Online Ensemble Learning Framework for Class Evolution under Incomplete Supervision

Zhi Cao[1,2,3], Shuyi Zhang[3], Chin-Teng Lin[4], Xin Yao[5†]
[1]*Anhui Transport Consulting & Design Institute Co., Ltd, Hefei, China*
[2]*University of Science and Technology of China, Hefei, China*
[3]*Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China*
[4]*CIBCI Lab, School of Computer Science, University of Technology, Sydney, Australia*
[5]*School of Data Science, Lingnan University, Hong Kong SAR, China*

***Abstract*—The composition of classes in real-world data streams is constantly evolving. To maintain the classifier performance, labeling a limited number of instances for a novel class is more affordable and feasible than annotating the entire data steam. This poses a challenge for learning with class evolution under incomplete supervision. However, existing semi-supervised algorithms typically require storing past data to adapt classifiers to novel classes. Despite many online learning algorithms are proposed to tackle class evolution without retaining historical data, these methods are supervised. In this paper, we propose a novel Semi-supervised Online Ensemble Learning Framework (SOELF) to handle class evolution under incomplete supervision. This framework adapts in an online learning manner without the storage of past data. It integrates a data generation module based on an online ensemble clustering model to generate diverse instances to exploit the unlabeled instances for model adaptation. The decaying G-mean metric is employed to determine the pseudo-labels of these generated instances. Furthermore, an inverse-prior-ratio model adaptation strategy is designed to introduce more instances from existing classes for model adaptation when novel classes emerge with labeled instances, thereby balancing the exploitation between classes. Experimental studies on both synthetic and real-world data streams demonstrate that our method achieves higher accuracy compared to existing online learning algorithms and the state-of-the-art semi-supervised algorithm that requires data storage.**

***Index Terms*—ensemble model, online learning, class evolution, incomplete supervision, semi-supervised learning**

I. Introduction

In real-world data stream mining, a pervasive challenge is the constantly evolving composition of classes, referred as "class evolution" [23]. In the evolving environment, novel classes may emerge, existing classes may disappear, and disappeared classes may reoccur. Although it is crucial for classifiers to adapt to novel classes, labeling the entire data stream for model adaptation is impractical. A feasible solution involves experts labeling a few instances from these novel classes. For example, labeling a few instances for a novel trending topic on *Twitter* or for a novel object encountered by a robot in a new environment [9] would significantly aid in maintaining the classifiers. However, this introduces challenges for learning with class evolution under incomplete supervision [30]. Furthermore, real-world data streams are generated in large volumes and at high velocities [20], imposing considerable demands on memory constraints and processing speed. Online learning, which processes instances one-by-one and does not rely on storing historical data, has emerged as a promising approach for managing such data streams [27].

However, in addressing class evolution under incomplete supervision, existing semi-supervised algorithms either process data streams in a chunk-by-chunk manner [19], [15], [29] or rely on a buffer module to store past instances [8], [10], [7]. In both cases, classifiers become capable of classifying novel classes only after collecting a substantial number of instances. Many online learning frameworks have also been proposed, capable of adapting to class evolution from a single instance [16], [23], [3]. However, these methods are all supervised. Under incomplete supervision, these methods exclusively learn from labeled instances and overlook the abundance of unlabeled instances in the data stream, leading to suboptimal performance in multi-class classification.

In this work, we introduce a novel Semi-supervised Online Ensemble Learning Framework (SOELF) to address class evolution under incomplete supervision in data streams. Unlike existing semi-supervised approaches to class evolution, which necessitate storing historical data, our framework processes instances in an online manner, eliminating the need for retaining past data. To the best of our knowledge, this is the first work to address online learning with class evolution under incomplete supervision.

SOELF integrates a data generation module with an inverse-prior-ratio model adaptation strategy to exploit unlabeled instances for model adaptation. Specifically, the data generation module is constructed based on an online ensemble clustering model, and generates diverse instances by exploiting the current unlabeled instance, thereby enhancing the adaptation of classifiers within the ensemble framework. The decaying G-mean metric is utilized to assign reliable pseudo-labels to these generated instances. Moreover, the inverse-prior-ratio model adaptation strategy introduces more instances from existing classes for model adaptation when novel classes emerge with labeled instances, preventing the model from focusing excessively on these labeled instances and ensuring balanced exploitation between classes. Comprehensive experiments on

† Corresponding author.

both synthetic and real-world data streams under diverse incomplete supervision scenarios demonstrate that SOELF achieves higher accuracy compared to both existing online learning algorithms and the state-of-the-art semi-supervised algorithm that requires data storage. The source code is available at https://github.com/lflfdxfn/SOELF-Implementation.

Our main contributions are as follows:

- We propose a novel online learning framework SOELF to handle class evolution under incomplete supervision. To the best of our knowledge, this is the first work to investigate the problem of online learning with class evolution under incomplete supervision.
- We design a data generation module to generate diverse instances from a single unlabeled instance. This module is based on an online ensemble clustering model, utilizing a decaying G-mean metric to provide more reliable pseudo-labels. Experimental results demonstrate the superiority of this module in enhancing the classification performance.
- We design an inverse-prior-ratio model adaptation strategy that introduces more instances from existing classes when novel classes emerge, balancing exploitation between classes and achieving more accurate predictions.

II. Related Work

Research addressing class evolution under incomplete supervision encompasses tasks such as novel class detection and novel class adaptation [6]. In this paper, our primary focus is on the task of effectively adapting the model to class evolution in the data stream instead of detection.

Concerning model adaptation, various semi-supervised algorithms have been proposed to maintain an efficient classification model. Based on the number of maintained classifiers, these algorithms are categorized into ensemble-model approaches and single-model approaches. For ensemble-model approaches, a new base classifier is built with a recent bunch of incoming data, and is incorporated into the ensemble model. In ActMiner [19], the new base classifier is trained on each data chunk containing both labeled instances and unlabeled instances assigned with pseudo-labels. SCDMiner [13] retains instances with a dynamic chunk boundary for better model adaptation. The following work, such as SAND [14], ECHO [15], NovelDetector$^{\text{DAE}}$ [22], ESCR [29], selects labeled instances and unlabeled instances with high-confidence pseudo-labels within each dynamic chunk for training a new base classifier. SENCForest [21] introduces a two-level mechanism to adapt models to novel classes. This mechanism involves using buffered instances at both the ensemble level, in which a new forest is built, and the individual base classifier level, in which subtrees are grown. However, all aforementioned ensemble-based frameworks heavily rely on a substantial amount of buffered historical data for model adaptation. In online learning scenarios, it is impractical to train a new base classifier for each incoming instance.

