Chapter 3

Early Classification Approach for Multivariate Time Series with Sensors of Different Sampling Rate

In this chapter, we focus on to develop an early prediction approach to predict a class label of an incomplete MTS by using a given labeled MTS dataset. We consider that the MTS is generated from the sensors of different sampling rate, and thus it contains unequal number of samples in its components.

3.1 Introduction

With the recent advancements in the technologies, sensor-based devices have become an integral part of our life. These devices generate MTS data, which requires a classification approach to classify it in real-time. However, the challenge is to maintain a desired level of accuracy by using only fewer data points of the MTS. In the era of Internet of Things (IoT), the sensors can have *different sampling rate* in order to capture finer and non-redundant measurements about the phenomenon being observed. As a result, the generated MTS contains different number of samples in its components. An

important example of having such MTS is autonomous vehicle which is equipped with several sensors including accelerometer, gyroscope, temperature, light, and so on. These sensors help to monitor the inside and outside environment of the vehicle, behavior of driver [71], road environment [72], and traffic flow prediction [73].

Figure 3.1 illustrates an example scenario, where on-vehicle sensors are used to classify the outside environment on the road, such as fog, to turn on the fog lights of the vehicle. The sensors have different sampling rate. As early as the classifier classifies the road as foggy using the MTS (generated by the sensors), the fog lights will get turned on and the driver will also be notified. Early classification of the MTS is therefore helping to avoid accidents that could occur due to poor visibility on the road.

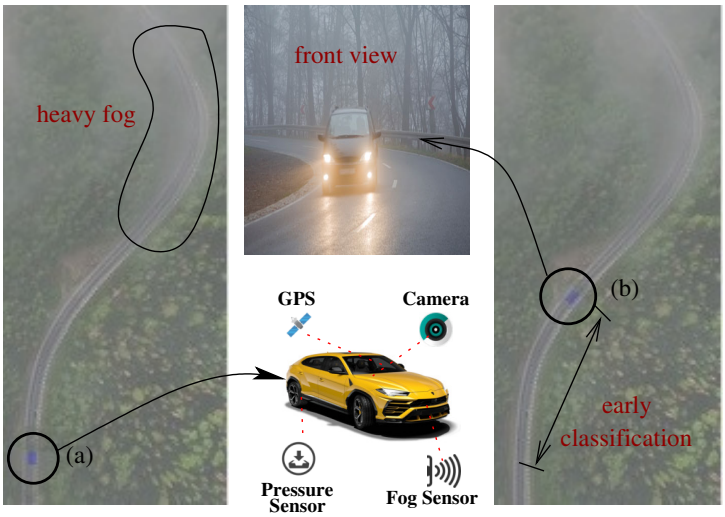


Figure 3.1: Illustration of the early classification for detection of the fog status on the road. Point (a) shows an example, where the sensors equipped vehicle is approaching a turn with heavy fog on the road. The fog lights of the vehicle get turned on as it reaches near the turn, as shown by point (b).

As the MTS contains different length of components, its early classification becomes a challenging task. We therefore focus on to solve the early classification problem for such MTS by using a labeled training dataset. Here, the main problem is to estimate the class-wise MRLs using training MTS dataset with the desired level of accuracy (denoted by $\boldsymbol{\alpha}$) with $0 < \boldsymbol{\alpha} \leq 1$.

3.1.1 Motivation

Literature indicates following limitations that motivated the work in this chapter.

- The existing work [22, 26, 28, 29] assume that the MTS contains only the components of equal length which are generated by using the sensors of *equal sampling rate*. In real-world applications such as intelligent transportation system, different types of sensors are used to observe various environmental conditions such as humidity, light intensity, temperature, *etc.* Since the different sensors measure the changes in the different environmental conditions, it is inappropriate to record such changes using the sensors of equal sampling rate.
- Another major limitation of the existing work [23, 26, 29] is that they do not use *correlation* between the components of the MTS. In autonomous vehicles, different types of sensor generate an MTS where components are correlated to each other. Avoiding such correlation reduces the accuracy of the classifier.

Considering these limitations in the literature, this chapter proposes an early classification approach for MTS and demonstrates its utility in road surface classification.

3.1.2 Major Contributions of the Work

The major contributions of this chapter are as follows.

- This chapter proposes an early classification approach, which builds an ensemble classifier, to solve the early classification problem. It first estimates the MRL of each component of MTS by using a GP classifier [42]. Later, the estimated MRLs are used to classify an incomplete MTS.
- We utilize the correlation among the components to compute the class-wise MRLs. The correlation influences the accuracy of the classification. The proposed approach predicts the class label of a time series and forwards it to a next classifier to predict the class label of the next time series of the same MTS, and so on.
- Next, the proposed approach uses a variable $\boldsymbol{\alpha}$, which helps to provide the earliness

in the classification. This variable is considered during the estimation of the MRL of the time series while building the classifier.

- Finally, we demonstrate an application of road surface classification using on-vehicle sensors with different sampling rate. The performance of the ensemble classifier is also evaluated on real-world existing datasets [13]. This work considers accuracy, earliness, and F_1 score as evaluation metrics to compare the proposed approach with existing approaches [25, 26, 29].

The rest of the chapter is organized as follows: Next section states the assumptions and defines the terms used in this work. Section 4.3 presents an early classification approach for MTS. Next, the performance of the proposed approach is evaluated for road surface classification in Section 3.4 and for existing datasets in Section 3.5. Finally, Section 4.5 concludes this chapter.

3.2 Preliminaries and Problem Statement

In this section, we state the assumptions made about the early classification and define the terms. We also define the problem statement with overview of the solution approach.

3.2.1 Preliminaries

This work considers a labeled dataset, denoted by $\mathbf{D}$, which consists N number of MTS. The dataset has l class labels. The class label of an MTS is denoted by L_i, where $1 \leq i \leq l$. The number of MTS of L_i class label is denoted by N_i, where $1 \leq i \leq l$ and $N = N_1 + N_2 + \cdots + N_l$. Each MTS in dataset $\mathbf{D}$ consists n time series (components). Each MTS in $\mathbf{D}$ is the series of data points indexed in time order. The dataset with N time series of $\mathbf{C}_i$ component is denoted by $\mathbf{D}_i$, where $1 \leq i \leq n$. This work considers following assumptions or constraints:

- The dataset $\mathbf{D}$ should have sufficient data points in MTS (*i.e.,* complete MTS) during building of the classifier.

- Sampling rate of the sensors, that generated the time series of the MTS, should be fixed (not dynamic) during the whole process of data collection.

Definition 3.1 (Component) *A time series in MTS is called a component. It is an ordered set of data points. Let* $\mathbf{C}$ *be an MTS that consists of* n *time series. Such* n *time series are called as* n *components of the MTS. The* i^{th} *component of* $\mathbf{C}$ *is denoted by* $\mathbf{C}_i$*, where* $1 \leq i \leq n$*.*

Definition 3.2 (Complete MTS) *An MTS is said to be complete if each of its components has* M *data points, where* M *denotes the maximum number of data points (i.e., full length) in the MTS of training dataset* $\mathbf{D}$*. The* j^{th} *complete MTS of* $\mathbf{D}$ *is given as* $\mathbf{C}^j = \{\mathbf{C}_1^j, \mathbf{C}_2^j, \cdots, \mathbf{C}_n^j\}$ *where* $\mathbf{C}_i^j \in \mathbb{R}^M$ *for* $1 \leq i \leq n$ *and* $\mathbf{C}^j \in \mathbb{R}^{n \times M}$*. A complete MTS can be achieved after the complete execution of the human activity.*

Definition 3.3 (MTS classification) *For a given training dataset* $\mathbf{D}$ *with* l *class labels* $\mathbf{L} = L_1, L_2, \cdots, L_l$*, the task of MTS classification is to learn a mapping function* $h : \mathbb{R}^{n \times M} \rightarrow \mathcal{L}$*, where* $\mathcal{L} \in \mathbf{L}$*.*

Definition 3.4 (Early classification of MTS) *The task of early classification of MTS is to estimate a class discriminating MRL using given training dataset* $\mathbf{D}$ *while maintaining a desired level of accuracy* $\boldsymbol{\alpha}$*. Mathematically, it is defined as* $h_f : \mathbb{R}^{n \times f} \rightarrow \mathcal{L}$*, where* $\mathcal{L} \in L$ *and* $f \leq M$*.*

Definition 3.5 (Earliness) *As early classification model does not wait for the completion of MTS, earliness comes into existence. Such earliness is defined as the number of data points (in percent) that are not used in the prediction. It is calculated as*

$$\mathcal{E}(\%) = \frac{M - f}{M} \times 100, \tag{3.1}$$

where, M *denotes the length of complete activity and* f *denotes the number of data points of the ongoing activity (i.e., new MTS) that are used in the classification.*

Definition 3.6 (Accuracy) *The accuracy of classification is defined as percentage ratio of number of correctly classified MTS (denoted by N^p) to the total number of MTS in dataset* $\mathbf{D}$ *(denoted by N). It is mathematically given as*

$$\mathcal{A}_{\mathbf{D}}(\%) = \frac{N^p}{N} \times 100. \tag{3.2}$$

• **MTS with components of Different sampling rate (MTD):** The sampling rate of a sensor is the number of data points taken per unit time. This work considers that the components of an MTS in the given dataset $\mathbf{D}$ may not have the equal sampling rate. Such MTS is called as Multivariate Time series with components of Different sampling rate (MTD). Let λ_i be the sampling rate of the component $\mathbf{C}_i$, where $1 \leq i \leq n$. The total data points taken or the length of $\mathbf{C}_i$ at time instance t is $\lambda_i \times t$. The components are arranged in decreasing order of their sampling rate as $\lambda_1 \geq \lambda_2 \geq \ldots, \geq \lambda_n$.

