p$. If class labels are same but not equal to $\mathcal{L}^p$ then SECM waits for more points till the next smallest MRL of $\mathcal{M}$ and also updates the $\mathcal{L}^p$ correspondingly. Later, if all the data points are arrived in the MTS $\mathbf{C}^p$ and the class labels of nearest attributes are still not same for the components then an unseen class label is assigned to $\mathbf{C}^p$. The steps of prediction are shown at Lines $16-25$ in Algorithm 5.1. Finally, the number of data points of the incomplete MTS that are used in the classification becomes its MRL. Such MRL is further used to compute earliness of the classification.

5.4 Case study: washing machine

This section conducts a case study for a domestic appliance, *i.e.,* washing machine. We first discuss an experimental setup to collect a labeled MTS dataset for different types of faults. Later, the performance of SECM is evaluated on the collected dataset using accuracy, earliness, and F_1 score.

5.4.1 Experimental setup

In this experiment, we use five different types of sensors to collect MTS data of various faults from washing machines. Figure 5.4 illustrates a fully-automatic top load washing machine with five sensors including accelerometer, force, gas, sound, and temperature. Each sensor is connected to a NodeMCU module (ESP8266) which transmits the data to Raspberry Pi through Wi-Fi. Sampling rate of the sensors is 10 Hz to capture subtle information of the faults. A NodeMCU with four sensors is attached to wall of inner

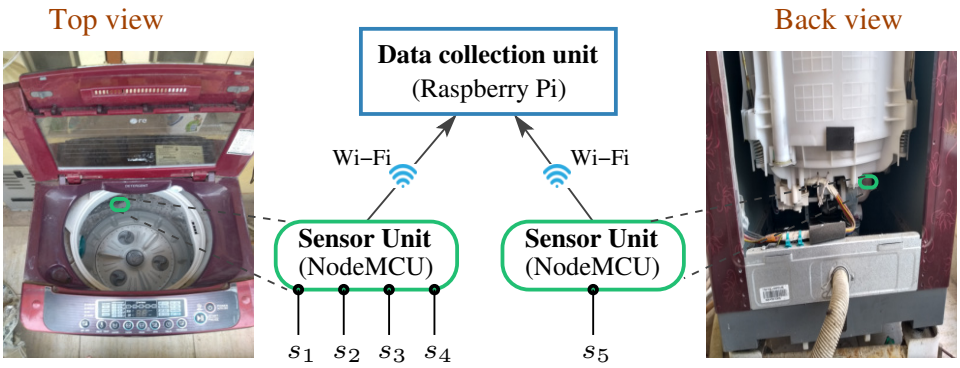


Figure 5.4: Experimental setup for data collection using NodeMCU and Raspberry Pi from washing machine. Sensor connected to NodeMCU are s_1- Accelerometer, s_2- Force, s_3- Gas, s_4- Sound, and s_5- Temperature.

drum. Accelerometer and force sensor are used to record the vibration and laundry weight for identifying the imbalanced drum, respectively. Gas and sound sensors are used to monitor the dryer for any burning smell and presence of any foreign object (*e.g.,* coin, pen, ring), respectively. Another NodeMCU with temperature sensor is attached to the motor to detect overheating problem. Apart from these sensors, a voltmeter socket is also used to supply the electricity for the washing machine and further to record its electricity consumption.

5.4.2 Data collection

We create a labeled MTS dataset by collecting the data for following faults: motor overheating (**F1**), imbalanced drum (**F2**), clogged drain pipe (**F3**), cloth burning in dryer (**F4**), and foreign object (*e.g.,* coin, pen, ring, *etc.*) with laundry (**F5**). Each of these faults corresponds to a class label in the dataset. In addition to this, the MTS data is also collected for normal or non-faulty operation (**F6**), which also adds a class label in the dataset and makes the total number of classes six *i.e.,* $l = 6$. For each single operation (faulty or normal), the sensory data is collected for 60 seconds at 10 Hz using aforementioned five sensors. The sensors together form an MTS with 600 (*i.e.,* $M = 600$) data points in each of its time series. As this work also uses a time

series of voltage readings, the formed MTS consists of six components or time series, *i.e.,* $n = 6$. This work considers four fully-automatic washing machines from Samsung and Whirlpool companies. For each of four washing machines, we collected 100 MTS for each of six class labels, which provided total $4 \times 6 \times 100 = 2400$, *i.e.,* $N = 2400$ MTS in the dataset. We call this dataset as Washing Machine Faults (WMF).