For single-model approaches, only a single multi-class classifier is maintained to incorporate novel classes. In the SACCOS [10] framework, the classifier is retrained using recent instances when novel classes emerge. In [28], a matrix sketching method is constructed. The global matrix is adopted to capture the information of the stored instances from novel classes, and the local matrix is extended with additional rows for these classes. The clustering method MINAS [8] constructs micro-clusters by clustering buffered instances and then incorporates them into the model. DISSFCM [4] employs a semi-supervised fuzzy clustering algorithm on the latest chunk data along with the past labeled prototypes to update the set of prototypes. More recently, a prototype-based semi-supervised learning framework [7] is proposed and achieves the state-of-the-art performance in handling evolving data streams with limited labels. This model maintains a dynamic set of prototypes that are extracted at multiple levels of granularity. Prototypes for novel classes are extracted using the current data chunk and then integrated into the model. However, all of these methods require the storage of historical data to extract patterns from novel classes, which is impractical in online learning scenarios. Consequently, both ensemble-model and single-model approaches in existing semi-supervised algorithms are inadequate for handling strict online learning scenarios.

Many online learning frameworks have been proposed to handle the class evolution problem and have demonstrated their effectiveness. In [16], adapted one-versus-all (OVA) decision trees are constructed in a class-based ensemble manner. A new OVA decision tree is initialized and then incorporated when an instance of a novel class arrives. Sun et al. [23] propose a more general OVA framework, CBCE. In CBCE, an innovative activate/deactivate strategy is incorporated to adapt classifiers to class evolution. More recently, the ensemble of ensemble OVA framework (EEOF) proposed in [3] stands out for its novel model adaptation strategy to circumvent the underlying assumption on the prior probabilities of classes which is commonly employed in previous work. This unique feature empowers EEOF to handle a broader spectrum of class evolution scenarios, achieving state-of-the-art classification performance. However, all aforementioned frameworks are supervised. In the context of class evolution under incomplete supervision, these supervised online learning frameworks tend to overlook the unlabeled data of existing classes and excessively focus on novel classes with only a few labeled instances, deteriorating the multi-class classification performance. To address the aforementioned issues, we investigate the problem of online learning with class evolution under incomplete supervision and propose our framework, SOELF, to deliver more accurate predictions.

III. Problem Formulation

Given a continuous data stream $\{\boldsymbol{x}_t\}_1^{+\infty}$, where $\boldsymbol{x}_t \in \mathbb{R}^d$ is the feature vector arriving at time t. The class label of $\boldsymbol{x}_t$ is y_t, which belongs to the class set $\mathcal{Y}_t$ at time t. Here, $\mathcal{Y}_t$ is the set of classes with positive underlying prior probabilities at time t, i.e. $\mathcal{Y}_t = \{c_i | p_t^{(i)} > 0\}$. Class evolution is categorized into three principal forms: 1) class emergence, 2) class disappearance,

and 3) class reoccurrence. A class c_{novel} emerges at time t if $c_{novel} \in \mathcal{Y}_t$ and $c_{novel} \notin \bigcup_{i=1}^{t-1} \mathcal{Y}_i$. A class c_{disp} disappears at time t if $c_{disp} \notin \mathcal{Y}_t$ and $c_{disp} \in \mathcal{Y}_{t-1}$. A class c_{reoc} disappears at time d and reoccurs at time t if $c_{reoc} \in \mathcal{Y}_t$ and $c_{reoc} \notin \bigcup_{i=d}^{t-1} \mathcal{Y}_i$ and $c_{reoc} \in \mathcal{Y}_{d-1}$.

Under incomplete supervision, for a novel class c_{novel}, the expert labels the first q instances in the data stream for model adaptation. In other words, for $\boldsymbol{x}_t$ belonging to c_{novel}, its true label is available if the number of labeled instances for c_{novel} is less than q, i.e. $\sum_{i=1}^{t} \mathbb{1}[y_i == c_{novel}] \leq q$. Additionally, we consider that the expert may continue to label the data stream with a probability of θ for maintenance: if $\sum_{i=1}^{t} \mathbb{1}[y_i == c_{novel}] > q$, the true label is available with a probability of θ, where $0 \leq \theta < 1$. The available true label arrives after the prediction $\hat{y}_t$ is made before time $t+1$.

The task of online learning with class evolution under incomplete supervision is to construct an online classifier $\mathcal{F}$ that is capable of: 1) making an accurate prediction $\hat{y}_t$ for the arriving instance $\boldsymbol{x}_t$, and 2) updating according to $\boldsymbol{x}_t$ or $(\boldsymbol{x}_t, y_t)$, depending on the availability of y_t.

IV. Proposed Framework: SOELF

In the light of the limitations of existing approaches, we propose SOELF to address the problem of online learning with class evolution under incomplete supervision. In this section, we will introduce the details of SOELF.

A. Data Generation Module

Unlike existing semi-supervised approaches to class evolution, which are constraint by the storage of historical data, our method SOELF synergizes a novel data generation module with the state-of-the-art online classifier in handling class evolution, EEOF. The data generation module generates diverse instances from the current single unlabeled instance in an online manner, enabling the classifier to adapt without data retention.

In SOELF, to exploit the abundant unlabeled instances, we maintain an online ensemble clustering model $\mathcal{C}$ alongside the ensemble of ensemble OVA classifiers $\mathcal{F}$ in SOELF for data generation. The online ensemble clustering model $\mathcal{C}$ consists of n base online clustering models for n appeared classes, i.e. $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_n\}$, where $\mathcal{C}_i$ is the base online clustering model for class c_i with a set of k cluster centers, i.e. $\mathcal{C}_i = \{\boldsymbol{center}_{ik}\}_k$. The number of cluster centers k varies across different base online clustering models. The ensemble of ensemble OVA classifiers $\mathcal{F}$ consist of n ensemble of OVA classifiers $\mathcal{F}_i$, i.e. $\mathcal{F} = \{\mathcal{F}_1, ..., \mathcal{F}_n\}$. Each $\mathcal{F}_i$ consists of m base OVA classifiers $\mathcal{F}_{ij}$, i.e. $\mathcal{F}_i = \{\mathcal{F}_{i1}, ..., \mathcal{F}_{im}\}$, where m is the ensemble size. Each base OVA classifier $\mathcal{F}_{ij}$ predicts whether $\boldsymbol{x}_t$ belongs to class c_i (designated as the positive class, $+1$) or not (designated as the negative class, -1). The final classification result $\hat{y}_t$ is assigned to the class with the highest averaged score:

$$\hat{y}_t = \arg\max_i \frac{1}{m} \sum_{j=1}^{m} \mathcal{F}_{ij}(\boldsymbol{x}_t). \tag{1}$$