Figure 3.2 illustrates an example scenario of an MTD with n components in the dataset $\mathbf{D}$. Each component consists of M data points in the MTD and is arranged from top to bottom in decreasing of their sampling rates. Part (a) of Fig. 3.2 illustrates that the sampling rate of $\mathbf{C}_1$ component is more than the $\mathbf{C}_2$ component which are generated by using gas sensor, temperature sensor, and so on. Part (b) shows that at time instance t, the number of data points collected by different components, are not same due to different sampling rate. Let m_i denotes the number of data points taken by the component $\mathbf{C}_i$ till time instance t, at sampling rate λ_i. We consider that the component $\mathbf{C}_1$ consists of M data points at time instance t, *i.e.,* $m_1 = \lambda_1 \times t = M$. However, the data points in $\mathbf{C}_2$ is $m_2 = \lambda_2 \times t = M\frac{\lambda_2}{\lambda_1}$, where $m_2 < m_1$ if $\lambda_2 < \lambda_1$. We denote $\frac{\lambda_2}{\lambda_1} = \lambda_1^2$. The number of data points in $\mathbf{C}_2$ is $M\lambda_1^2$. Similarly, the component $\mathbf{C}_n$ consists the least number of data points $m_n = M\lambda_1^n$. In the MTD, sensors aggregate the temporal data at different sampling rate.

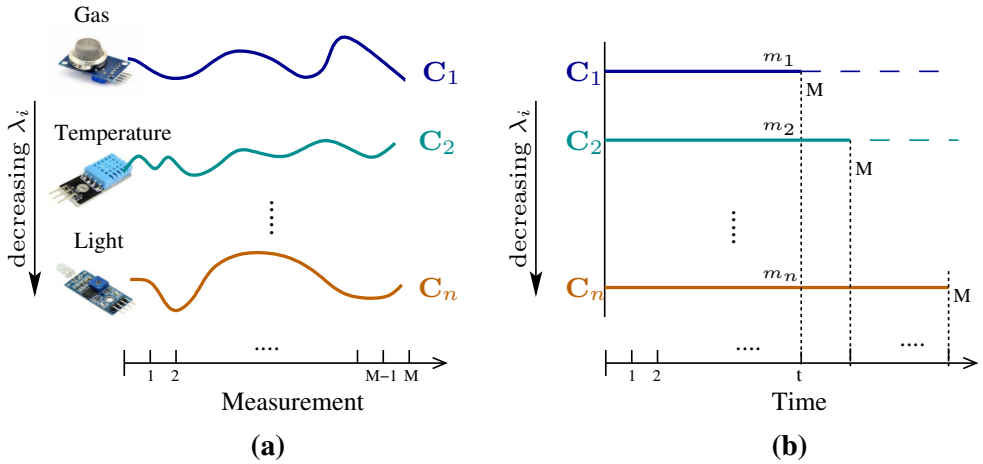


Figure 3.2: Illustration of the components of an MTD. Part (a) shows MTD with n components and each consists of M data points. Part (b) shows the length of the components at the given time instance t.

3.2.2 Problem statement and overview of the solution

The early classification approach for MTS consists of various advantages in the field of transportation system, as discussed in the introduction. The existing early classification approaches for the MTS assumed that the sampling rate of all the components are equal. However, such assumption is not suitable for real-life scenarios. This chapter therefore proposes an approach for solving the problem: how fast an approach classifies a given MTS with components of different sampling rate.

3.2.2.1 Problem statement

In this chapter, we solve the following problem: *Consider a dataset* $\mathbf{D}$ *with* N *number of MTD, where each MTD has* n *components of* M *data points. How do we predict the class label of an incomplete MTD as soon as possible with* $\boldsymbol{\alpha}$ *accuracy of the classification?*

3.2.2.2 Overview of the solution

This work proposes an early classification approach to classify an incomplete MTD with $\boldsymbol{\alpha}$ accuracy of the classifier. The proposed approach first estimates the MRL of

each component using the posterior probabilities obtained from GP classifier. Next, the approach builds a classifier $\mathbf{H}_i$ using the estimated MRL of $\mathbf{C}_i$, where $1 \leq i \leq n$. The classifier for labeling an incomplete MTD with $\boldsymbol{\alpha}$ accuracy is therefore given as $\mathbf{H} = \{\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_n\}$. Finally, a method is proposed to classify the incomplete MTD using the classifier $\mathbf{H}$.

3.3 Early Classification of MTD

In this section, we propose an approach for early classification of an incomplete MTD with desired level of accuracy $\boldsymbol{\alpha}$. The block diagram of the proposed approach with its two phases is shown in Figure 3.3.

3.3.1 Building a classifier using MRL

This step builds a classifier by using the given labeled MTS dataset $\mathbf{D}$. The input and the output of this phase are as follows:

Input: *A labeled dataset* $\mathbf{D}$ *of* N *MTS which consists* n *time series* $\mathbf{C}_i^j$ *of length* M*, where* $1 \leq i \leq n$ *and* $1 \leq j \leq N$*.*

Output: *A ensemble classifier which consists the MRLs for early classification of* $\mathbf{D}$*.*

This phase computes the MRL and builds the classifier for early classification of $\mathbf{D}$ as shown in Figure 3.3.

3.3.1.1 Compute MRL

The main focus of this step to compute the MRL of the MTD for each class with $\boldsymbol{\alpha}$ accuracy. The MRL of an MTD is effectively the MRL of the n^{th} component (*i.e.,* $\mathbf{C}_n$). The earliness is only possible when the classifier does not use the full-length time series. The proposed approach uses GP classifier for MRL estimation as it is well suited for time series modeling. It considers a time series $\mathbf{C}_i^j$ of M data points for computing the posterior class probabilities. In the proposed approach, GP uses a joint

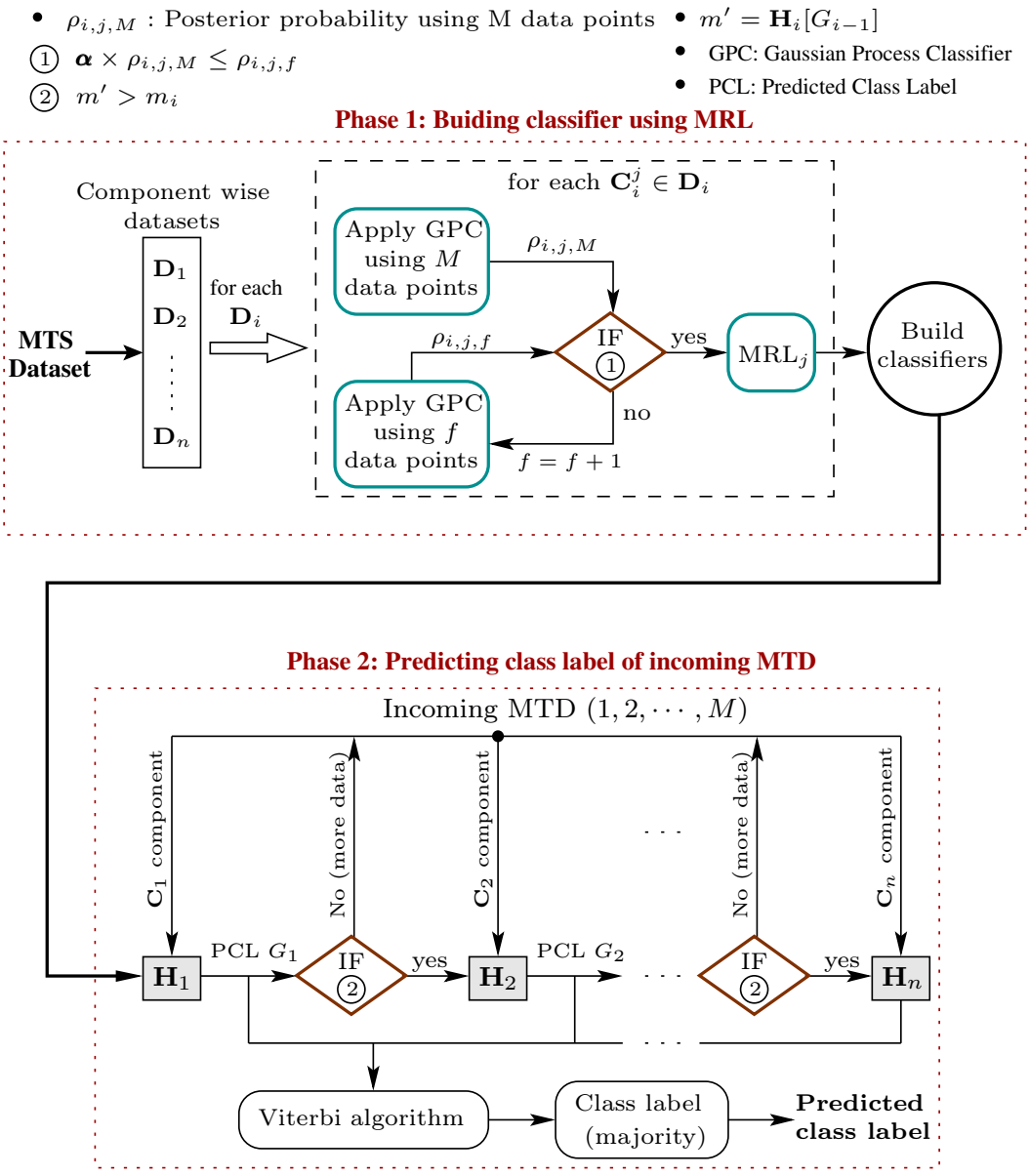


Figure 3.3: Block diagram of the proposed approach for building the classifier and predicting the class label of an incomplete MTD.