5.4.3 Results and discussions

This work divides the WMF dataset into training and testing data with 70% (*i.e.,* 1680) and 30% (*i.e.,* 720) MTS, respectively. SECM uses training data for building a classification model by learning attributes and MRLs for the available classes (*i.e.,* seen) and then evaluates the model on testing data. This work employs a standard 10-fold cross-validation method during learning. We use following evaluation metrics:

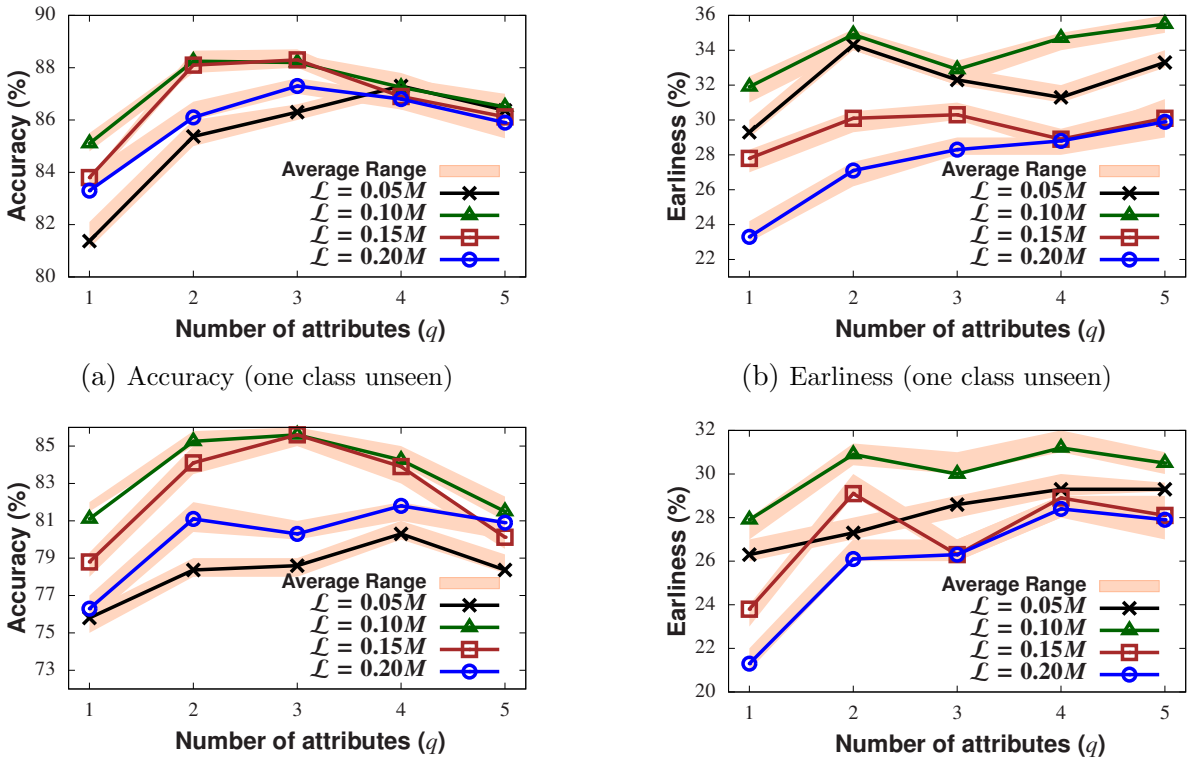
- *Accuracy:* It is the percentage of number of MTS in the testing data that are correctly classified by SECM.
- *Earliness:* It is the percentage of data points of a complete time series that are not used in the classification.
- *Precision:* It is the percentage of times that a fault is classified correctly. Precision indicates a quality aspect of classification model.
- *Recall:* It measures the completeness and relevance of the classification model. Recall is the percentage of times that a particular fault is detected by the model.
- F_1 *score:* It gives an integrated score using precision and recall as

$$\frac{2 \times precision \times recall}{precision + recall}. \tag{5.14}$$

5.4.3.1 Selection of length and number of attributes

This work first selects a suitable length of attribute that can provide desired level of accuracy with earliness using different number of attributes in the time series. Figure 5.5

