“© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Online Ensemble of Ensemble OVA Framework for Class Evolution with Dominant Emerging Classes

Zhi Cao[1,2], Shuyi Zhang[1,3], Chin-Teng Lin[2]
[1]*Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China*
[2]*CIBCI Lab, School of Computer Science, University of Technology, Sydney, Australia*
[3]*CERCIA, School of Computer Science, University of Birmingham, Birmingham, UK*

***Abstract*—In real-world data stream mining, the composition of classes undergoes unpredictable changes, giving rise to the challenge of class evolution, encompassing class emergence, disappearance, and reoccurrence. However, most existing approaches require the storage of past data to adapt their model. While some studies have focused on online learning approaches, they are built on an underlying assumption that the number of instances in any single class is consistently less than the sum of other classes. This assumption becomes invalid when a class emerges with a dominant amount, e.g., news about a pandemic outbreak, harming the performance of existing methods. In this paper, we thoroughly investigate this scenario and propose a novel online ensemble of ensemble one-versus-all framework (EEOF) to handle class evolution adaptively. The novel ensemble of ensemble architecture boosts diversity in each one-versus-all classifier. A novel adaptive model adaptation method is also designed to balance the error feedback between the emerging class and the other classes. A confidence-triggered fallback mode is integrated to prevent performance drop due to a wrong decision regarding class disappearance. Experimental studies are conducted on both synthetic and real-world data streams to show that our method achieves higher accuracy in diverse class evolution scenarios compared with the state-of-the-art method, particularly when classes emerge with dominant amounts.**

***Index Terms*—data stream mining, online learning, class evolution, ensemble model**

I. Introduction

The non-stationary nature of the data stream brings numerous challenges for the classification task, such as class evolution, indicating that the composition of classes is not confined to a fixed set [1]. As the environment evolves, novel classes may emerge, existing classes may disappear, and disappeared classes may reoccur. Another challenge is the constant generation of data in large quantities and at high velocities. This poses significant demands on memory constraints and processing speed. Online learning, characterized by single-pass processing of each instance and the absence of past data storage, is well-suited to meet these requirements [2]. The classifier needs to track and adapt to the evolution of classes from a single available data example.

However, most research on class evolution either considers the arrival of data in a chunk-by-chunk manner [3], [4] or requires a buffer module to store past instances [5]–[7]. Very few studies are conducted with online learning frameworks on class evolution [8], [9]. These studies construct their models using an ensemble of one-versus-all (OVA) classifiers based on the underlying assumption that the number of any single class is always smaller than the sum of other classes, hence only consider under-sampling updating strategies. However, this assumption is invalid when a novel class emerges with a dominant number of instances. This one-sided under-sampling strategy causes the model to receive more error feedback from the dominant emerging class than the existing classes, leading to inaccurate predictions.

Y. Sun et al. [9] further investigated class disappearance and recurrence in class evolution. An activation/deactivation strategy is proposed to minimize the interference from disappeared classes. However, this approach solely relies on a stationary probability threshold to determine the disappearance of a class. In scenarios where a class emerges with a dominant number of instances, existing classes that generate instances with minimal probabilities are inclined to be mistakenly regarded as disappeared classes, resulting in unreliable predictions.

In this paper, a novel ensemble of ensemble one-versus-all framework (EEOF) is proposed to address class evolution in the online learning procedure. The ensemble of ensemble architecture introduces diversity to enhance the performance of OVA classifiers. We attribute the failure of the previous OVA framework primarily to the improper assumption, and design a novel adaptive model adaptation method to generate dynamic weights for the learning of each ensemble OVA classifier, ensuring a balanced error feedback between the dominant emerging classes and other existing classes. To prevent existing classes being wrongly regarded as disappeared classes, a confidence-triggered fallback mode is designed. This mode temporarily incorporates the inactivated OVA classifiers in classification when the activated OVA classifiers fail to provide reliable predictions. Experiments on synthetic and real-world data streams demonstrate the superiority of our framework.