normal distribution to model a time series for computing the posterior probabilities. The posterior probability of $\mathbf{C}_i^j$ using M data points for all class labels, is given by

$$\rho_{\mathbf{i,j,M}} = \{\rho_{(i,j,M,1)}, \rho_{(i,j,M,2)}, \cdots, \rho_{(i,j,M,l)}\}. \tag{3.3}$$

Let the MRL of time series $\mathbf{C}_i^j$ is f, which is obtained as

$$\underset{f}{min}\ \mathbf{F}(f) = f, \quad s.t.\ \ \boldsymbol{\alpha} \times \rho_{i,j,M,q} \leq \rho_{i,j,f,q}, \tag{3.4}$$

where, $1 \leq f \leq M$ and $1 \leq q \leq l$. Equation 3.4 is shown as condition 1 in Figure 3.3 for the all the classes. The GP classifier is a probabilistic classifier which applies the GP prior and Bayes' rule to model the time series. Let X denotes a time series of length M as $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M\}$ and $\mathbf{x}_k \in \mathbb{R}^a$, where a represents the number of variables in a time series. GP computes the posterior probability of a time series X for class L_q using Bayes' rule as follows

$$P\left(\frac{L_q}{X}\right) = \prod_{k=1}^{M} \frac{P(\frac{\mathbf{x}_k}{L_q})P(L_q)}{P(\mathbf{x}_k)}, \tag{3.5}$$

where, $P(L_q)$ and $P(\mathbf{x}_k)$ are prior and marginal probabilities (can be obtained from the given dataset), respectively. Now, the class label L_q of the time series X can be obtained as

$$\underset{L_q}{\text{argmax}}\ \left\{P\left(\frac{L_q}{X}\right)\right\},\ 1 \leq q \leq l. \tag{3.6}$$

For a time series $\mathbf{C}_i^j$, ρ_{i,j,M,L_q} and ρ_{i,j,f,L_q} can be obtained using Equation 3.5 with M and f data points, respectively. By substituting ρ_{i,j,M,L_q} and ρ_{i,j,f,L_q} in Equation 3.4,

the condition can be expressed as

$$\boldsymbol{\alpha} \prod_{k=1}^{f} \frac{P\left(\frac{\mathbf{x}_k}{L_q}\right)}{P(\mathbf{x}_k)} \times \prod_{k=f}^{M} \frac{P\left(\frac{\mathbf{x}_k}{L_q}\right)}{P(\mathbf{x}_k)} \leq \prod_{k=1}^{f} \frac{P\left(\frac{\mathbf{x}_k}{L_q}\right)}{P(\mathbf{x}_k)}. \tag{3.7}$$

On simplification,

$$\boldsymbol{\alpha} \prod_{k=f}^{M} P\left(\frac{\mathbf{x}_k}{L_q}\right) - \prod_{k=f}^{M} P(\mathbf{x}_k) \leq 0. \tag{3.8}$$

The GP classifier considers a time series as a finite set of random variables which are the outcome of a stochastic process. As it uses a joint normal distribution to model the posterior probabilities for each time series in the dataset, the likelihood term $P\left(\frac{\mathbf{x}_k}{L_q}\right)$ can be written as

$$P\left(\frac{\mathbf{x}_k}{L_q}\right) = \frac{1}{\sqrt{(2\pi)^a |\boldsymbol{\Sigma}|}} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}_k - \mu)^{\mathbf{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \mu)},\ 1 \leq k \leq M, \tag{3.9}$$

where, $\boldsymbol{\Sigma}$ is a covariance matrix and $|\boldsymbol{\Sigma}|$ is its determinant. The covariance matrix $\boldsymbol{\Sigma}$ is computed over training time series instances. An entry in the matrix represents a covariance between two time series of the training dataset. By substituting $P(\frac{\mathbf{x}_k}{L_q})$ from Equation 3.9 into Equation 4.11, we get

$$\underbrace{\boldsymbol{\alpha} \frac{(M-f)}{\sqrt{(2\pi)^a |\boldsymbol{\Sigma}|}} \mathrm{e}^{-\frac{1}{2} \sum_{k=f}^{M} (\mathbf{x}_k - \mu)^{\mathbf{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}_k - \mu)}}_{term1} - \underbrace{\prod_{k=f}^{M} P(\mathbf{x}_k)}_{term2} \leq 0. \tag{3.10}$$

As we can not compute the gradient of the above expression, the proposed approach

uses an empirical search method to find f that satisfies the inequality. Moreover, the difference between *term1* and *term2* in Equation 3.10 is always increasing with increase in f. We therefore preferred binary search to solve the inequality.

3.3.1.2 Build classifier

The objective of this work is to solve early classification problem where a classification model is build to classify an incomplete MTD as early as possible while maintaining $\boldsymbol{\alpha}$ accuracy. This step builds an ensemble classifier $\mathbf{H}$ using the estimated MRL from the previous step. The ensemble classifier $\mathbf{H}$ consists of n classifiers (*i.e.,* $\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_n$) corresponding to n components of the MTD as shown in Figure 3.3. Let $\mathbf{D}_i$ denotes a labeled dataset of all the time series of i^{th} component only, where $1 \leq i \leq n$. The proposed approach uses GP to construct a classifier $\mathbf{H}_1$ for $\mathbf{D}_1$ using full-length time series, and use the estimated MRLs for constructing remaining classifiers (*i.e.,* $\mathbf{H}_2, \mathbf{H}_3, \cdots, \mathbf{H}_n$).

As the MRL is computed separately for each time series, the proposed approach finds a class representative MRL for each L_q class, where $1 \leq q \leq l$. Let R_q is an array of MRL of all the time series of i^{th} component that belong to L_q. The class representative MRL for L_q can be computed as

$$\text{MRL}_{i,q} = \underset{f}{\text{argmin}} \left(\frac{1}{N_q} \sum_{j=1}^{N_q} (\boldsymbol{\alpha} - \alpha_{i,j,f,q}) \right), \tag{3.11}$$

where, $f = R_q[k]$ and $1 \leq k \leq N_q$. Further, $\alpha_{i,j,f,q}$ denotes the estimated accuracy that is obtained using the MRL of $\mathbf{C}_i^j$, and it can be computed as $\alpha_{i,j,f,q} = \frac{\rho_{i,j,f,q}}{\rho_{i,j,M,q}}$. Algorithm 3.1 illustrates all the steps for building the ensemble classifier using the MRL of the given dataset $\mathbf{D}$.

Furthermore, the proposed approach builds a Hidden Markov Model [74] as $\Lambda = (A, B, \pi)$, where A, B, π denote a state transition matrix, an observation matrix, and

the initial state distribution, respectively. The model Λ is built using the training dataset $\mathbf{D}$ and the ensemble classifier $\mathbf{H}$. The proposed approach uses the model Λ in prediction phase to obtain the hidden state sequence for the observations of an incomplete MTD. In our case, states correspond to the class labels and observations correspond to the components of the MTD. Let the set of states, denoted by $\mathbf{S}$, is given as $\mathbf{S} = \{S_1, S_2, \cdots, S_l\}$ and the set of observations, denoted by $\mathbf{O}$, is given as $\mathbf{O} = \{\mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_n\}$. The transition matrix $A = \{a_{xy}\}$ is $l \times l$ with

$$a_{xy} = P(\text{state } S_y \text{ at } t+1 \mid \text{state } S_x \text{ at } t). \tag{3.12}$$

To compute a_{xy},

a) Find the state S_x as

$$S_x = \underset{L_q}{\text{argmax}} \left\{ P\left(\frac{L_q}{\mathbf{C}^j_{i,m_i}} \right) \right\},\ 1 \leq q \leq l. \tag{3.13}$$

b) Obtain the MRL m' for observation $\mathbf{C}^j_{i+1}$ using $\mathbf{H}_{i+1}$ for state S_x as $m' = \mathbf{H}_{i+1}[S_x]$.

c) Obtain the state S_y for $\mathbf{C}^j_{i+1,m'}$ using Equation 3.13. At this point, state transition probability from S_x to S_y for j^{th} time series using $\mathbf{C}_{i+1}$ observation is given as

$$a_{xy} = \underset{\forall(S_x \to S_y)}{\text{avg}} \left\{ P\left(\frac{S_y}{\mathbf{C}^j_{i+1,m'}} \right) \right\}. \tag{3.14}$$

As the observation matrix B and initial state distribution π use the data points of an unseen MTD, we compute these matrices during the prediction phase.