shows accuracy and earliness results using different number of attributes, *i.e.,* $q = \{1, 2, 3, 4, 5\}$ at $\boldsymbol{\alpha} = 0.9$. In parts (a) and (b), the results are obtained by taking one class (*i.e.,* minimum) as unseen. Similarly, parts (c) and (d) show the results by taking four classes (*i.e.,* maximum) out of six as unseen (because at least two classes are required to build classification model). At $q = 2$, SECM obtains maximum accuracy (*i.e.,* 88.2% and 85.1%) with maximum earliness (*i.e.,* 34.9% and 31.1%) by using the attribute of length 10% of MTS (*i.e.,* $\mathcal{L} = 0.10M$), as shown by 'brown' ellipses. Though SECM gets marginal gain on accuracy at $q = 3$ for $\mathcal{L} = 0.10$ and 0.15, it loses earliness of the classification which is crucial for real-time fault identification. We therefore set $q = 2, \mathcal{L} = 0.10M$ and $\boldsymbol{\alpha} = 0.9$ for the performance evaluation of SECM.



(a) Accuracy (one class unseen)

(b) Earliness (one class unseen)

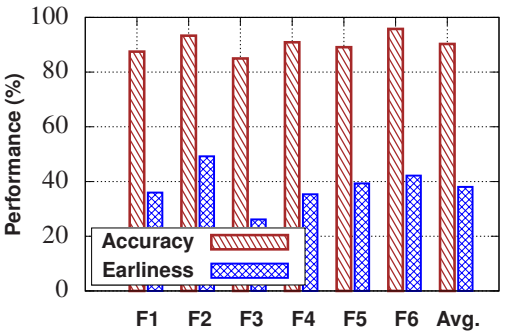
(c) Accuracy (four classes unseen)

(d) Earliness (four classes unseen)

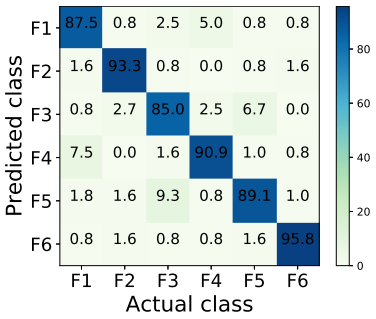
Figure 5.5: Accuracy and earliness results using varying number of attributes in the time series.

5.4.3.2 Performance of SECM for seen faults

First of all, the performance of SECM is evaluated by classifying only seen faults where classification model is built by taking the MTS of all types of faults during training. Figure 5.6 illustrates the obtained accuracy and earliness results in part (a) and confusion matrix in part (b). It is observed that SECM is able to maintain the desired level of accuracy (*i.e.,* 90%) with an average earliness of 38.8%. Earliness is maximum (*i.e.,* 49.5%) for **F2** class label, which indicates that the MTS corresponding to imbalanced drum fault has better identifiable patterns at early stage. Next, it can be observed from the confusion matrix that SECM is able to predict correct class label maximum times (*i.e.,* 95.8%) for **F6**. It indicates that the MTS of normal operation is significantly different from other faults.



(a) Accuracy and earliness.



(b) Confusion matrix (in %).

Figure 5.6: Performance of SECM for seen classes (*i.e.,* faults).

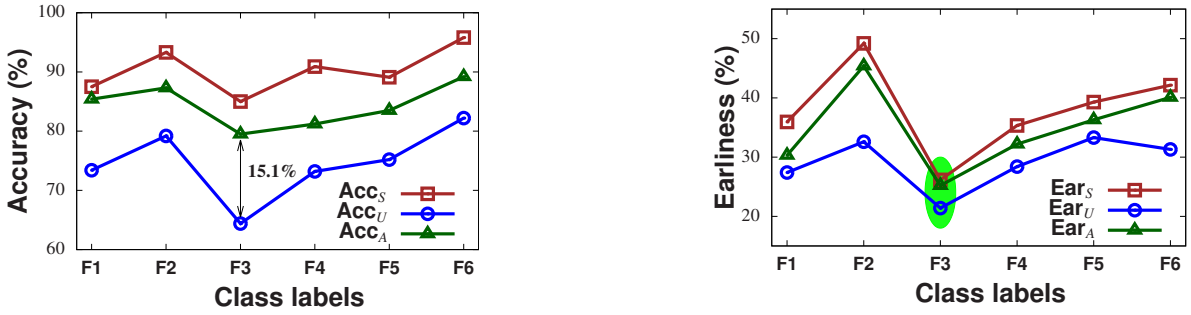
5.4.3.3 Performance of SECM for unseen faults