Given a labeled instances $(\boldsymbol{x}_t, y_t)$, the corresponding base online clustering model for class y_t and the online classifier $\mathcal{F}$ are updated. Note that the online ensemble clustering model $\mathcal{C}$ is also capable of classification by calculating the distances between $\boldsymbol{x}_t$ and the cluster centers. The classification result of $\mathcal{C}$ is assigned to the class with the closest cluster center to $\boldsymbol{x}_t$:

$$\hat{y}_{t,\mathcal{C}} = \arg\max_i \{\min_k (\textbf{Dist}(\boldsymbol{x}_t, \boldsymbol{center}_{ik}))\}^{-1}. \tag{2}$$

The classification performance of $\mathcal{F}$ and $\mathcal{C}$ in the data stream can be partially estimated using the labeled instances. We employ the decaying G-mean $G_{t,\mathcal{M}}$ to monitor the performance of both models:

$$G_{t,\mathcal{M}} = \sqrt[n]{\prod_{i=1}^{n} R_{i,t,\mathcal{M}}}, \tag{3}$$

$$R_{i,t,\mathcal{M}} = \begin{cases} R_{i,t-1,\mathcal{M}} & \text{if } y_t \neq c_i \\ \beta R_{i,t-1,\mathcal{M}} + (1-\beta)\mathbb{1}[y_t == \hat{y}_{t,\mathcal{M}}] & \text{if } y_t = c_i, \end{cases} \tag{4}$$

where $R_{i,t,\mathcal{M}}$ is the decaying recall [25] of the model $\mathcal{M} \in \{\mathcal{F}, \mathcal{C}\}$ for class c_i at time t and $\hat{y}_{t,\mathcal{F}}$ is the prediction of $\mathcal{F}$, i.e. $\hat{y}_t = \hat{y}_{t,\mathcal{F}}$.

Given an unlabeled instance $\boldsymbol{x}_t$, a set of diverse samples with pseudo-labels $\{(\boldsymbol{s}_{t,ij}, \hat{y}_{t,gen})\}$ is generated for the adaptation of the online classifier $\mathcal{F} = \{\mathcal{F}_{ij}\}$. The selection of the pseudo-label $\hat{y}_{t,gen}$ is based on the estimated classification performance of $\mathcal{F}$ and $\mathcal{C}$. The prediction from the model with the higher decaying G-mean is more likely to be chosen as the pseudo-label, i.e. $\hat{y}_{t,gen} = \hat{y}_{t,\mathcal{F}}$ with a probability of $prob_{\mathcal{F}}$ and $\hat{y}_{t,gen} = \hat{y}_{t,\mathcal{C}}$ with a probability of $prob_{\mathcal{C}}$, where $prob_{\mathcal{M}} = G_{t,\mathcal{M}}/(G_{t,\mathcal{F}} + G_{t,\mathcal{C}}), \mathcal{M} \in \{\mathcal{F}, \mathcal{C}\}$.

For each base OVA classifier $\mathcal{F}_{ij}$, the sample $\boldsymbol{s}_{t,ij}$ is generated by the weighted summation of random interpolations between $\boldsymbol{x}_t$ and cluster centers in the corresponding base online clustering model $\mathcal{C}_h$ for the chosen class $\hat{y}_{t,gen}$, i.e. $c_h = \hat{y}_{t,gen}$:

$$\boldsymbol{s}_{t,ij} = \sum_{k=1}^{|\mathcal{C}_h|} \rho_k \boldsymbol{s}_{t,ij}^{(k)}, \tag{5}$$

where $\boldsymbol{s}_{t,ij}^{(k)}$ is the random interpolation between $\boldsymbol{x}_t$ and the k-th cluster center $\boldsymbol{center}_{hk}$ in $\mathcal{C}_h$, and ρ_k is the normalized weight calculated based on the distance between $\boldsymbol{x}_t$ and the k-th cluster center in $\mathcal{C}_h$:

$$\boldsymbol{s}_{t,ij}^{(k)} = \boldsymbol{x}_t + \alpha_{ij}^{(k)} * (\boldsymbol{center}_{hk} - \boldsymbol{x}_t), \tag{6}$$

$$\rho_k = \frac{(\textbf{Dist}(\boldsymbol{x}_t, \boldsymbol{center}_{hk}))^{-1}}{\sum_{a=1}^{|\mathcal{C}_h|} (\textbf{Dist}(\boldsymbol{x}_t, \boldsymbol{center}_{ha}))^{-1}}. \tag{7}$$

Here, the random variable $\alpha_{ij}^{(k)}$ follows a uniform distribution on the interval $[0, 1]$. Interpolations from closer cluster centers will have higher weights. Detailed pseudo-code is displayed in Algorithm 1. Such a set of generated samples introduces diversity and enhances the performance of the online classifier $\mathcal{F}$ in an ensemble of ensemble OVA framework.

Algorithm 1 Generation

Input: Instance $\boldsymbol{x}_t$; Ensemble of Ensemble OVA classifier $\mathcal{F} = \{\mathcal{F}_{ij}\}$; Online ensemble clustering model $\mathcal{C} = \{\mathcal{C}_i\}$; $G_{t,\mathcal{F}}$ and $G_{t,\mathcal{C}}$, decaying G-mean of $\mathcal{F}$ and $\mathcal{C}$ respectively; $\hat{y}_{t,\mathcal{F}}$ and $\hat{y}_{t,\mathcal{C}}$, predictions of $\mathcal{F}$ and $\mathcal{C}$ respectively.
Output: Set of generated samples $\mathbb{S}_t = \{(\boldsymbol{s}_{t,ij}, \hat{y}_{t,gen})\}$; Class index h with $c_h = \hat{y}_{t,gen}$.

$\mathbb{S}_t \leftarrow \varnothing$
// Select pseudo-label based on the decaying G-mean
$prob_{\mathcal{F}} \leftarrow G_{t,\mathcal{F}}/(G_{t,\mathcal{F}} + G_{t,\mathcal{C}})$
$prob_{\mathcal{C}} \leftarrow G_{t,\mathcal{C}}/(G_{t,\mathcal{F}} + G_{t,\mathcal{C}})$
Select a value as $\hat{y}_{t,gen}$ from $\{\hat{y}_{t,\mathcal{F}}, \hat{y}_{t,\mathcal{C}}\}$ with probabilities of $prob_{\mathcal{F}}$ and $prob_{\mathcal{C}}$ respectively.
// Generate features of these samples
$h \leftarrow$**Find**$(i|c_i = \hat{y}_{t,gen})$
for $i \leftarrow 1 : n$ **do**
  for $j \leftarrow 1 : m$ **do**
    Calculate ρ_k and $\boldsymbol{s}_{t,ij}^{(k)}$ for $k \in [1, |\mathcal{C}_h|]$ using Eq. (6) and Eq. (7)
    Calculate $\boldsymbol{s}_{t,ij}$ using Eq. (5)
    $\mathbb{S}_t \leftarrow \mathbb{S}_t \cup \{(\boldsymbol{s}_{t,ij}, \hat{y}_{t,gen})\}$
  end for
end for

B. Inverse-Prior-Ratio Model Adaptation Strategy

Previous online learning algorithms that handle class evolution [16], [23], [3] are all supervised ones and are inadequate to address scenarios of class evolution under incomplete supervision. When novel classes emerge with labeled instances, these algorithms tend to excessively focus on these labeled novel classes, resulting in a deterioration of the multi-class classification performance. Remarkably, the performance of these methods may even deteriorate further as the number of labeled instances for novel classes increases, as evidenced by subsequent experimental results. We attribute this issue to the imbalanced exploitation between classes.