3.3.2 Predict a class label of MTD

The previous phase builds a classifier $\mathbf{H}$ which consists the required class-wise MRLs for the MTS. This section predicts a class label of an incomplete MTD by using the built classifier $\mathbf{H}$. The incomplete MTD $\mathbf{C}^p$ consists n number of time series with different

Algorithm 3.1: Building classifier using MRL

Input: A labeled dataset $\mathbf{D}$ of N MTD with l labels, the MTD has n time series where each consists of M data points and associated with λ_i, where $1 \leq i \leq n$;

Output: Classifiers for the dataset $\mathbf{D}$;

Construct $\mathbf{H}_1$ for $\mathbf{D}_1$ using GP classifier.
for *component* $i \leftarrow 2$ *to* n **do**
  for *label* $q \leftarrow 1$ *to* l **do**
    for *time series* $j \leftarrow 1$ *to* N_q **do**
      /* Posterior probabilities of $\mathbf{C}_i^j[M]$ using GP */
      $\rho_{\mathbf{i,j,M}} = \{\rho_{(i,j,M,1)}, \rho_{(i,j,M,2)}, \cdots, \rho_{(i,j,M,l)}\}$.
      $L = 1, R = M$.
      while $L \leq R$ **do**
        $f = \left\lfloor \frac{(L+R)}{2} \right\rfloor$.
        /* Posterior probabilities of $\mathbf{C}_i^j[f]$ using GP * /
        $\rho_{\mathbf{i,j,f}} = \{\rho_{(i,j,f,1)}, \rho_{(i,j,f,2)}, \cdots, \rho_{(i,j,f,l)}\}$.
        /* Let $\mathbf{\Gamma}_f$ is L.H.S. of Equation 3.10 */
        if *(*$(\boldsymbol{\alpha} \times \rho_{i,j,M,q} > \rho_{i,j,f,q})$ *or* $(\mathbf{\Gamma}_f > 0)$*)* **then**
          $L = f + 1$.
        else if *(*$\rho_{i,j,M,q} < \rho_{i,j,f,q}$*)* **then**
          $R = f - 1$.
        else
          $f_{i,j,q} = f$.
          break.
      /* Compute MRL of $\mathbf{C}_i^j$ as $f_{i,j,q}$ */
      $R_q[j] \leftarrow f_{i,j,q}$.
    Compute $\text{MRL}_{i,q}$ for L_q class using Equation 3.11.
/* The classifier for i^{th} component */
$\mathbf{H}_i = \{\text{MRL}_{i,L_1}, \text{MRL}_{i,L_2}, \cdots, \text{MRL}_{i,L_l}\}$.
/* The classifier for the dataset $\mathbf{D}$ */
$\mathbf{H} = \{\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_n\}$.
return $\mathbf{H}$.

sampling rate. The i^{th} component of an incomplete time series is denoted as $\mathbf{C}_i^p$, where $1 \leq i \leq n$. The time series of an incomplete MTD are indexed in non-increasing order of sampling rate. The proposed approach starts prediction as soon as $\mathbf{C}_1^p$ gets M data points. Prediction starts from $\mathbf{C}_1^p$ and ends at component $\mathbf{C}_n^p$. Later, it uses Viterbi algorithm [75] to find the maximum probable sequence of class labels from $\mathbf{C}_1^p$ to $\mathbf{C}_n^p$. Finally, the class label that appears maximum times in the sequence is assigned to the given MTD. The complete procedure is shown in Algorithm 3.2.

3.3.2.1 Label prediction of $\mathbf{C}_1^p$

Different from the existing works, the proposed approach considers correlation among the components of an incomplete MTD while predicting its class label. The correlation is achieved by forwarding the predicted class label of i^{th} component to predict the class label of $(i+1)^{th}$ component. The approach uses full length of $\mathbf{C}_1^p$ for prediction of class label by using $\mathbf{H}_1$. The predicted class label of $\mathbf{C}_1^p$ is given as

$$G_1 = \underset{L_q}{\operatorname{argmax}} \left\{ P\left(\frac{L_q}{\mathbf{C}_{1,m_1}^p} \right) \right\},\ 1 \leq q \leq l, \tag{3.15}$$

where, $\mathbf{C}_{1,m_1}^p$ is first component of the MTD with m_1 number of data points, $m_1 = M\lambda_1^1 = M$, and $G_1 \in \{L_1, L_2, \cdots, L_l\}$.

3.3.2.2 Label prediction of remaining time series of $\mathbf{C}_i^p$

Let predicted class label of C_{i-1}^p is given by G_{i-1}, where $G_{i-1} \in \{L_1, L_2, \cdots, L_l\}$ and $i \geq 2$. The sampling rate of the next time series ($\mathbf{C}_i^p$) is less than the previous time series $\mathbf{C}_{i-1}^p$. Therefore, $\mathbf{C}_i^p$ consists only m_i data points, where $m_i = M\lambda_i^1$. Here, a classifier has two options either wait for more (maximum $M - m_i$) data points or start predicting the class label with m_i without any further delay. To obtain the earliness,

the classification is started with only m_i data points. Let m' denotes the required data points to predict the class label G_{i-1} for time series $\mathbf{C}_i^p$. From Algorithm 3.1, the data points m' is equal to $\mathbf{H}_i[G_{i-1}]$. If $m_i < m'$ then classifier $\mathbf{H}_i$ waits and collects the required data points for class label G_{i-1}. After receiving the required data points (m'), the classifier $\mathbf{H}_i$ predicts the class label (G_i) of $\mathbf{C}_{i,m'}^p$ using Equation 3.15. This process is repeated until all the components of an incomplete MTD are classified. We call this process as class forwarding method.

Since the obtained sequence of predicted class labels (*i.e.,* $G_1, G_2, \cdots, G_n$) may not be the most probable sequence, the proposed approach uses Viterbi algorithm to find the most probable sequence of class labels. It considers the posterior class probabilities for all l classes for all the n components and constructs an observation matrix of the model Λ. As each class label corresponds to a state in the hidden markov model, the set of states are denoted as $\mathbf{S} = \{S_1, S_2, \cdots, S_l\}$. For an incomplete MTD $\mathbf{C}^p$, the observation matrix $B = \{b_y(i)\}$ is $l \times n$ with

$$\begin{aligned} b_y(i) &= P(\text{observation } i \text{ at } t \mid \text{state } S_y \text{ at } t) \\ &= P\left(\frac{\mathbf{C}_{i,m'}^p}{S_y}\right). \end{aligned} \tag{3.16}$$

The initial state distribution $\pi = \{\pi_x\}$ is $1 \times l$ with

$$\begin{aligned} \pi_x &= P(\text{state } S_x \text{ at } t = 0) \\ &= P\left(\frac{S_x}{\mathbf{C}_{1,M}^p}\right). \end{aligned} \tag{3.17}$$

Now, given model $\Lambda = (A, B, \pi)$ and an observation sequence $\mathbf{O} = \{\mathbf{C}_1^p, \mathbf{C}_2^p, \cdots, \mathbf{C}_n^p\}$, the objective is to find a most probable hidden state sequence that most likely generated the observation sequence $\mathbf{O}$. Obtaining such state sequence needs to check all possible state sequences. Viterbi algorithm can efficiently find them.

Let $\mathbf{G} = \{G_1, G_2, \cdots, G_n\}$ be the most probable sequence of class labels obtained using Viterbi algorithm. Now, the predicted class label L^p of $\mathbf{C}^p$ can be given as

$$L^p = L_q,$$
$$s.t. \quad \underset{1 \leq q' \leq l, q' \neq q}{\forall} count(L_q, \mathbf{G}) \geq count(L_{q'}, \mathbf{G}). \tag{3.18}$$

where, $count(L_q, \mathbf{G})$ counts the number of occurrences of class label L_q in vector $\mathbf{G}$.

3.3.3 Complexity Analysis

Time complexity: As a separate classifier is constructed for each component, all the classifiers can be constructed simultaneously. That means step 2 can be removed from the calculation of time complexity of the Algorithm 3.1. Now, the time complexity of Algorithm 3.1 is mostly depends on two **for** loops, one **while** loop, and GP classifier. The **for** loop at third and fourth steps run l and N_q times, respectively. The **while** loop runs $log_2(M)$ times and the computation of posterior probabilities takes $O(N^3)$ using GP. The computational complexity of GP can be approximated to $O(p^2N)$ by selecting a subset of p MTD from N and $p \ll N$ [42]. Therefore, time complexity of Algorithm 3.1 can be given as $O(N^2)$. Now, given the model $\Lambda = (A, B, \pi)$, the time complexity of Algorithm 3.2 can be given as $O(n \times nl^2)$, where nl^2 is computational time of Viterbi algorithm. Since n and l are very small, they can be taken as constant.

Space complexity: In Algorithm 3.1, GP computes the posterior class probabilities for each time series for all the components. Moreover, these probabilities are computed using varying number of data points (*i.e.,* from 1 to M). Hence, the space complexity of Algorithm 3.1 is $O(N \times n \times l \times M) = O(MN)$, as $\{l, n\} << N$. After construction, the ensemble classifier $\mathbf{H}$ needs $N \times n \times l^2$ space. The model Λ takes $l^2 + nl + l$ space to store its matrices. As $\mathbf{H}$ and Λ are given as input to Algorithm 3.2, its space complexity is given as $O((N \times n \times l^2) + (l^2 + nl + l)) = O(N)$.