Our main contributions are as follows: i) We propose a novel framework, EEOF, to handle class evolution in the online learning procedure. This framework injects diversity within the inner OVA classifiers. ii) We design an adaptive model adaptation method to balance the error feedback between the emerging class and the other classes, achieving better performance on scenarios where classes emerge with dominant amounts, as well as other diverse class evolution scenarios. iii) We design a novel confidence-triggered fallback mode to resolve the limitation of the fixed disappearance threshold strategy adopted by existing methods, achieving more accurate predictions.

II. Related work

Approaches for novel class adaptation can be categorized into the single-model and the ensemble-model approaches. For single-model approaches, a single classifier is adapted to novel classes in the data stream. Methods such as MINAS [5] and EMC [4] construct micro-clusters on buffered instances and incorporate them into the model. The efficacy of these clustering algorithms diminishes in online learning scenarios. Neural network-based models, as proposed in [10] and [7], are designed to handle high-dimensional data streams. However, both methods rely on a buffer module to update their neural networks.

For ensemble models, a commonly employed approach is to replace the outdated base classifier with a newly trained base classifier on the latest data. In C&NCBM [11], the least accurate base classifier is replaced with a newly trained one from the latest chunk. Some work builds base classifiers using a dynamic window size instead of fixed-size data blocks, as seen in SAND [6] and ESCR [12]. SENCForest [13] introduces two levels of mechanisms for adaptation. The buffered data is used either to grow existing trees or train a new one. Class-based ensemble models, such as CLAM [14] and SCANR [15], update with new per-class classifiers trained on incoming data chunks. However, all above approaches rely on stored data to adapt model to novel classes. In online learning scenarios, only the current single instance is available. It is impractical to build an entirely new base classifier for each incoming instance.

Only a few studies focus on online learning. Hashemi et al. [8] built OVA decision trees in a class-based ensemble manner. In CBCE [9], an activate/deactivate strategy is designed to adapt model to class evolution. These methods assume that the positive class (a specific class) is always the minority for each OVA model, and use one-side under-sampling strategies on the negative class (other classes). However, this assumption is invalid when a class emerges with a dominant amount, leading to unsatisfactory predictions. In CBCE [9], a predefined fixed threshold is used to determine class disappearance. When classes emerge with dominant amounts, an existing class will occur with low probability. This class may be mistakenly considered as a disappeared class, resulting in false predictions.

III. Problem formulation

Given a data stream $\{(\boldsymbol{x}_t, y_t)\}_1^{+\infty}$, where $\boldsymbol{x}_t$ is the feature vector arriving at time t and $y_t \in \mathbb{C}_t$ is the associated class label. Each class c in $\mathbb{C}_t$ has positive prior probability at time t, i.e. $p_t(c) > 0$. In online learning, only one example $\boldsymbol{x}_t$ is available at time t. The true label y_t of $\boldsymbol{x}_t$ arrives after the prediction $\hat{y}_t$ is made before time $t+1$.

In the data stream, the set of classes $\mathbb{C}_t$ may change over time, referred to as class evolution. Based on how classes evolve, class evolution is categorized into 3 principal forms: 1) class emergence, 2) class disappearance, and 3) class reoccurrence. A class c_{novel} emerges at time t if $c_{novel} \in \mathbb{C}_t$ and $c_{novel} \notin \bigcup_{i=1}^{t-1} \mathbb{C}_i$. A class $c_{disappear}$ disappears at time t if $c_{disappear} \notin \mathbb{C}_t$ and $c_{disappear} \in \mathbb{C}_{t-1}$. A class $c_{reoccur}$ disappears at time d and reoccurs at time t if $c_{reoccur} \in \mathbb{C}_t$ and $c_{reoccur} \notin \bigcup_{d}^{t-1} \mathbb{C}_i$ and $c_{reoccur} \in \bigcup_{i=1}^{d-1} \mathbb{C}_i$.