To address this issue, we design the inverse-prior-ratio model adaptation strategy to introduce more instances from existing classes for model adaptation when labeled novel classes emerge. Given a labeled incoming instance $(\boldsymbol{x}_t, y_t)$, we first estimate the prior probability for each class c_i at time t using the time decayed method [26]:

$$\hat{p}_t^{(i)} = \beta \hat{p}_{t-1}^{(i)} + (1-\beta)\mathbb{1}[y_t == c_i], \tag{8}$$

where $\hat{p}_t^{(i)}$ is the estimated prior probability and β is the time decay factor. As the labeled instances of a novel class c_i arrive, $\hat{p}_t^{(i)}$ for this class increases and $\hat{p}_t^{(j)} (j \neq i)$ for other classes decrease. To prevent the online classifier $\mathcal{F}$ from excessively focusing on the class c_i, the set of generated samples for class c_i should be used with a lower probability. In contrast, sets of generated samples for other classes $c_j (j \neq i)$ should be used with higher probabilities. As a consequence, for an unlabeled instance arriving at time t, suppose the pseudo-label of the generated samples is class c_h, i.e. $c_h = \hat{y}_{t,gen}$, we design the adaptation probability as:

$$prob_{adapt} = e^{-K * |\mathcal{F}| * \hat{p}_{t-1}^{(h)}}, \tag{9}$$

where $|\mathcal{F}|$ represents the number of appeared classes in the data stream, K is the hyperparameter defining the adaptation probability in the ideal scenario where all appeared classes have an equal prior probability. The negative sign in the exponent ensures an inverse relationship between $prob_{adapt}$ and the estimated prior probability $\hat{p}_{t-1}^{(h)}$, and the incorporation of $|\mathcal{F}|$ is used to counteract the decrease in prior probability resulting from the increasing number of classes. If these generated samples with pseudo-label $\hat{y}_{t,gen}$ is used for model adaptation, the estimated prior probability for each class is updated using Eq. (8).

Algorithm 2 An Overview of SOELF for data streams

1: Initialize $\mathcal{F}, \mathcal{C}, \hat{\boldsymbol{p}}_t, G_{t,\mathcal{F}}, G_{t,\mathcal{C}}$ from an offline training dataset.
2: **while** Instance $\boldsymbol{x}_t$ arrives **do**
3: // Model Prediction
4: $\hat{y}_t \leftarrow \mathcal{F}(\boldsymbol{x}_t)$ using Eq. (1)
5: // Model Adaptation
6: $\hat{y}_{t,\mathcal{F}} \leftarrow \hat{y}_t$, $\hat{y}_{t,\mathcal{C}} \leftarrow \mathcal{C}(\boldsymbol{x}_t)$ using Eq. (2)
7: **if** true label y_t is available **then**
8: Update $\hat{\boldsymbol{p}}_t$ with the occurrence of class y_t
9: // If a novel class emerges
10: **if** no $\mathcal{F}_i$ in $\mathcal{F}$ is available for y_t **then**
11: Initialize $\mathcal{F}_{novel}$ and $\mathcal{C}_{novel}$ for class y_t
12: $\mathcal{F} \leftarrow \mathcal{F} \cup \mathcal{F}_{novel}$, $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_{novel}$
13: **end if**
14: Update $G_{t,\mathcal{F}}$ and $G_{t,\mathcal{C}}$ with y_t, $\hat{y}_{t,\mathcal{F}}$, $\hat{y}_{t,\mathcal{C}}$ using Eq. (3) and Eq. (4)
15: Update $\mathcal{F}$ and $\mathcal{C}$ with $(\boldsymbol{x}_t, y_t)$ and $\hat{\boldsymbol{p}}_t$
16: **else**
17: $(\mathbb{S}_t, h) \leftarrow$**Generation**$(\boldsymbol{x}_t, \mathcal{F}, \mathcal{C}, G_{t,\mathcal{F}}, G_{t,\mathcal{C}}, \hat{y}_{t,\mathcal{F}}, \hat{y}_{t,\mathcal{C}})$
18: Calculate $prob_{adapt}$ using Eq. (9)
19: **if Uniform**$(0,1) \leq prob_{adapt}$ **then**
20: Update $\hat{\boldsymbol{p}}_t$ with the occurrence of class c_h
21: $\mathcal{F} \leftarrow$ **Updating-Set**$(\mathbb{S}_t, \mathcal{F}, \hat{\boldsymbol{p}}_t)$
22: **end if**
23: **end if**
24: **end while**

Algorithm 3 Updating-Set

Input: Generated set $\mathbb{S}_t = \{(\boldsymbol{s}_{t,ij}, \hat{y}_{t,gen})\}$; Ensemble of ensemble OVA classifier $\mathcal{F} = \{\mathcal{F}_{ij}\}$; Estimated prior probability vector $\hat{\boldsymbol{p}}_t = [\hat{p}_t^{(i)}]$.
Output: Updated $\mathcal{F}$.

for $i \leftarrow 1 : n$ **do**
  for $j \leftarrow 1 : m$ **do**
    if $\hat{y}_{t,gen} == c_i$ **then**
      $w_+ \sim$**Poisson**(1)
      Update $\mathcal{F}_{ij}$ with $w_+ *$**Loss**$(+1, \mathcal{F}_{ij}(\boldsymbol{s}_{t,ij}))$
    else
      $w_- \sim$**Poisson**$(\hat{p}_t^{(i)}/(1-\hat{p}_t^{(i)}))$
      Update $\mathcal{F}_{ij}$ with $w_- *$**Loss**$(-1, \mathcal{F}_{ij}(\boldsymbol{s}_{t,ij}))$
    end if
  end for
end for

C. Overview of SOELF

An overview of SOELF is displayed in Algorithm 2. Given an unlabeled instance, the data generation module generates a set of samples (line 17), and the inverse-prior-ratio model adaptation strategy is employed (line 18∼22). Details of the model adaptation process of $\mathcal{F}$ with the generated samples (line 21) is displayed in Algorithm 3. The model adaptation of $\mathcal{F}$ with a labeled instance (line 15) is similar to the Algorithm 3 with every instance in the set $\mathbb{S}_t$ replaced by the instance $(\boldsymbol{x}_t, y_t)$. The loss weights w_+ and w_- are designed as in [3] to handle the problem of dynamic class imbalance.