Algorithm 3.2: Predict a class label of MTD

Input: The vector of the classifiers $\mathbf{H}$ using **Algorithm 3.1** and an incomplete MTD $\mathbf{C}^p$;

Output: Predict class label L^p of $\mathbf{C}_i^p$, where $1 \leq i \leq n$;

$\mathbf{H}_1$ predicts the class label G_1 for $\mathbf{C}_1^p$, when $m_1 = M$.
for *component* $i \leftarrow 2$ *to* n **do**
  /* The available data points of $\mathbf{C}_i^p$ */
  $m_i = \max\{M\lambda_i^1, m_{i-1}\}$.
  /* MRL to predict G_i for C_i^p */
  $m' = \mathbf{H}_i[G_{i-1}]$.
  if *($m' \leq m_i$)* **then**
    $\mathbf{H}_i$ predicts the class label G_i for $\mathbf{C}_i^p$ using GP.
  else
    /* Collect more data points */
    Wait till m_i is equal to m'.
    $\mathbf{H}_i$ predicts the class label G_i for $\mathbf{C}_i^p$ using GP.
/* Given model Λ and $\mathbf{O} = \{\mathbf{C}_1^p, \mathbf{C}_2^p, \cdots, \mathbf{C}_n^p\}$ */
Obtain $\mathbf{G}$ using Viterbi algorithm [75].
Predict class label L^p of $\mathbf{C}^p$ from $\mathbf{G}$ using Equation 3.18.
return L^p.

Function *Viterbi_Alg($\Lambda, \mathbf{O}$)*:
begin
  for each state $x = 1$ to l
    $\delta_1(x) = \pi_x b_x(\mathbf{C}_1^p)$.
    $\psi_1(x) = 0$.
  for each time step $t = 1$ to $n - 1$
    for each state $y = 1$ to l
      $\delta_{t+1}(y) = \max_{1 \leq x \leq l}[\delta_t(x)a_{xy}]b_y(\mathbf{C}_{t+1}^p)$.
      $\psi_{t+1}(y) = \operatorname{argmax}_{1 \leq x \leq l} \delta_t(x)a_{xy}$.
  $\hat{S}_n = \operatorname{argmax}_{1 \leq x \leq l} \delta_n(x)$.
  for $t = n - 1$ to 1
    $\hat{S}_t = \psi_{t+1}(\hat{S}_{t+1})$
    $\hat{\mathbf{S}} \leftarrow \text{append}(\hat{S}_t)$
  return $\hat{\mathbf{S}}$
end

3.4 On-Road Experiment and Results

This section evaluates the proposed early classification approach for road surface classification. We discuss the experimental setup to collect the sensory data using a SensorTag kit and a smartphone. The sensory data is proprocessed to create a dataset that consists only the complete MTS. Next, the approach uses this dataset to train the ensemble classifier **H**. Finally, the classifier **H** is evaluated using accuracy, earliness, and F_1 score.

3.4.1 Prototype

This work uses a SensorTag kit (*i.e.,* CC2650STK) and a smartphone (*i.e.,* Samsung Galaxy Alpha). The SensorTag kit comes in small package that includes ten sensors, such as, accelerometer, gyroscope, light, temperature, and so on. The SensorTag runs on battery power which lasts for months to year. It uses iBeacon technology based low energy Bluetooth for the transmission of sensory data. It also provides an app for Android based smartphone to customize the setting of the sensors, such as turn on/off and sampling rate. The provided app receives the sensory data and stores in a file (*i.e.,* csv). This experiment considers four sensors including accelerometer, light, temperature, and humidity from SensorTag kit with a sampling period 500ms, 1500ms, 1000ms, 1200ms, respectively. The GPS coordinates are also recorded along the data of sensors using smartphone.

3.4.2 Dataset creation

In this experiment, we collect the sensory data of four types of road surface: rough (S1), uneven (S2), bumpy (S3), and smooth (S4). The experiment is carried out with eight participants using two vehicles (*i.e.,* bike and car). SensorTag location on the vehicles is shown in Figure 3.4. The driving activity is performed by 8 participants on 4 types of road surfaces. Each participant performs the activity 12 times on each road surface

using 2 vehicles. Therefore, total number of MTD is $8 \times 12 \times 4 \times 2 = 768$, *i.e.,* $N = 768$. The sensor readings are taken three times a day to observe the influence of environment light on the driving pattern. The number of used sensors are 4, which correspond to 4 components (time series, $n = 4$) of MTD. The driving activity is performed on different lengths (*i.e.,* 0.5 km to 2 km) of road. We found that the data points collected for 1 km length of road (with an average speed of $20 \sim 30$ km/hr) are required for successful classification. The full length is obtained as $M = \frac{\text{distance}}{\text{speed of vehicle}} \times$ no. of samples/second.

For example, in 3 minutes (180 seconds), the maximum number of data points that can be collected from the sensors, are as follows: $360 = 180 \times 2$ for accelerometer, $120 = 180 \times 2/3$ for light, $180 = 180 \times 1$ for temperature, and $150 = 180 \times 5/6$ for humidity. Due to variation in the speed of vehicles, the minimum number of data points, that are obtained in each of the time series of the MTD, is 110 (*i.e.,* $M = 110$). The created dataset is referred as road surface dataset. We preprocess the road surface dataset for cleaning the random errors and missing values using binning and imputation [76].

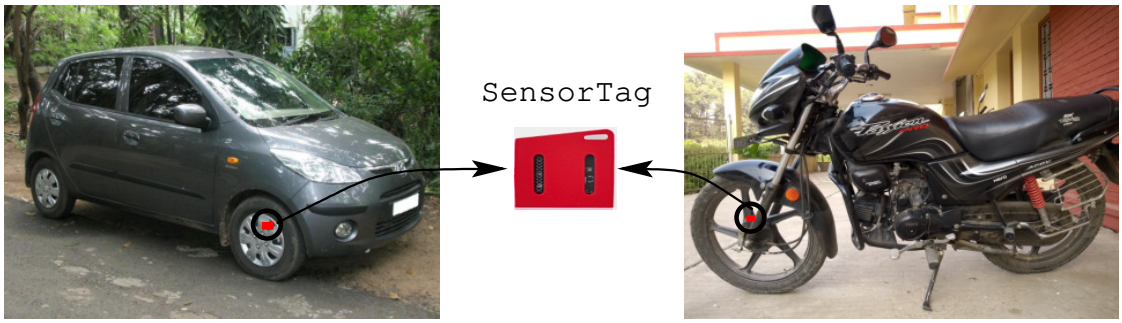


Figure 3.4: SensorTag location on the vehicles during experiment.

3.4.3 Evaluation metrics

This work uses following evaluation metrics:

- ***Accuracy:*** We calculate the accuracy using Equation 3.2.
- ***Earliness:*** It is calculated using Equation 3.1.
- ***F_1 score:*** This work defines F_1 score to study the impact of earliness on accuracy

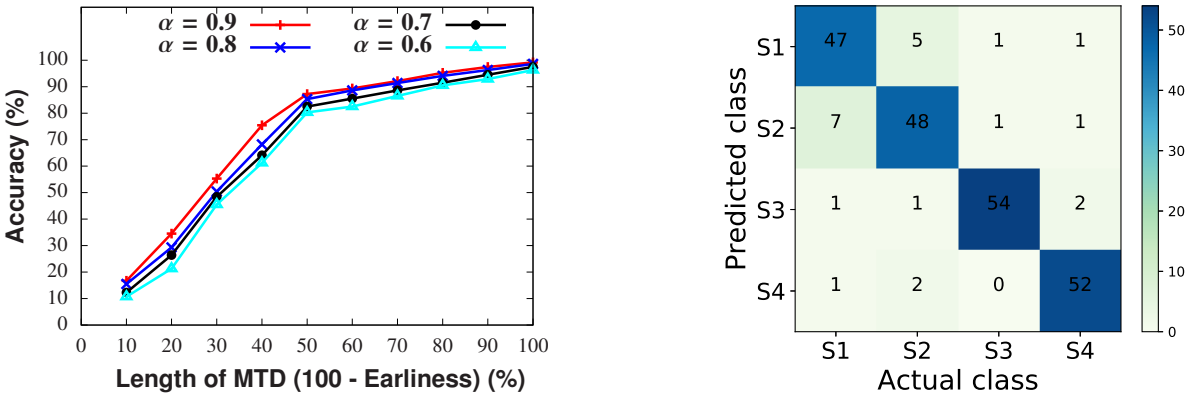
of the early classifier. It is mathematically expressed as

$$F_1 = \frac{2 \times Accuracy \times (1 - \frac{MRL}{M})}{(1 - \frac{MRL}{M}) + Accuracy}. \tag{3.19}$$

- ***Confusion matrix:*** The confusion matrix in this work, is a table that is used to describe the performance of the proposed approach on the given labeled dataset. The confusion matrix is used to understand the class-wise performance of the proposed classifier **H**.

3.4.4 Results

The proposed early classifier (*i.e.,* **H**) is evaluated on the road surface dataset. The dataset is divided into two parts with 70% and 30% MTD for training and testing, respectively. The classifier **H** is built using 10×5 cross-validation method.