The task of online learning with class evolution is to construct an online classifier $\mathcal{F}$ that is capable of 1) making accurate prediction $\hat{y}_t$ for arriving example $\boldsymbol{x}_t$ at time t, and 2) updating after receiving the true label y_t.

IV. Proposed algorithm

A. Ensemble of ensemble OVA framework

In existing OVA frameworks, k OVA classifiers are maintained for the k-class classification problem. Each OVA classifier serves as a binary classifier, predicting whether an instance $\boldsymbol{x}_t$ belongs to a specific class c_i (designated as the positive class, $+1$), or to the remaining $k-1$ classes (designated as the negative class, -1). Given $\mathcal{F}_i$ providing the prediction score for class c_i, the instance x_t is classified into the class estimated with the highest prediction score:

$$\hat{y}_t = argmax_i \mathcal{F}_i(\boldsymbol{x}_t). \tag{1}$$

In this work, we maintain an ensemble of OVA classifiers, denoted as $\mathcal{E}_i$, for each class c_i to introduce diversity among the inner OVA classifiers and achieve a more accurate estimation of the prediction score. We name this approach to be an ensemble of ensemble OVA framework, denoted as $\mathcal{O} = \{\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_n\}$, where ensemble OVA model $\mathcal{E}_i = \{\mathcal{F}_{i1}, \mathcal{F}_{i2}, ..., \mathcal{F}_{im}\}$. Here, n is the total number of classes that appear in the data stream, and m is the ensemble size. The prediction score for class c_i is estimated by the average of prediction scores across all base classifiers in $\mathcal{E}_i$. The classification result $\hat{y}_t$ is determined as:

$$\hat{y}_t = argmax_i \frac{1}{m} \sum_{j=1}^{m} \mathcal{F}_{ij}(\boldsymbol{x}_t), \tag{2}$$

B. Model adaptation

In the previous work such as [8] and [9], it is assumed that the positive class represents the minority class, while the negative class represents the majority class. This assumption is valid only in limited stationary scenarios. However, in dynamic real-world data streams, instances from different classes do not arrive in an uniform and predictable manner. When a class emerges with a dominant amount, under-sampling strategies applied on the negative class will make the new OVA classifier for the emerging class to be trained predominantly on this class, leading to an overestimated prediction.

To address this issue, we propose an adaptive cost-sensitive model adaptation method to balance the error feedback between the dominant emerging class and the other classes. In the ensemble OVA model for the emerging class, the loss from the negative class is deliberately overweighted. In the ensemble OVA models for other classes, this dominant emerging class is treated as part of the negative class, and the loss is underweighted. This procedure relies on the dynamic prior probability p_t^i of each class c_i at time t, which is estimated by [16]:

$$w_t^i = \beta w_{t-1}^i + (1-\beta)\mathbb{1}[y_t == c_i], \tag{3}$$

where w_t^i is the estimated prior probability, β is the time decay factor. For each ensemble OVA model $\mathcal{E}_i$, assuming the loss weight for the positive class is set to 1 ($l_+ = 1$), the loss weight for the negative class is adjusted as follows:

$$l_- = l_+ w_t^i/(1 - w_t^i) = w_t^i/(1 - w_t^i). \tag{4}$$

This adaptive method is not only designed for the considered scenario of dominant emerging class, but also applicable in various other scenarios exhibiting similar imbalance issues.

In online learning, model is adapted with only a single instance. To introduce the diversity within the ensemble OVA model $\mathcal{E}_i$, we generate the loss weight using the Poisson distribution, a commonly used technique in online learning [17]. The detailed pesudo-code is displayed in Algorithm 1.