V. Experiments

In this section, SOELF is compared with state-of-the-art methods on synthetic and real-world data streams under incomplete supervision. An ablation study is conducted to highlight the significance of different components in SOELF.

A. Experimental Settings

1) Data Streams: We utilize three tabular datasets, namely Letter Recognition[1] (16 attributes), Statlog (Landsat Satellite)[2] (36 attributes), and Covertype[3] (54 attributes), as well as an image dataset, MNIST[4], for generating synthetic data streams on class evolution scenarios of 1) class emergence and 2) class disappearance and reoccurrence under diverse cases of incomplete supervision. We adopt the same approach as in the former work [23]. In class emergence scenarios, we select two classes as existing classes and one class as the emerging class. In scenarios of class disappearance and reoccurrence, we select three classes as existing class, with one class disappearing for a while and then reoccurring later.

We further investigate SOELF on six real-world data streams that exhibit different types of class evolution. These data streams include four preprocessed data streams of the TwitterCrawl data stream [18], namely Tweet Stream - A/B/C/20 classes in [23], Poker-hand [5], and KDDCUP99 [24]. The key characteristics of these real-world data streams are displayed in Table I.

TABLE I: Characteristics of real-world data streams.

Data Stream	Instances	Features	Classes	Class Evolution Types
Tweet Stream - A	39600	242	3	Class Emergence
Tweet Stream - B	15004	247	3	Class Disappearance and Reoccurrence
Tweet Stream - C	68750	242	4	Class Emergence
Tweet Stream - 20 classes	143381	524	20	Class Emergence, Disappearance and Reoccurrence
Poker-hand	829201	25	10	Class Emergence, Disappearance and Reoccurrence
KDDCUP99	494021	117	23	Class Emergence, Disappearance and Reoccurrence

Besides selecting various data streams that involve class evolution, we thoroughly examine algorithms under diverse scenarios of incomplete supervision. The number of labeled instances q for a novel class is chosen from $\{10, 100\}$. The probability of labeling instances θ in the data stream is chosen from $\{0, 0.05, 0.10\}$, where $\theta = 0$ means that no instances in the data stream are labeled except for the first q instances for a novel class. These settings provide six incomplete supervision scenarios for each data stream mentioned above.

2) Comparison Algorithms: Five algorithms are chosen for comparison:

- CBCE [23] / EEOF [3] - Popular and state-of-the-art supervised online learning methods for handling class evolution respectively.
- CBCE-v / EEOF-v - Variants of CBCE and EEOF. In CBCE/EEOF, a class is excluded from classification if its estimated prior probability falls below a threshold, which is misleading under incomplete supervision. A class may be considered disappeared simply because the model has not received its labeled data in a long time. For a fair comparison, we compare our method with CBCE-v and EEOF-v, in which no classes are considered disappeared.
- PT [7] - The state-of-the-art semi-supervised method for handling class evolution. This algorithm processes data chunk-by-chunk, requiring the storage of historical data. In [7], this method does not have a short name. For convenience, we refer to this method as PT (prototype).

Supervised online learning methods CBCE/EEOF/CBCE-v/EEOF-v adapt to the incomplete supervision scenarios by utilizing only labeled instances. The semi-supervised method PT operates by first making predictions and then updating on a data chunk. The size of the initial offline training dataset is set to 1000 for all algorithms. In CBCE/EEOF/CBCE-v/EEOF-v/SOELF, the same base OVA classifier, Kernelized Logistic Regression (KLR) [17], is employed with the same hyperparameters for a fair comparison. The hyperparameters of KLR are tuned on the initial offline dataset following methods suggested in previous work [23]. Specifically, a 5-folder cross-validation is employed to determine the the learning rate η and the regularization parameter λ. The kernel width σ is set to the 5th percentile of the pairwise distances between all instances. The ensemble size is set to 10 for EEOF/EEOF-v/SOELF as in the former work [3]. Many online clustering algorithms, such as CluStream [1], DenStream [2], and DBSTREAM [12], can be integrated into SOELF. In this paper, we select DenStream as the base online clustering model. The hyperparameters for DenStream are configured as follows: the decaying factor $\beta_{DenStream}$ is set to 0.01, the stream speed v is set to 1 to process instances one-by-one, the number of instances for initialization is set to 4, ϵ that defines the ϵ neighborhood is set to the averaged value of the μ-th distances of instances during initialization. The hyperparameter K in the inverse-prior-ratio model adaptation strategy in SOELF is set to 3.3. For PT, the chunk size is set to 1000, the same as the size of initial offline dataset. Other hyperparameters not mentioned in this section are set to their suggested values in their corresponding papers.

3) Evaluation Metrics: We utilize the G-mean metric using a sliding window to monitor the performance of multi-class classification along the data stream. The G-mean values are averaged to assess the overall performance of an algorithm on the data stream, as described in [3]. The G-mean using a sliding window is defined as $G_W(t) = \textbf{G-mean}(\{(y_i, \hat{y}_i)\}_{t-W+1}^{t})$, where W is the sliding window size set to 200. The average of G-mean using a sliding window with a length of N is defined as $\frac{1}{N-W+1}\sum_{i=W}^{N} G_W(i)$. We perform 10 independent runs for each algorithm on each data stream. The significance of the difference between algorithms is examined using the Wilcoxon rank-sum test [11].

B. Experimental Analysis

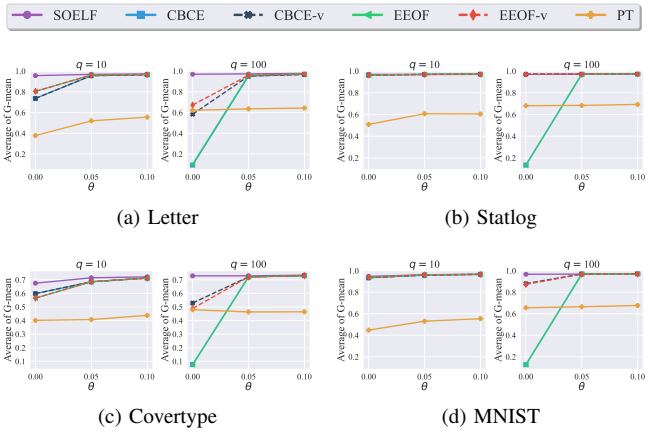
1) Performance on Synthetic Data Streams: The performance of algorithms on synthetic and real-world data streams with the fewest labeled instances for novel classes, i.e. $q=10$ and $\theta=0$, is displayed in Table II. The Win/Tie/Loss results indicate that SOELF achieves significantly superior performance in most cases, while in a few cases it shows either comparable performance or a slight degradation. Specifically, for synthetic data streams with class emergence, SOELF exhibits substantial improvements over the best competing results, achieving gains of 18.9% on Letter and 12.8% on Covertype. Fig. 1

[1]https://archive.ics.uci.edu/dataset/59/letter+recognition
[2]https://archive.ics.uci.edu/dataset/146/statlog+landsat+satellite
[3]https://archive.ics.uci.edu/dataset/31/covertype
[4]http://yann.lecun.com/exdb/mnist/

TABLE II: Performance (mean/std) averaged over 10 runs on synthetic and real-world data streams with the fewest labeled instances ($q = 10$ and $\theta = 0$). The highest mean result is highlighted in boldface.