(a) Accuracy versus earliness

(b) Confusion matrix

Figure 3.5: Illustration of the experimental results of the proposed approach for road surface dataset using $\boldsymbol{\alpha} = 0.9$.

In road surface dataset, the order of the four components in the MTD is $\mathbf{C}_1$ (accelerometer), $\mathbf{C}_2$ (temperature), $\mathbf{C}_3$ (humidity), and $\mathbf{C}_4$ (light). The normalized sampling rate of the components are $\{1, 0.5, 0.4, 0.3\}$, which are normalized with respect to the maximum sampling rate (*i.e.,* accelerometer). Here, the ensemble classifier **H** ob-

tains the 90.4% accuracy using only 52.1% data points of the MTD and the computed F_1 score is 0.70 using Equation 3.19. Figure 3.5 shows the tradeoff along the progress of the MTD with an interval of 10% in part (a) and the confusion matrix using $\boldsymbol{\alpha} = 0.9$ in part (b). It can be clearly observed from part (a) of Figure 3.5 that the ensemble classifier **H** is able to achieve around 85% accuracy with an earliness of more than 45% at all settings of $\boldsymbol{\alpha}$. The road surface dataset has four classes and each class has 56 MTD in the testing part. It is clear from part (b) of Figure 3.5 that the bumpy class gets maximum accuracy and the MTD of bumpy class has never predicted as smooth.

3.4.4.1 Impact of equal and different sampling rate

This work evaluates the impact of equal and different sampling rate components on the MRL of proposed approach using road surface dataset and an existing PEMS-SF dataset [13]. Let MRL_{eql} and MRL_{dif} denote the MRL of the MTS with equal and different sampling rate components, respectively. Figure 3.6 illustrates the impacts of sampling rate on the MRL of the proposed approach at $\boldsymbol{\alpha} = \{0.5, 0.6, 0.7, 0.8, 0.9\}$. The approach requires smaller MRL_{dif} than MRL_{eql} for both the datasets at all $\boldsymbol{\alpha}$. Its is also observed that the difference between MRL_{dif} and MRL_{eql} increases with $\boldsymbol{\alpha}$. It indicates that the improvement in performance of the approach with different sampling rate components is more than the equal sampling rate components.

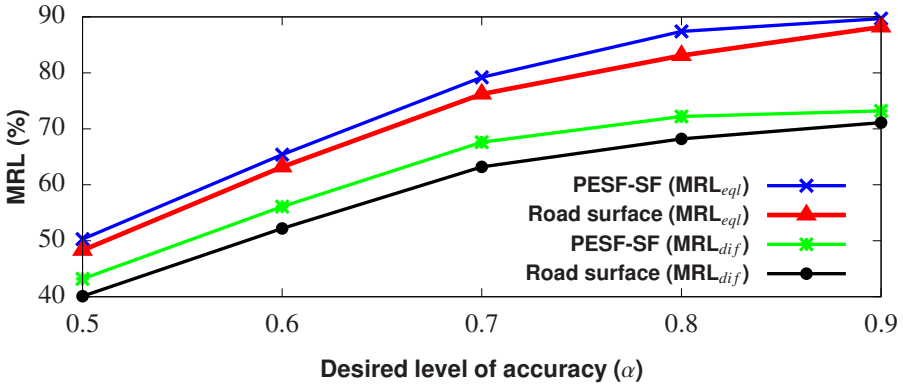


Figure 3.6: Impact of equal and different sampling rate components on the MRL.

3.5 Results on Existing Datasets and Discussions

This section evaluates the performance of the proposed approach using the existing labeled datasets which are publicly available in [13]. First, the existing datasets are described in detail. These datasets are used to evaluate the performance of the proposed work by using the performance metrics, discussed in Section 3.4.3. Later, the proposed work is compared with the existing recent work including MSD [26], Optimizing Accuracy and Earliness (OAE) [25], and DMP+PPM [29].

3.5.1 Dataset

This work uses freeway occupancy (called as *PEMS-SF*), Heterogeneity Human Activity Recognition (HHAR), and Gas Mixtures Detection (GMD) datasets, which are available in the UCI repository [13].

3.5.1.1 PEMS-SF dataset

It is created by the California Department of Transportation to study the freeway occupancy rate of car lanes in the area of San Francisco bay. The dataset consists of the measurements collected over the period of 15 months. Total 963 sensors were deployed at different locations of highways. The sensors collect 144 data points each day, which forms MTD. The number of sensors work as the components of MTD. The proposed approach considers ten random sensors as ten components of MTD, *i.e.,* $n = 10$. The collected data points from each sensor in a day is equal to the length of the time series, *i.e.,* $M = 144$. Each measurement represents the occupancy rate of a car lane and normalizes to $[0, 1]$. The dataset contains 440 MTD ($N = 440$) with the labeled as the day of the week.

3.5.1.2 HHAR dataset

This dataset is created to understand the impact of heterogeneities of the used sensors on human activity classification. It consists of six different activities standing (A1), sitting (A2), walking (A3), stair up (A4), stair down (A5), and biking (A6). Each activity is performed by nine participants in a specified manner. The activities are recorded using two motion sensors (accelerometer and gyroscope) of smartphones and smartwatches. The number of used sensors are the four components of MTD, *i.e.,* $n = 4$. The continuous readings of sensors are preprocessed to fix the length of full time series to 500 data points, *i.e.,* $M = 500$. After preprocessing, the dataset contains 20000 MTD, *i.e.,* $N = 20000$.

3.5.1.3 GMD dataset

This dataset consists the temporal data of eight sensors monitoring the mixture of Ethylene with Carbon Monoxide or Methane at different level of concentrations in a wind tunnel. The proposed approach uses the reading of eight gas sensors as eight components of MTD, *i.e.,* $n = 8$. The dataset contains 30 configurations of mixture using four levels of concentrations (*i.e.,* zero, low, medium, and high). Each mixture is recorded for five minutes with ten samples per second, which makes a time series of length 3000. The dataset is first preprocessed and then 18 different configurations are considered in this experiment, *i.e.,* $l = 18$. The measurements taken in first and last one minute are removed from the time series, which reduced the length to 1800 data points, *i.e.,* $M = 1800$.

3.5.2 Experimental results

This section presents the experimental results of the proposed approach on the three above mentioned datasets. These datasets have different number of components as discussed in Section 3.5.1. Each dataset is divided into two parts with 70% and 30%

MTD in training and testing, respectively. The proposed approach employs a standard $10 \times 5-$fold cross-validation method to train the classifier $\mathbf{H}$ for all the datasets. In $5-$fold cross-validation, the dataset is divided into 5 equal parts (folds) where 4 of them are used for training and remaining 1 is used for validating the trained classifier. With 4 parts for training and 1 part for validation, there are total 5 possible combinations on which the classifier is trained and validated. This process is repeated 10 times in $10 \times 5-$fold cross-validation and the resultant classifier is used for testing.

The proposed approach considers different sampling rate for each component to evaluate the performance. This work uses four different sets of the sampling rate, denoted as Ω_1, Ω_2, Ω_3, and Ω_4. The set Ω_1 and Ω_2 consist equal gap between sampling rate of two consecutive component, and set Ω_3 and Ω_4 have varying gap. The set Ω_2 and Ω_4 also have some components with identical sampling rate. Table 3.1 shows the different settings of sampling rate taken in the implementation.

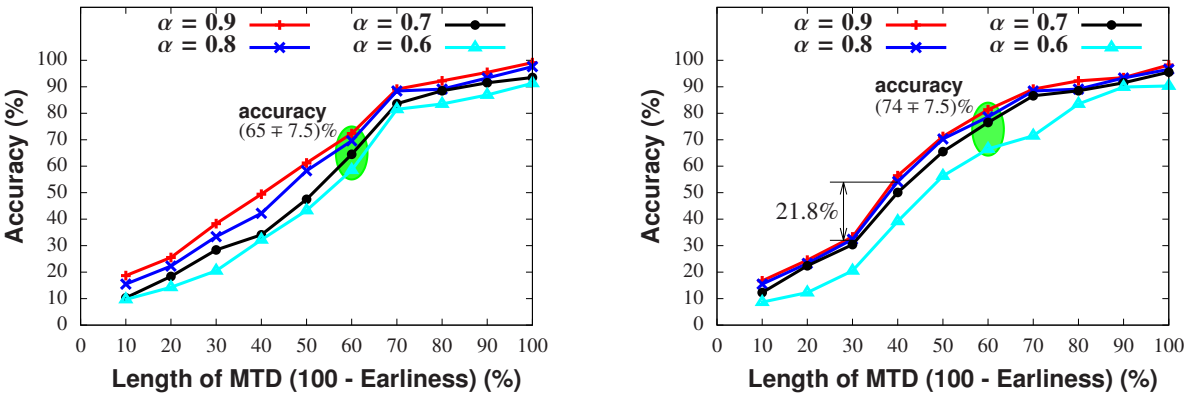
Table 3.1: Sets of sampling rate setting for the used datasets.