Algorithm 1 Adaptation

Input: Instance $(\boldsymbol{x}_t, y_t)$; EEOF Model $\mathcal{O} = \{\mathcal{E}_i\}$, where $\mathcal{E}_i = \{\mathcal{F}_{ij}\}$; Score matrix $\hat{Y}_t = [\hat{y}_{t,ij}]$, where $\hat{y}_{t,ij} = \mathcal{F}_{ij}(\boldsymbol{x}_t)$; Estimated prior probability vector $\mathcal{W}_t = [w_t^i]$.
Output: Updated EEOF model $\mathcal{O}$.

for $i = 1 : n$ **do**
  for $j = 1 : m$ **do**
    if $y_t == c_i$ **then**
      $l_+ \sim Poisson(1)$
      if $l_+ > 0$ **then**
        $Update\ \mathcal{F}_{ij}$ with $l_+ * Loss(+1, \hat{y}_{t,ij})$
      end if
    else
      $l_- \sim Poisson(w_t^i/(1 - w_t^i))$
      if $l_- > 0$ **then**
        $Update\ \mathcal{F}_{ij}$ with $l_- * Loss(-1, \hat{y}_{t,ij})$
      end if
    end if
  end for
end for

C. Confidence-triggered fallback mode

In the previous work [9], a class with w_t^i below a fixed disappearance threshold is considered as a disappeared class. When an existing class disappears, the OVA classifier for that class is deactivated and excluded from the classification in Eq. (1). When a disappeared class reoccurs, the deactivated OVA classifier for that class is reactivated. However, using a stationary disappearance threshold is not suitable for a dynamic data stream. In the specific scenario where a class emerges with a dominant amount, we observe that existing classes often appear with minimal prior probabilities due to the overwhelming presence of the emerging classes. These existing classes are mistakenly excluded from the classification process, resulting in unreliable predictions. In addition, selecting an optimal disappearance threshold becomes impractical because a fixed threshold may not cater all possible cases that could arise in the data stream.

To tackle this, we introduce a confidence-triggered fallback mode to enhance the reliability of predictions, as shown in Algorithm 2. In our method, we define the unreliable classification as one where all prediction scores are below a confidence threshold, which means that all activated ensemble OVA models predict that the instance $\boldsymbol{x}_t$ does not belong to any of their corresponding classes. We set the value of confidence threshold as 0 in this case. When an unreliable classification occurs, all inactivated ensemble OVAs are temporally activated and incorporated to classify $\boldsymbol{x}_t$.

Algorithm 2 Classification

Input: Instance $\boldsymbol{x}_t$; EEOF Model $\mathcal{O} = \{\mathcal{E}_i\}$, where $\mathcal{E}_i = \{\mathcal{F}_{ij}\}$.
Output: Prediction $\hat{y}_t$; Score matrix $\hat{Y}_t = [\hat{y}_{t,ij}]$.

$\hat{Y}_t = [\hat{y}_{t,ij}]$, $\hat{y}_{t,ij} \leftarrow 0$; $S = [s_i]$, $s_i \leftarrow -1$
for each $\mathcal{E}_i \in \mathcal{O}$ **do**
  for each base classifier $\mathcal{F}_{ij} \in \mathcal{E}_i$ **do**
    $\hat{y}_{t,ij} \leftarrow \mathcal{F}_{ij}(\boldsymbol{x}_t)$
  end for
  if $\mathcal{E}_i$ is activated **then**
    $s_i \leftarrow \frac{1}{m} \sum_{j=1}^{m} \hat{y}_{t,ij}$
  end if
end for
// Confidence-triggered fallback mode
$confidence_score = max(\{s_1, ..., s_n\})$
if $confidence_score <= confidence_threshold$ **then**
  for each $\mathcal{E}_i \in \mathcal{O}$ **do**
    if $\mathcal{E}_i$ is inactivated **then**
      $s_i \leftarrow \frac{1}{m} \sum_{j=1}^{m} \hat{y}_{t,ij}$
    end if
  end for
end if
$\hat{y}_t \leftarrow argmax_i(s_i)$

An overview of EEOF is displayed in Algorithm 3.