Next, we evaluate SECM for unseen fault classification. In Figure 5.7, the results are shown with following configurations:

- Acc_S and Ear_S: Accuracy and earliness of the class when it is seen and other classes are also seen.
- Acc_U and Ear_U: Accuracy and earliness of the class when it is unseen and other classes are seen.
- Acc_A and Ear_A: Average accuracy and earliness of all classes when the class is

unseen and others are seen.

It is observed from the figure that SECM is able to classify the unseen faults with an accuracy of $(72 \pm 8.3)\%$ and an earliness of $(27 \pm 6.7)\%$. Though SECM gets lower Acc_U and Ear_U values for all the class labels, it is able to maintain an average accuracy of $(84 \pm 3.7)\%$ with an average earliness of $(37 \pm 8.2)\%$ as shown by Acc_A and Ear_A, respectively. An interesting observation is seen for **F3** class label, where Acc_U is substantially lesser (*i.e.,* 15.1%) than Acc_A while Ear_U differs marginally from Ear_A as shown in part (b).



(a) Accuracy

(b) Earliness

Figure 5.7: Impact of unseen classes on accuracy and earliness of SECM.

Further, class-wise performance of SECM for unseen class labels are reported using precision, recall, and F_1 score as illustrated in Figure 5.8. The right most bar group shows the average performance of the six classes. We observe that SECM is able to achieve an average precision of 79.7%, recall of 75.6%, and F_1 score of 78.5%, which is not significantly higher. It is mainly due the presence of unseen class label. However, SECM is able to maintain significant level of balance between correctness and completeness of the classification model which is indicated by F_1 score.

5.4.3.4 Sensitivity analysis of SECM

Figure 5.9 illustrates the performance results of the sensitivity studies with varying number of unseen classes for different values of $\boldsymbol{\alpha}$. As the proposed approach needs minimum two classes for training the classifier, we can take maximum four classes (out

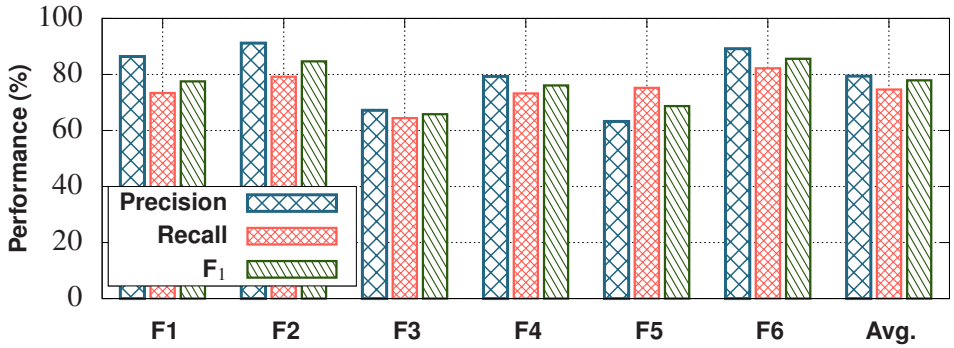
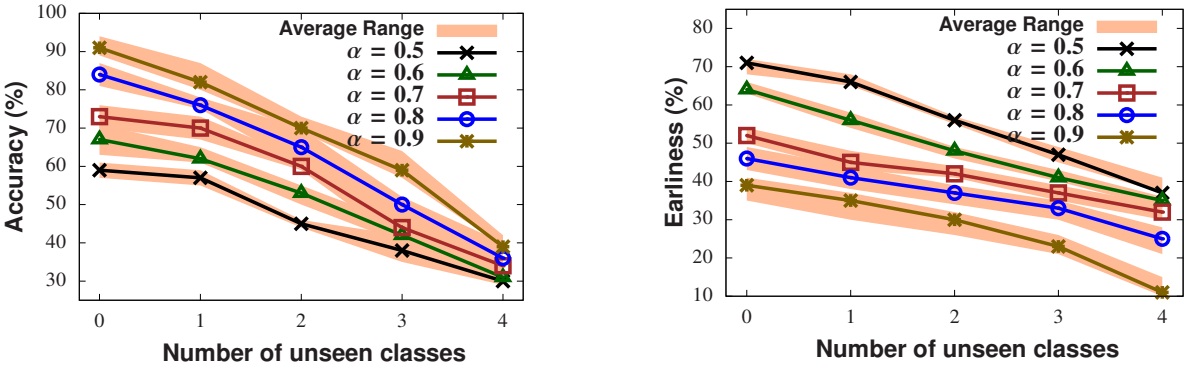