Dataset	**Set of sampling rate (Ω)**
PEMS-SF	$\Omega_1 = \{1, .95, .90, .85, .80, .75, .70, .65, .60, .55\}$ $\Omega_2 = \{1, .90, .80, .80, .70, .70, .60, .60, .50, .40\}$ $\Omega_3 = \{1, .85, .80, .75, .65, .60, .50, .45, .35, .30\}$ $\Omega_4 = \{1, .90, .70, .55, .55, .45, .35, .35, .30, .20\}$
HHAR	$\Omega_1 = \{1, .90, .80, .70\}$, $\Omega_2 = \{1, .80, , 80, .60\}$ $\Omega_3 = \{1, .80, .70, .45\}$, $\Omega_4 = \{1, .60, .60, .35\}$
GMD	$\Omega_1 = \{1, .95, .90, .85, .80, .75, .70, .65\}$ $\Omega_2 = \{1, .90, .80, .80, .70, .70, .60, .50\}$ $\Omega_3 = \{1, .85, .80, .75, .65, .60, .50, .45\}$ $\Omega_4 = \{1, .90, .70, .55, .55, .45, .35, .35\}$

3.5.2.1 Tradeoff between accuracy and earliness

To analyze the tradeoff between accuracy and earlinesss, this work computes the accuracy of the ensemble $\mathbf{H}$ along the progress of the MTD. An interval of 10% of MTD is chosen to compute the accuracy at $\boldsymbol{\alpha} = \{0.9, 0.8, 0.7, 0.6\}$ for PEMS-SF and HHAR

datasets, as depicted in parts (a) and (b) of Figure 3.7, respectively. It is clear from Figure 3.7 that the performance of the proposed approach degrades consistently as the value of $\boldsymbol{\alpha}$ decreases for all the datasets. At 60% length of the MTD, HHAR dataset obtains maximum accuracy (*i.e.*, $(74 \mp 7.5)\%$) among all the datasets for all values of $\boldsymbol{\alpha}$, as shown by 'green' oval. An interesting event occurs between 30% to 40% length of MTD as shown in part (b), where accuracy is suddenly increased around 21.8%. It indicates that the proposed approach finds better correlation for the MTD with less number of components.



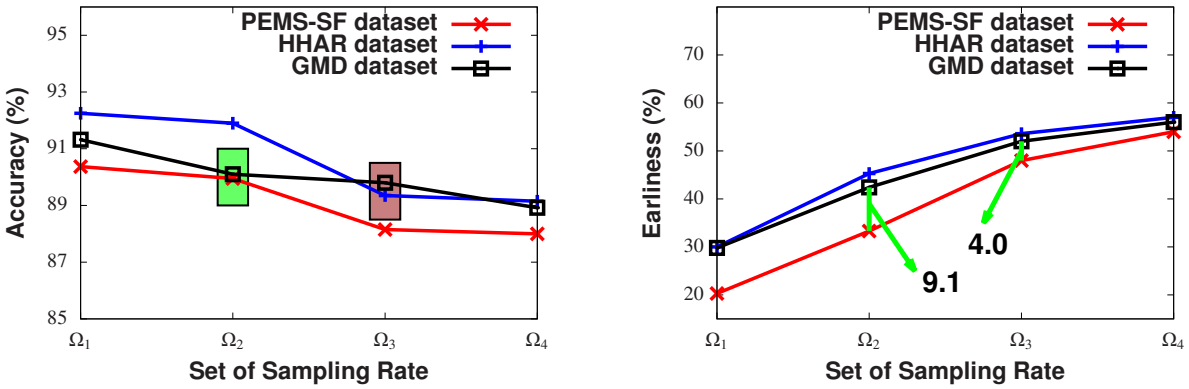
(a) PEMS-SF dataset (b) HHAR dataset

Figure 3.7: Illustration of the tradeoff between accuracy and earliness for PEMS-SF and HHAR.

3.5.2.2 Impact of the set of sampling rate

We discuss the accuracy and earliness results of the proposed approach on three datasets with four sets of sampling rate. Parts (a) and (b) of Figure 3.8 illustrate the accuracy and earliness results at $\boldsymbol{\alpha} = 0.9$, respectively. In part (a) of Figure 3.8, at Ω_2, PEMS-SF and GMD datasets obtain almost same accuracy as both have more number of components than HHAR dataset, which is shown by 'green' rectangle. Later, at Ω_3, since the number of data points in MTD of GMD dataset is much higher than PEMS-SF dataset, the GMD dataset improves its accuracy very fast and even obtains better accuracy than HHAR dataset, as shown by 'brown' rectangle. In part (b) of Figure 3.8,

the earliness of PEMS-SF dataset shows maximum improvement from Ω_2 to Ω_3 (*i.e.*, $5.1\% = 9.1\% - 4.0\%$) while other datasets show nearly uniform variation for different sets of sampling rate.



(a) Accuracy (b) Earliness

Figure 3.8: Impact of different sets of sampling rate on accuracy and earliness.

3.5.3 Performance comparison

This section compares the performance of the proposed approach with the existing work including MSD [26], OAE [25], and DMP+PPM [29]. Table 3.2 illustrates the accuracy and earliness results on PEMS-SF ($M = 144$), HHAR ($M = 500$), and GMD ($M = 1800$) datasets using $\boldsymbol{\alpha} = 0.9$. The following points are observed from the table:

- The proposed approach outperforms the existing approaches on accuracy and earliness metrics at both sets of sampling rate. This is because the approach utilizes correlation between the components of the MTD, which helps to achieve better accuracy using less data points. At Ω_1 for PEMS-SF dataset, the MSD is able to obtain nearly equal accuracy compared to the proposed approach with a marginal difference (*i.e.*, $1.44\% = 90.37\% - 88.93\%$), but it uses more data points.
- The results show a substantial drop in the accuracy of the existing approaches from Ω_1 to Ω_4. For example, the OAE shows the maximum drop (*i.e.*, $14.0\% = 82.15\% - 68.15\%$) in accuracy for PEMS-SF dataset while the proposed approach shows only 2.37% drop. It suggests that the performance of the proposed ap-

proach degrades marginally even when there exists a large difference between the sampling rate of first and last components.

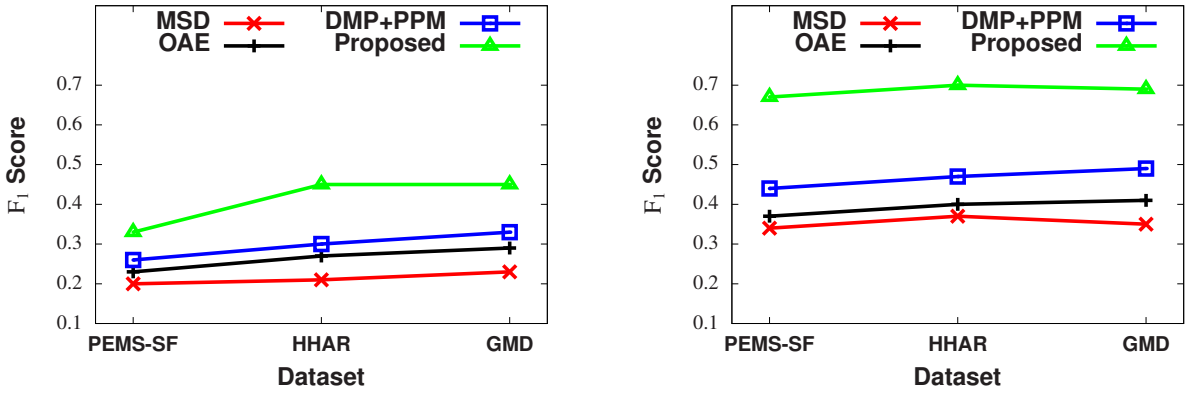
- The proposed approach provides better results for HHAR dataset compared to other datasets. The proposed approach achieves 89.15% accuracy by using only 215 (*i.e.,* 57% earliness) data points.

Table 3.2: Illustration of accuracy and earliness results using $\boldsymbol{\alpha} = 0.9$.

Dataset	Approach	$\boldsymbol{\Omega_1}$		$\boldsymbol{\Omega_4}$	
		Accuracy	Earliness (MRL)	Accuracy	Earliness (MRL)
PEMS-SF $M = 144$	**MSD**	88.93%	11.3% (128)	76.90%	21.3% (113)
	OAE	82.15%	13.5% (125)	68.15%	25.5% (107)
	DMP+PPM	76.45%	16.2% (121)	65.48%	32.9% (97)
	Proposed	**90.37%**	**20.3% (115)**	**88.0%**	**54.0% (66)**
HHAR $M = 500$	**MSD**	84.19%	12.0% (440)	67.13%	25.6% (372)
	OAE	83.12%	16.2% (419)	71.12%	28.6% (357)
	DMP+PPM	84.41%	18.6% (407)	72.21%	34.4% (328)
	Proposed	**92.25%**	**30.0% (350)**	**89.15%**	**57.0% (215)**
GMD $M = 1800$	**MSD**	85.21%	13.6% (1555)	71.35%	23.1% (1384)
	OAE	80.13%	17.6% (1483)	68.26%	29.7% (1265)
	DMP+PPM	79.57%	20.6% (1429)	69.0%	37.8% (1120)
	Proposed	**91.32%**	**29.8% (1264)**	**88.92%**	**56.0% (792)**

3.5.3.1 $\mathbf{F_1}$ score

The proposed approach is also compared with the existing approaches using F_1 score for all the datasets. The F_1 score is computed using Equation 3.19 and the results using Ω_1 and Ω_4 are illustrated in parts (a) and (b) of Figure 3.9, respectively. It can be clearly observed from the results that the proposed approach consistently performing better than the existing approaches. In part (a) of Figure 3.9, existing approaches hardly achieve F_1 score more than 0.30 whereas the proposed approach obtains 0.45 (for HHAR dataset), which clearly indicates that the proposed approach is able to maintain better balance between accuracy and earliness. Moreover, as the sampling rate of last component is decreased, as in Ω_4, the proposed approach outperforms the existing approaches with a substantial difference, as shown in part (b) of Figure 3.9.