Data Stream	SOELF	CBCE	CBCE-v	EEOF	EEOF-v	PT
Letter	**0.9561/0.0057**	0.7360/0.0670	0.7360/0.0670	0.8042/0.0459	0.8042/0.0459	0.3788/0.0000
Statlog	**0.9692/0.0024**	0.9628/0.0074	0.9628/0.0074	0.9650/0.0029	0.9650/0.0029	0.5090/0.0000
Covertype	**0.6751/0.0140**	0.5983/0.0386	0.5983/0.0386	0.5652/0.0213	0.5652/0.0213	0.4014/0.0000
MNIST	**0.9473/0.0072**	0.9345/0.0080	0.9345/0.0080	0.9396/0.0019	0.9396/0.0019	0.4479/0.0000
Tweet Stream - A	**0.5859/0.0101**	0.0940/0.0039	0.0940/0.0039	0.1549/0.0255	0.1549/0.0255	0.2940/0.0000
Tweet Stream - B	**0.6326/0.0081**	0.5000/0.0420	0.5000/0.0420	0.4893/0.0345	0.4893/0.0345	0.4593/0.0000
Tweet Stream - C	**0.4992/0.0120**	0.0962/0.0039	0.0962/0.0039	0.1049/0.0121	0.1049/0.0121	0.2614/0.0000
Tweet Stream - 20 classes	0.0001/0.0000	0.0000/0.0000	0.0000/0.0000	0.0000/0.0000	0.0000/0.0000	**0.0006/0.0000**
Poker-hand	**0.0758/0.0082**	0.0314/0.0271	0.0341/0.0298	0.0218/0.0182	0.0225/0.0183	0.0334/0.0000
KDDCUP99	**0.6381/0.0109**	0.0512/0.0000	0.5835/0.0000	0.5544/0.0006	0.5818/0.0006	0.0870/0.0000
Win/Tie/Loss	-	0/1/9	0/1/9	0/0/10	0/0/10	1/0/9

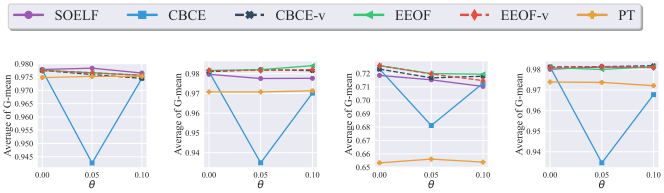
* Win/Tie/Loss counts the number of data streams where the competing algorithm significantly outperforms, shows no significant difference, or performs significantly worse than SOELF (Wilcoxon rank-sum test at 95% confidence level).



(a) Letter (b) Statlog

(c) Covertype (d) MNIST

Fig. 1: Performance of algorithms on synthetic data streams with class emergence under various scenarios of incomplete supervision.



(a) Letter (b) Statlog (c) Covertype (d) MNIST

Fig. 2: Performance of algorithms on synthetic data streams with class disappearance and reoccurrence under various scenarios of incomplete supervision.

further presents the performance of algorithms on synthetic data streams with class emergence under various scenarios of incomplete supervision. The performance of CBCE and EEOF drops as the number of labeled instance for novel classes q increases from 10 to 100 for scenarios with $\theta = 0$. We attribute this to 1) the misjudgment of disappeared classes, and 2) the adaptation of the classifier excessively focusing on the novel classes. CBCE-v and EEOF-v are variants in which no existing classes are considered disappeared, and they perform better than CBCE and EEOF in scenarios with $q = 100$ and $\theta = 0$. However, a decline in performance is still observed as q increases due to the second reason, as shown in Fig. 1(a)(c)(d). The semi-supervised method PT requires a collection of data for adapting to novel classes. As a result, PT cannot be capable of classifying new class data upon the class emergence as online learning methods do. Consequently, its performance falls behind the online learning methods almost in all cases. The proposed online learning method, SOELF, addresses the aforementioned issues. In the scenarios with $\theta = 0$, the performance of SOELF remains stable and improves with increasing q. In addition, the performance of SOELF is considerably stable as θ changes. The superiority of SOELF is particularly pronounced when $\theta = 0$. We attribute these to the balanced exploitation of unlabeled instances achieved by the data generation module and the inverse-prior-ratio model adaptation strategy in SOELF. In summary, SOELF is capable of effectively managing various scenarios of incomplete supervision regarding class emergence, and outperforms other methods.

We further conduct experiments on synthetic data streams with class disappearance and reoccurrence, as shown in Fig. 2. Since there are no novel classes in this form of class evolution, the parameter q is not considered. Only the parameter θ varies. Our method, SOELF, is specifically designed to manage the model adaptation of an online classifier on class emergence, rather than class disappearance and reoccurrence, which lie outside the targeted scenarios of SOELF. Still, SOELF performs comparably in cases such as Letter and MNIST, exhibiting a slight decrease in performance (less than 0.01 in the average of G-mean) compared to the best performance achieved by other algorithms in cases such as Statlog and Covertype.