Algorithm 3 An overview of EEOF in data stream

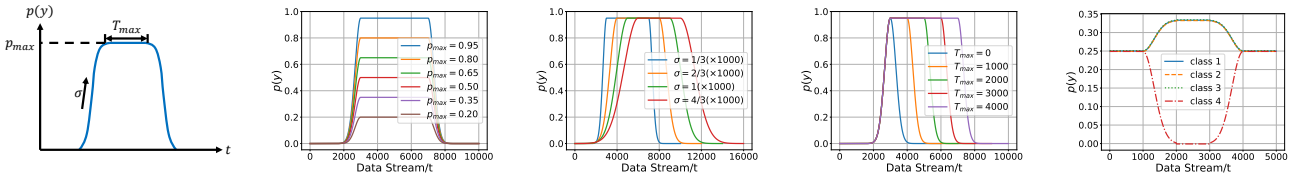
$\mathcal{O} \leftarrow \emptyset$, $\mathcal{W}_t \leftarrow \emptyset$ *// Initialize EEOF model and estimated prior probability vector*
while Instance $\boldsymbol{x}_t$ arrives **do**
  if $t == 1$ **then**
    $\hat{y} \leftarrow 0$, $\hat{Y}_t \leftarrow 0$ *// The first instance in data stream*
  else
    $\hat{y}, \hat{Y}_t \leftarrow Classification(\mathcal{O}, \boldsymbol{x}_t)$
  end if
  Receive true label y_t
  Update $\mathcal{W}_t$ with the occurrence of class y_t
  if no $\mathcal{E}_i$ in $\mathcal{O}$ is available for y_t **then**
    Initialize $\mathcal{E}_{novel}$ with m base classifier for class y_t *// Class emergence.*
    $\mathcal{O} \leftarrow \mathcal{O} \cup \mathcal{E}_{novel}$
  else if $\mathcal{E}_i$ in $\mathcal{O}$ is available for y_t and $w_t^i == 0$ **then**
    Activate $\mathcal{E}_i$ for classification *// Class reoccurrence*
  end if
  for $w_t^i \in \mathcal{W}_t$ **do**
    if $w_t^i < disappearance_threshold$ **then**
      Inactivate $\mathcal{E}_i$ for classification *// Class disappearance*
      $w_t^i \leftarrow 0$
    end if
  end for
  $\mathcal{O} = Adaptation(\mathcal{O}, \mathcal{W}_t, \hat{Y}_t, \boldsymbol{x}_t, y_t)$
end while

V. Experimental study

In this section, We demonstrate the effectiveness of EEOF compared with the state-of-the-art method on a range of synthetic and real-world data streams. An ablation study is further conducted to examine different components in EEOF.

A. Experimental settings

1) Data stream: To thoroughly explore the previously neglected scenario of class emergence with a dominant amount, we consider data stream with $k - 1$ existing classes and 1 emerging class. We simulate the dynamics of the prior probability of the emerging class with p_{max}, σ, and T_{max}, which describe the extent of the dominance, the rate of rise to the dominance, and the duration of the dominance respectively as depicted in Fig. 1(a). We use the curve of Gaussian distribution to model the change of $p(y)$ with time



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

Fig. 1. (a): The curve of the prior probability of the emerging class with a dominant amount described by p_{max}, T_{max}, and σ. (b)(c)(d): Different scenarios with varying values for these parameters. (e): The prior probability curves of classes in the scenario of class disappearance and class reoccurance.

t. The remaining classes are considered as existing classes with the same positive prior probability. We choose $k = 5$ in this case. The default values for these parameters are set as: $p_{max} = 0.95$, $\sigma = 1000 * \frac{1}{3}$, $T_{max} = 4000$. We vary p_{max} with values ranging in $[0.20, 0.35, 0.5, 0.65, 0.80, 0.95]$, σ with values $1000 * [\frac{1}{3}, \frac{2}{3}, 1, \frac{4}{3}]$, and T_{max} with values $[0, 1000, 2000, 3000, 4000]$, as shown in Fig. 1(b)(c)(d). We also simulate the scenario of class disappearance and re-occurrence with 4 existing classes, and 1 disappeared and then reoccurred class. The prior probability curves of these classes are illustrated in Fig. 1(e). Four datasets, namely Letter Recognition[1], Statlog (Landsat Satellite)[2], Covertype[3], and MNIST[4] are used for generating synthetic data streams.