(a) Using Ω_1 (b) Using Ω_4

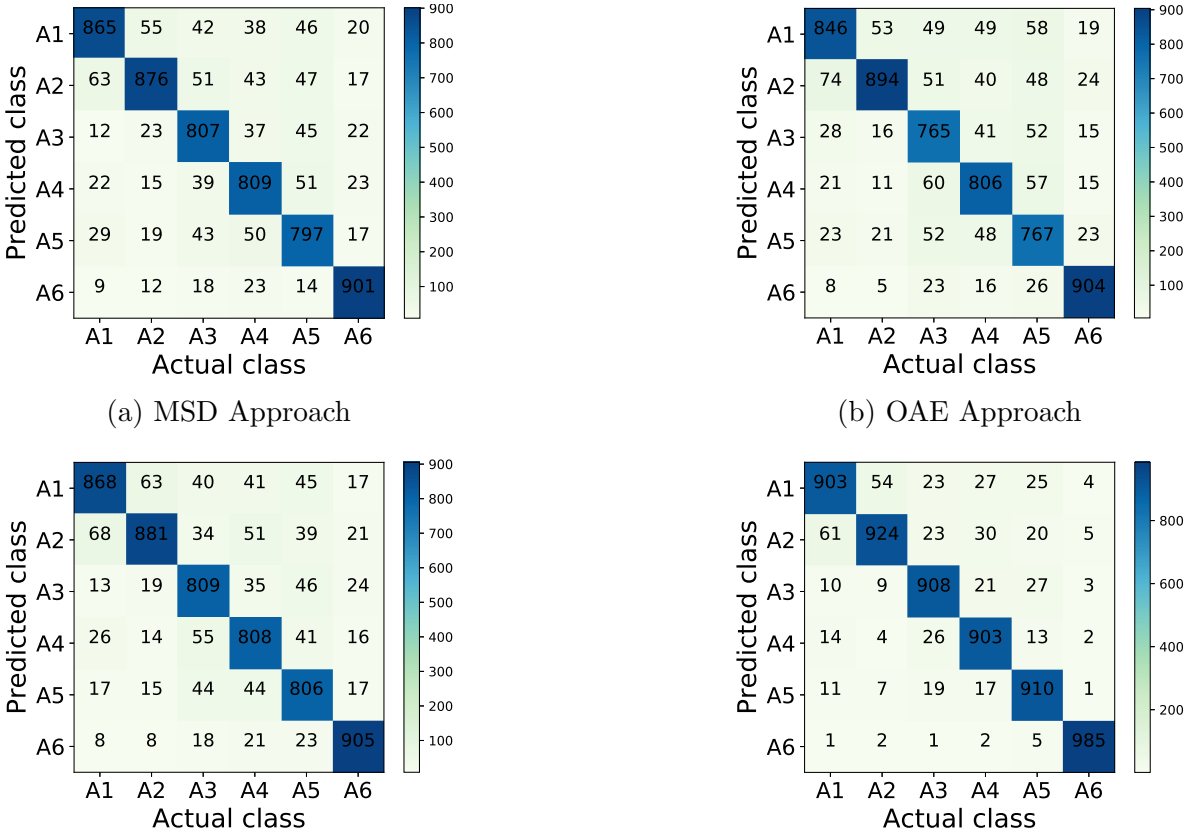
Figure 3.9: Comparison of the proposed approach with existing approaches using F_1 score metric.

3.5.3.2 Confusion matrix

We present the confusion matrices for HHAR testing dataset only, as shown in Figure 3.10. The reason for choosing this dataset is that it provides better comparison states among the different classification approaches. This dataset has six class labels, *i.e.,* A1 to A6 as standing, sitting, walking, stair up, stair down, and biking. Each class has equal number of MTD, *i.e.,* 1000. Figure 3.10 shows that the accuracy is highest (*i.e.,* 98.5%) for A6 class. This is because, the activity A6 consists many identifiable patterns in the MTD which can be easily differentiated from other class MTD. The result also illustrates that the proposed work provides the highest accuracy in all the classes. It is also observed that all the existing approaches show maximum confusion between the MTD of first two classes (*i.e.,* standing and sitting).

3.5.3.3 Impact of desired accuracy level ($\boldsymbol{\alpha}$)

Next, we analyze the impact of the level of accuracy on the earliness for PEMS-SF dataset as it has maximum number of components in the MTD. Figure 3.11 illustrates the results among MSD, OAE, DMP+PPM, and the proposed approach. The earliness of the proposed approach is computed at $\boldsymbol{\alpha} = \{0.5, 0.6, 0.7, 0.8, 0.9\}$. This result is



(a) MSD Approach

(b) OAE Approach

(c) DMP+PPM Approach

(d) Proposed Approach

Figure 3.10: Confusion matrices for HHAR dataset using Ω_1.

obtained using Ω_1 set of sampling rate for PEMS-SF dataset. It is clear from the result that the proposed approach consistently performs better than the existing approaches for different settings of $\boldsymbol{\alpha}$.

3.5.3.4 Execution time

Finally, the proposed approach is compared with existing approaches based on the execution time. Let T_{pro} and T_{ext} denote the execution time of the proposed and one of the existing approaches [25, 26, 29], respectively. Table 3.3 illustrates that how fast the proposed approach is than the existing approach, which is computed as $\frac{T_{\text{ext}} - T_{\text{pro}}}{T_{\text{ext}}}$ (in %). The experiment is carried out 100 times using Ω_1 and $\boldsymbol{\alpha} = 0.9$, and an average execution time is taken for comparison. It is observed from the Table 3.3 that the

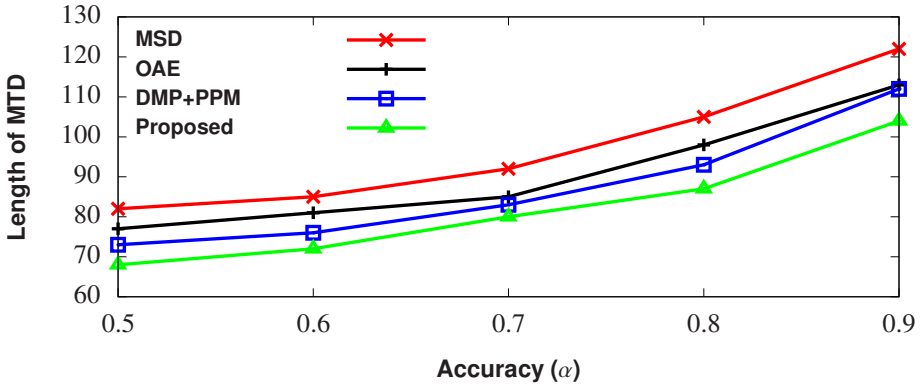


Figure 3.11: Impact of $\boldsymbol{\alpha}$ on the earliness for PEMS-SF dataset using Ω_1.

proposed approach is faster than the existing approaches. It also illustrates that MSD takes 17.3% more time than the proposed approach for PEMS-SF. This is because the MSD does not use correlation between components of the MTS.

Table 3.3: Illustration of execution time comparison for different datasets.

Existing approach	**Dataset**		
	PEMS-SF	HHAR	GMD
MSD	**17.3%**	13.4%	15%
OAE	10.9%	7.4%	13.2%
DMP+PPM	4.9%	8.2%	6.1%

3.6 Conclusion

In this chapter, we proposed an approach for early classification of an incomplete MTD while maintaining the desired level of accuracy. The MRLs are estimated for each class label using a probabilistic classifier. The estimated MRLs are used to build an ensemble classifier. The built classifier uses a class forwarding method to predict the class label of an incomplete MTD. Finally, the performance of the proposed approach is evaluated for early classification of the road surface in the intelligent road transportation system. The experimental results illustrated that the proposed approach achieves better results than the competitive methods on existing datasets from other domains. Although our focus

in this chapter is on the early classification of MTD, the approach can be adjusted to other applications such as gas leakage detection, similarity analysis in driving patterns, and disease diagnostics.

This chapter mainly focused on handling the components of different sampling rate while classifying an incomplete MTS. We utilized the posterior class probabilities for learning the MRLs that helped in building a class forwarding method from highest to lowest sampling rate component. This method inherently incorporated the correlation in the early decision of class label. In the next chapter, we work on the early classification of MTS in the presence of faulty sensors. We discuss the identification of faulty data components by using the training data. These components not only degrade the earliness of the classifier but misguide the learning process also, which may result in poor classification accuracy.

Publication

- **Ashish Gupta**, H. P. Gupta, B. Biswas, and T. Dutta, "An early classification approach for multivariate time series of on-vehicle sensors in transportation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5316-5327, 2020.