Figure 5.8: Precision, recall, and F_1 score results for unseen class labels.

of six) as unseen. We observe that the performance of SECM degrades as the number of unseen classes increases for all the values of $\boldsymbol{\alpha}$. Though SECM is very sensitive to the number of unseen classes, it is still able to achieve an accuracy of 37.2% with an earliness of 10.7% for $\boldsymbol{\alpha} = 0.9$ even when four out of six classes are unseen. Next, it is easy to observe that SECM is able to maintain the desired level of accuracy $\boldsymbol{\alpha}$ of the early classification up to one unseen class. However, such accuracy is obtained at the cost of earliness. The results clearly indicate that $\boldsymbol{\alpha}$ significantly influences the performance of SECM.



(a) Accuracy

(b) Earliness

Figure 5.9: Performance results of sensitivity analysis of SECM.

5.5 Experimental Results on Existing Datasets

This section evaluates the performance of SECM on two relevant datasets available in UCI repository [13]. We first discuss the existing datasets and then illustrate the results

on these datasets. Later, we compare the performance of SECM with three existing approaches including MSD [26], DMP+PPM [29], and MTSEC [32].

5.5.1 Existing datasets

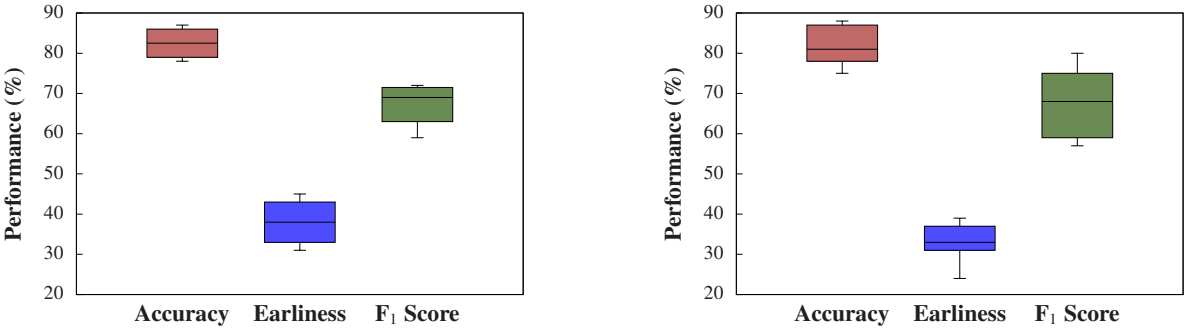
Hydraulic System Monitoring (HSM) [94] is an MTS dataset which is collected to monitor the condition of a hydraulic system using various sensors such as temperature, pressure, volume flow, *etc.* In order to monitor the status of the system, the sensors are attached to different components including accumulator, valve, cooler, and pump. Each component may undergo different types of faults. We consider only the faults related to the hydraulic accumulator. This work uses the MTS data from six pressure sensors along with one time series of motor power readings taken at 100 Hz for 60 seconds. It means there are total seven components ($n = 7$) in the MTS with 6000 data points ($M = 6000$). In HSM dataset, there are total 2205 ($N = 2205$) MTS corresponding to four faults (class labels *i.e.,* $l = 4$) in the accumulator.

GMD dataset [95] is collected to detect chemical leakage based on mixture of turbulent gases. This work uses down sampled dataset available in [13]. Eight gas sensors are used to detect the mixture, which generate the MTS with eight components ($n = 8$). As the sensors were exposed to the mixture for 300 seconds and taking readings in every 100 ms (*i.e.,* 10 Hz), the generated MTS consists of 3000 data points ($M = 3000$) in each of its component. We increase the number of MTS in the dataset by adding the Gaussian noise with the given data, which makes total 600 MTS ($N = 600$). Each MTS corresponds to one of the 30 different mixtures of gases (*i.e.,* $l = 30$).