We also evaluate EEOF on 6 real-world data streams that exhibit the problem of class evolution, including 4 preprocessed data streams of TwitterCrawl [18], namely Tweet Stream-20 classes/A/B/C in [9], Poker-hand [19], and KDDCUP99 [20]. The key characteristics 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
KDDCUP99	494021	117	23	Class Emergence, Disappearance and Reoccurrence
Poker-hand	829201	25	10	Class Emergence, Disappearance and Reoccurrence

2) Parameter settings: To the best of our knowledge, only two existing approaches are designed for online learning with class evolution. We compare EEOF to the state-of-the-art method CBCE [9]. To ensure a fair comparison, we use the same base model, truncated Kernelized Logistic Regression (KLR) [21], and the same values for hyperparameters. In KLR, we select the Gaussian Kernel as the kernel function, and the kernel width σ_{kernel} is set to the 5-th percentile of the pairwise distances between all pairs of instances [22]. The regularization parameter λ and the learning rate η in KLR are tuned on an initial fraction of data streams. We conduct a variant of 5-folder cross-validation, where four out of every five instances are used for training, and the remaining one instance is used for validation. The tuned values are listed in Table II. Other hyperparameters are set to commonly

[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
TUNED HYPERPARAMETERS OF KLR FOR DATA STREAMS

Data Stream	η	λ	σ_{kernel}
Letter (Synthetic)	0.01	1	1
Statlog (Synthetic)	0.01	1	14
Covertype (Synthetic)	2	0.0001	5.4
MNIST (Synthetic)	6	0.0001	7
TwitterCrawl (Real-world)	0.3	0.0005	0.13
KDDCUP99 (Real-world)	0.0001	0.0001	0.3
Poker-hand (Real-world)	8	0.001	3.4

recommended or empirically determined values: maximum stored kernel vectors in KLR is set to 5000, the time decay factor $\beta = 0.9$, the disappearance threshold is set to 10^{-5}, and ensemble size in each ensemble OVA model is 10.

3) Evaluation metrics: In the data stream with class evolution, the distribution of instances across different classes is dynamically imbalanced. To assess the overall performance of multi-classfication, we compute the average of the G-mean values for multiple classes within all sliding windows of a fixed size 200. While the average of the time-decayed G-mean is commonly used in the field of online learning [2], this metric does not account for changes in the set of classes. The time-decayed recalls for disappeared classes remain constant, and are still included after these classes have disappeared. In [9], the average of G-mean in every data chunk is calculated. However, this metric is primarily used to compare their method with chunk-based algorithms. In our experimental evaluation, we compare algorithms in an online learning manner. The chosen metric is required to track the multi-classification performance at each time step and accumulate them to obtain the overall performance. We conduct 10 independent runs for each algorithm on each data stream, and the significance of the difference between algorithms is examined using the Wilcoxon rank sum test [23]. Our souce code is available at https://github.com/lflfdxfn/EEOF

B. Experimental analysis

1) Performance on synthetic data streams: We first investigate methods on the synthetic data streams with a dominant class emergence. Our method EEOF surpasses CBCE in all choices of p_{max}, as shown in Fig. 2(a)(d)(g)(j). The performance of CBCE does not change obviously when p_{max} is smaller than 0.8, but drops rapidly when $p_{max} = 0.95$. In this case, CBCE focuses more on the dominant novel class while weakening the classification performance on exisiting classes. Moreover, when the novel class emerges with a prior probability of 0.95, existing classes are sampled with

a prior probability of 1.25%, and may be mistakenly regarded as disappeared classes and excluded from the classification. EEOF outperforms CBCE across all choices of σ, and the performance remains stable in cases of letter, statlog, and MNIST, as shown in Fig. 2(b)(e)(h)(k). The performance of CBCE drops as σ decreases. EEOF maintains an unaffected performance in the MNIST case and higher performance in all cases of T_{max}, as shown in Fig. 2(c)(f)(i)(l), The performance of CBCE drops as the emerging class remains dominant with larger duration. Such a thoroughly experimental study on the neglected scenarios demonstrates the superiority of our proposed algorithm.