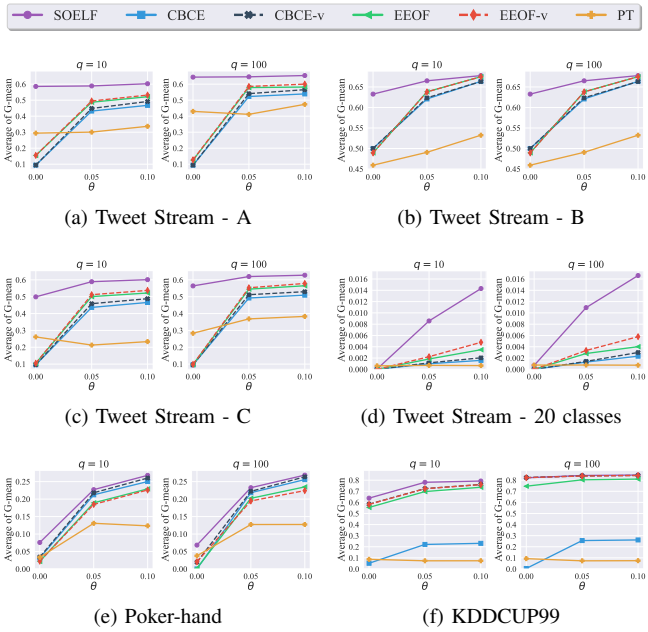
2) Performance on Real-world Data Streams: As observed in Table II, SOELF exhibits substantial improvements over the best competing results on most cases, achieving gains of 99.3% on Tweet Stream - A, 26.5% on Tweet Stream - B, 91.0% on Tweet Stream - C, 122.3% on Poker-hand, and 9.4% on KDDCUP99. The performance of algorithms on real-world data streams under various scenarios of incomplete supervision is further depicted in Fig. 3. SOELF demonstrates superior

TABLE III: Performance (mean/std) averaged over 10 runs of the ablation study on synthetic and real-world data streams with the fewest labeled instances ($q = 10$ and $\theta = 0$). The highest mean result is highlighted in boldface.

Data Stream	SOELF	SOELF-GenF	SOELF-GenC	SOELF-Inverse
Letter	**0.9561/0.0057**	0.9187/0.0388	0.9414/0.0187	0.9450/0.0068
Statlog	**0.9692/0.0024**	0.9671/0.0063	0.9503/0.0130	0.9291/0.0066
Covertype	**0.6751/0.0140**	0.5796/0.0507	0.6590/0.0181	0.6677/0.0052
MNIST	0.9473/0.0072	0.9447/0.0064	0.9475/0.0060	**0.9631/0.0019**
Tweet Stream - A	**0.5859/0.0101**	0.3963/0.0618	0.5467/0.0116	0.5620/0.0127
Tweet Stream - B	0.6326/0.0081	0.5688/0.0241	0.6163/0.0056	**0.6378/0.0053**
Tweet Stream - C	0.4992/0.0120	0.3390/0.0209	0.4764/0.0133	**0.5077/0.0039**
Tweet Stream - 20 classes	**0.0001/0.0000**	0.0000/0.0000	0.0000/0.0000	0.0000/0.0000
Poker-hand	0.0758/0.0082	**0.0905/0.0217**	0.0397/0.0027	0.0374/0.0041
KDDCUP99	0.6381/0.0109	0.5966/0.0066	**0.6510/0.0098**	0.5863/0.0125
Win/Tie/Loss	-	0/3/7	1/2/7	1/3/6

* Win/Tie/Loss counts the number of data streams where the variant significantly outperforms, shows no significant difference, or performs significantly worse than SOELF (Wilcoxon rank-sum test at 95% confidence level).

performance across nearly all scenarios of class evolution under incomplete supervision, except for the Tweet Stream - 20 classes with $\theta = 0$, where all algorithms fail to yield satisfactory predictions. As mentioned in [23], Tweet Stream data streams involve concept drift. Although SOELF shows comparable or slightly worse performance on synthetic data streams with class disappearance and reoccurrence, in Tweet Stream - B, where the same form of class evolution exists and concept drift is involved, SOELF still exhibits significant improvements due to its continuous exploitation on unlabeled instances to adapt classifier to concept drift, as shown in Fig. 3(b). Other online learning methods, namely CBCE, EEOF, CBCE-v, EEOF-v, exhibit unstable performance with varying θ, and their performance decreases with the increase of q in several cases, such as Tweet Stream - A and Poker-hand with $\theta = 0$, as shown in Fig. 3(a)(e). SOELF produces more stable and more accurate predictions in these data streams. For KDDCUP99, as shown in 3(f), SOELF shows its privilege particularly when $q = 10$. While PT performs stably in all cases, this method still suffers from the delay of model adaptation on novel classes. Experimental results on real-world data streams featuring ample novel classes, substantial data volume, and high-dimensional data, effectively showcase the efficacy of our proposed framework SOELF in addressing class evolution under incomplete supervision.



(a) Tweet Stream - A (b) Tweet Stream - B

(c) Tweet Stream - C (d) Tweet Stream - 20 classes

(e) Poker-hand (f) KDDCUP99

Fig. 3: Performance of algorithms on real-world data streams under various scenarios of incomplete supervision.

3) Ablation Study: Three variants of SOELF are designed for comparison. In SOELF-GenF and SOELF-GenC, the unlabeled instance $\boldsymbol{x}_t$ is assigned with the pseudo-label $\hat{y}_{t,\mathcal{F}}$ and $\hat{y}_{t,\mathcal{C}}$ respectively for model adaptation to examine the sufficiency of the data generation module. A common approach would be to adapt the online classifier $\mathcal{F}$ using generated samples on all unlabeled instances. In SOELF-Inverse, we set $prob_{adapt} = 1$ to examine the sufficiency of the inverse-prior-ratio model adaptation strategy.

We present the comparison results for the case with the fewest labeled instances due to page limitation, i.e. $q = 10$ and $\theta = 0$, as displayed in Table III. The Win/Tie/Loss counts the number of cases where these variants significantly outperform, show no significantly difference, or perform significantly worse, compared with SOELF respectively. Upon comparing SOELF-GenF, SOELF-GenC with SOELF, it is noteworthy that although SOELF-GenC outperforms SOELF slightly in 1 case, SOELF provides significantly more accurate predictions in 7 cases, particularly evident in data streams such as Letter, Covertype, and Tweet Stream - A/B/C. These improvements are attributed to the diverse generated samples with reliable pseudo-labels facilitated by the integration of the decaying G-mean metric. Upon comparing SOELF-Inverse with SOELF, SOELF exhibits slightly worse performance in 1 cases, but significantly outperforms in 6 cases, particularly in data streams such as Letter, Statlog, Poker-hand, and KDDCUP99. Unlike adapting online classifier with every unlabeled instance in SOELF-Inverse, the inverse-prior-ratio model adaptation strategy in SOELF ensures a balanced exploitation of unlabeled instances across classes on model adaptation, thereby preventing the online classifier from excessively focusing on any particular class. Therefore, the default framework of SOELF

is generally recommended for addressing diverse scenarios of class evolution under incomplete supervision.

VI. Conclusion

In this work, we propose a novel framework, SOELF, to address the problem of online learning with class evolution under incomplete supervision. To the best of our knowledge, this is the first study to investigate this problem. Unlike existing semi-supervised approaches, our algorithm operates in an online manner without the need to store historical data. SOELF integrates a data generation module based on an online ensemble clustering model and the decaying G-mean metric, to exploit unlabeled instances and generate diverse samples with reliable pseudo-labels for model adaptation. Additionally, an inverse-prior-ratio model adaptation strategy is introduced to prevent the online classifier from overfitting to novel classes with labeled instances, balancing exploitation between classes. Experimental results on both synthetic and real-world data streams across various scenarios demonstrate the superiority of SOELF on handling class evolution under incomplete supervision.