5.5.2 Experimental results

In this section, we take accuracy, earliness, and F_1 score for assessing the performance of SECM on aforementioned existing datasets and the obtained results are shown in Figure 5.10 using box plot. As this work evaluates SECM by using different configu-

ration of seen and unseen classes, it provides a set of values for each evaluation metric (*i.e.,* accuracy, earliness, and F_1 score). Such a set is plotted using a box plot method in Figure 5.10. It can be seen from the figure that SECM is able to obtain a median accuracy of more than 80% for both the datasets with a median earliness of 38.9% for HSM and 32.7% for GMD dataset. As the number of class labels are more (*i.e.,* 30) in GMD dataset, the range is larger when compared to HSM dataset. Further, obtained F_1 scores indicate that SECM performs good at balancing the precision and recall. In other words, the performance of SECM for unseen class is as good as for seen classes.



(a) HSM dataset (b) GMD dataset

Figure 5.10: Performance results of SECM on the existing datasets.

5.5.3 Comparison with existing approaches

The performance of SECM is compared with three existing approaches using accuracy, earliness, and F_1 score, as illustrated in Table 5.1. As the existing approaches do not classify the MTS of unseen class, this work uses two variants of SECM (SECM_v_1 and SECM_v_2) for comparison. SECM_v_1 considers the classification of MTS for seen classes only while SECM_v_2 classifies the MTS of both seen and unseen classes. SECM_v_2 considers one unseen class at a time. It is observed that SECM (both variants) outperforms the existing approaches on different evaluation metrics for all the datasets. With a small compromise of accuracy and earliness, SECM_v_2 becomes capable enough to classify the MTS of unseen classes. Though MTSEC performs significantly good on accuracy with a marginal difference of around 2% compared to the accuracy of SECM_v_2, it substan-

tially loses on earliness with around 17%. Further, DMP+PPM performs better than MTSEC on earliness but worse on accuracy. It indicates that deep learning method, which is used in MTSEC, helps to increase the accuracy but not the earliness.

Table 5.1: Comparison of SECM with existing approaches using different datasets.

	WMF dataset			**HSM dataset**			**GMD dataset**		
	Accuracy	Earliness (MRL)	F_1 score	Accuracy	Earliness (MRL)	F_1 score	Accuracy	Earliness (MRL)	F_1 score
MSD [26]	79.21%	12.1% (527)	0.41	81.50%	18.2% (4908)	0.46	78.31%	15.7% (2529)	0.55
DMP+PPM [29]	78.21%	26.3% (442)	0.39	73.1%	30.2% (4188)	0.44	76.31%	28.2% (2154)	0.51
MTSEC [32]	85.21%	20.0% (480)	0.59	83.23%	22.2% (4668)	0.56	86.13%	19.2% (2424)	0.61
SECM_v_1	**90.49%**	**38.9% (366)**	0.71	**92.3%**	**43.3% (3401)**	0.75	**90.75%**	**40.2% (1794)**	0.79
SECM_v_2	**87.37%**	**36.2% (383)**	0.69	**85.95%**	**41.3% (3522)**	0.65	**86.15%**	**37.2% (1884)**	0.72

5.6 Conclusion

This chapter proposed an early classification approach, SECM, to classify an incomplete MTS. Unlike existing approaches, the proposed approach is capable enough to classify the MTS of unseen class label by using the semantic information of the seen classes. Such semantic information is extracted in the form of most distinctive subsequences (attributes) of the time series. We developed an attribute learning model to obtain best set of attributes for each seen class label and utilized them for learning MRLs to achieve earliness in the classification. Various experiments are carried to evaluate the SECM on a collected dataset and two existing datasets. The experimental results showed that SECM performs significantly well for unseen classes, with a marginal compromise of accuracy and earliness. This work also motivates further research towards applying deep learning methods in the early classification of MTS with unseen class.

Publication

- **Ashish Gupta**, H. P. Gupta, B. Biswas, and T. Dutta, "An Unseen Fault Classification Approach for Smart Appliances using Ongoing Multivariate Time Series," *IEEE Transactions on Industrial Informatics*, pp. 1-8, 2020.