We also investigate algorithms on synthetic data streams with class disappearance and class reoccurrence, as shown in Table III. It is evident that EEOF performs significantly better than CBCE in all cases, further proving the superiority of EEOF on handling diverse class evolution scenarios.

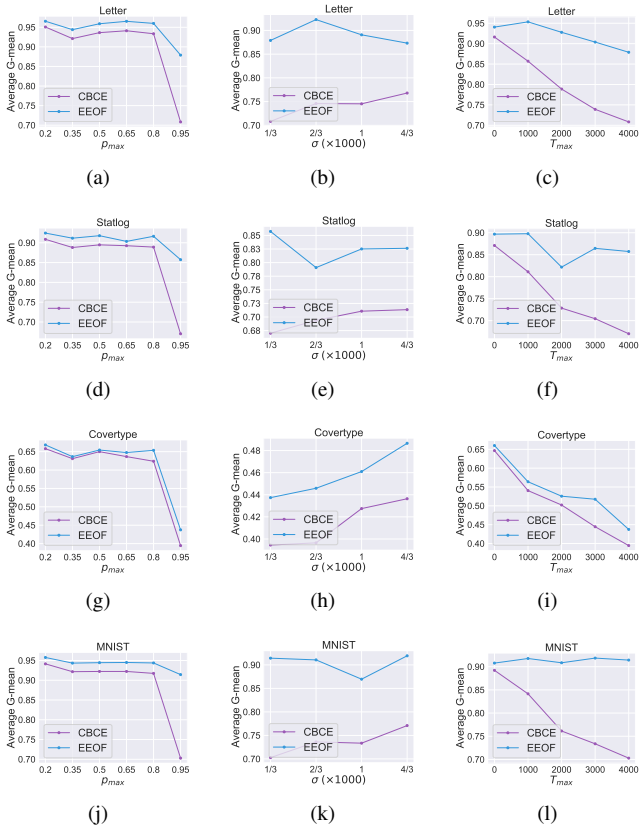


Fig. 2. The average G-mean values of CBCE and EEOF in diverse synthetic data streams for class evolution with a dominant emerging class.

2) Performance on real-world data streams: As shown in Table IV, EEOF achieves significantly higher G-mean results, except for Tweet Stream-B. As shown in Table I, Tweet Stream-B exhibits class disappearance and reoccurrence, which falls outside the focused scenarios of our method EEOF, and the underlying assumption in CBCE still holds. However, the performance difference between EEOF and CBCE in Tweet Stream-B is negligible compared to the improvements of EEOF in the other real-world data streams,

TABLE III
THE AVERAGE OF G-MEAN (MEAN/STD) IN 10 RUNS ON SYNTHETIC DATA STREAMS WITH CLASS DISAPPEARANCE AND CLASS REOCCURRENCE.

Algorithm	Letter	Statlog	Covertype	MNIST
EEOF	**0.9572/0.0006**†	**0.8864/0.0004**†	**0.7257/0.0024**†	**0.9547/0.0137**†
CBCE	0.9374/0.0004	0.8578/0.0015	0.7008/0.0021	0.9022/0.0035

The best mean result is highlighted in boldface. If this result is significantly better than the result obtained from the other algorithm (Wilcoxon rank sum test at 95% confidence level), it is marked with a † symbol.

TABLE IV
THE AVERAGE OF G-MEAN (MEAN/STD) IN 10 RUNS ON REAL-WORLD DATA STREAMS.