Acknowledgment

This work was supported in part by Research and Application on Intelligent Highway Inspection, Diagnosis, and Maintenance Decision-Making under Grant K240401HF, and in part by an internal grant of Lingnan University.

References

[1] Aggarwal, C.C., Yu, P.S., Han, J., Wang, J.: A framework for clustering evolving data streams. In: Proceedings 2003 VLDB Conference, pp. 81–92. Morgan Kaufmann (2003)

[2] Cao, F., Estert, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: Proceedings of the 2006 SIAM international conference on data mining. pp. 328–339. SIAM (2006)

[3] Cao, Z., Zhang, S., Lin, C.T.: Online ensemble of ensemble OVA framework for class evolution with dominant emerging classes. In: 2023 IEEE International Conference on Data Mining (ICDM). pp. 968–973 (2023)

[4] Casalino, G., Castellano, G., Mencar, C.: Incremental adaptive semi-supervised fuzzy clustering for data stream classification. In: 2018 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS). pp. 1–7 (2018)

[5] Cattral, R., Oppacher, F., Deugo, D.: Evolutionary data mining with automatic rule generalization. Recent Advances in Computers, Computing and Communications **1**(1), 296–300 (2002)

[6] Din, S.U., Shao, J., Kumar, J., Mawuli, C.B., Mahmud, S.M.H., Zhang, W., Yang, Q.: Data stream classification with novel class detection: a review, comparison and challenges. Knowledge and Information Systems **63**(9), 2231–2276 (2021)

[7] Din, S.U., Ullah, A., Mawuli, C.B., Yang, Q., Shao, J.: A reliable adaptive prototype-based learning for evolving data streams with limited labels. Information Processing & Management **61**(1), 103532 (2024)

[8] de Faria, E.R., Ponce de Leon Ferreira Carvalho, A.C., Gama, J.: MINAS: multiclass learning algorithm for novelty detection in data streams. Data Mining and Knowledge Discovery **30**(3), 640–680 (2016)

[9] Fontanel, D., Cermelli, F., Mancini, M., Caputo, B.: On the challenges of open world recognition under shifting visual domains. IEEE Robotics and Automation Letters **6**(2), 604–611 (2021)

[10] Gao, Y., Chandra, S., Li, Y., Khan, L., Bhavani, T.: SACCOS: a semi-supervised framework for emerging class detection and concept drift adaption over data streams. IEEE Transactions on Knowledge and Data Engineering **34**(3), 1416–1426 (2022)

[11] Gibbons, J.D., Chakraborti, S.: Nonparametric Statistical Inference, pp. 977–979. Springer Berlin Heidelberg (2011)

[12] Hahsler, M., Bolaños, M.: Clustering data streams based on shared density between micro-clusters. IEEE Transactions on Knowledge and Data Engineering **28**(6), 1449–1461 (2016)

[13] Haque, A., Khan, L., Baron, M.: Semi supervised adaptive framework for classifying evolving data stream. In: Advances in Knowledge Discovery and Data Mining. pp. 383–394 (2015)

[14] Haque, A., Khan, L., Baron, M.: SAND: Semi-supervised adaptive novel class detection and classification over data stream. Proceedings of the AAAI Conference on Artificial Intelligence **30**(1) (2016)

[15] Haque, A., Khan, L., Baron, M., Thuraisingham, B., Aggarwal, C.: Efficient handling of concept drift and concept evolution over stream data. In: 2016 IEEE 32nd International Conference on Data Engineering (ICDE). pp. 481–492 (2016)

[16] Hashemi, S., Yang, Y., Mirzamomen, Z., Kangavari, M.: Adapted one-versus-all decision trees for data stream classification. IEEE Transactions on Knowledge and Data Engineering **21**(5), 624–637 (2009)

[17] Kivinen, J., Smola, A., Williamson, R.: Online learning with kernels. IEEE Transactions on Signal Processing **52**(8), 2165–2176 (2004)

[18] Li, R., Wang, S., Deng, H., Wang, R., Chang, K.C.C.: Towards social user profiling: unified and discriminative influence model for inferring home locations. In: KDD. pp. 1023–1031 (2012)

[19] Masud, M.M., Gao, J., Khan, L., Han, J., Thuraisingham, B.: Classification and novel class detection in data streams with active mining. In: Advances in Knowledge Discovery and Data Mining. pp. 311–324 (2010)

[20] Mohammadi, M., Al-Fuqaha, A., Sorour, S., Guizani, M.: Deep learning for IoT big data and streaming analytics: a survey. IEEE Communications Surveys & Tutorials **20**(4), 2923–2960 (2018)

[21] Mu, X., Ting, K.M., Zhou, Z.H.: Classification under streaming emerging new classes: a solution using completely-random trees. IEEE Transactions on Knowledge and Data Engineering **29**(8), 1605–1618 (2017)

[22] Mustafa, A.M., Ayoade, G., Al-Naami, K., Khan, L., Hamlen, K.W., Thuraisingham, B., Araujo, F.: Unsupervised deep embedding for novel class detection over data stream. In: 2017 IEEE International Conference on Big Data (Big Data). pp. 1830–1839 (2017)

[23] Sun, Y., Tang, K., Minku, L.L., Wang, S., Yao, X.: Online ensemble learning of data streams with gradually evolved classes. IEEE Transactions on Knowledge and Data Engineering **28**(6), 1532–1545 (2016)

[24] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. pp. 1–6 (2009)

[25] Wang, S., Minku, L.L., Yao, X.: A learning framework for online class imbalance learning. In: 2013 IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL). pp. 36–45 (2013)

[26] Wang, S., Minku, L.L., Yao, X.: Online class imbalance learning and its applications in fault detection. International Journal of Computational Intelligence and Applications **12**(04), 1340001 (2013)

[27] Wang, S., Minku, L.L., Yao, X.: A systematic study of online class imbalance learning with concept drift. IEEE Transactions on Neural Networks and Learning Systems **29**(10), 4802–4821 (2018)

[28] Zhang, Z., Li, Y., Zhang, Z., Jin, C., Gao, M.: Adaptive matrix sketching and clustering for semisupervised incremental learning. IEEE Signal Processing Letters **25**(7), 1069–1073 (2018)

[29] Zheng, X., Li, P., Hu, X., Yu, K.: Semi-supervised classification on data streams with recurring concept drift and concept evolution. Knowledge-Based Systems **215**, 106749 (2021)

[30] Zhou, Z.H.: A brief introduction to weakly supervised learning. National Science Review **5**(1), 44–53 (2017)