Algorithm	Tweet Stream-A	Tweet Stream-B	Tweet Stream-C	Tweet Stream-20 classes	KDDCUP99	Poker-hand
EEOF	**0.6444/0.0064**†	0.7245/0.0014	**0.6205/0.0025**†	**0.0586/0.0009**†	**0.8410/0.0005**†	**0.3956/0.0089**†
CBCE	0.6226/0.0017	**0.7284/0.0020**†	0.5957/0.0020	0.0571/0.0014	0.2593/0.0002	0.3839/0.0026

The best mean result is highlighted in boldface. If this result is significantly better than the result obtained from the other algorithm (Wilcoxon rank sum test at 95% confidence level), it is marked with a † symbol.

particularly in KDDCUP99. These results demonstrate the capability of EEOF in handling diverse class evolution scenarios.

3) Ablation study: We design three variants to investigate the importance of each novel component within EEOF in handling the neglected scenarios and real-world data streams: i) In EOF, a single OVA model replaces the ensemble OVA model, which can be regarded as EEOF with $m = 1$. ii) In EEOF-Adapt, the model adaptation method in EEOF is replaced by the method in CBCE. iii) In EEOF-Pred algorithm, the confidence-triggered fallback mode is removed. For synthetic data streams, we set the same default values for p_{max}, σ, and T_{max}, and vary them by choosing the the minimum and maximum values of the selected ranges. As shown in Table V and Table VI, by comparing EEOF with EOF, it is observed that the ensemble of ensemble OVA structure improves the performance in all cases. The replacement of the designed model adaptation method causes the EOF algorithm to perform weaker in almost all class emergence scenarios, except for two MNIST cases with $T_{max} = 0$ and $p_{max} = 0.20$. This is due to the short duration and the non-dominant probability of the emerging class respectively. It can be concluded that the confidence-triggered fallback mode helps to achieve equivalant or better performance by comparing EEOF with EEOF-Pred in both tables, except for Tweet Stream-B, where the degradation

TABLE V
THE AVERAGE OF G-MEAN (MEAN/STD) IN 10 RUNS OF THE ABLATION STUDY ON SYNTHETIC DATA STREAMS WITH CLASS EMERGENCE.

Data Stream	p_{max}	T_{max}	$\sigma(\times 1000)$	EEOF	EOF	EEOF-Adapt	EEOF-Pred
Letter	0.20	4000	1/3	**0.9654/0.0010**†	0.9562/0.0061	0.9638/0.0002	0.9524/0.0010
Letter	0.95	4000	1/3	**0.8789/0.0161**†	0.8463/0.0092	0.8448/0.0102	0.7105/0.0010
Letter	0.95	0	1/3	**0.9406/0.0015**†	0.9338/0.0034	0.9387/0.0009	0.9176/0.0010
Letter	0.95	4000	4/3	**0.8732/0.0022**†	0.8685/0.0067	0.8601/0.0006	0.7695/0.0004
Statlog	0.20	4000	1/3	**0.9245/0.0005**†	0.9159/0.0031	0.9223/0.0005	0.9120/0.0006
Statlog	0.95	4000	1/3	**0.8573/0.0023**†	0.8411/0.0164	0.8046/0.0054	0.6699/0.0004
Statlog	0.95	0	1/3	**0.8968/0.0007**†	0.8898/0.0038	0.8956/0.0005	0.8728/0.0007
Statlog	0.95	4000	4/3	**0.8263/0.0020**†	0.8075/0.0126	0.7951/0.0010	0.7188/0.0016
Covertype	0.20	4000	1/3	**0**.6683/0.0050	0.6510/0.0100	**0.6716/0.0030**	0.6683/0.0050
Covertype	0.95	4000	1/3	**0.4374/0.0139**†	0.4239/0.0153	0.4108/0.0071	0.4221/0.0081
Covertype	0.95	0	1/3	**0.6600/0.0059**†	0.6387/0.0135	0.6591/0.0038	**0.6600/0.0059**
Covertype	0.95	4000	4/3	**0.4866/0.0102**†	0.4699/0.0164	0.4374/0.0073	0.4683/0.0108
MNIST	0.20	4000	1/3	0.9577/0.0009	0.9443/0.0028	**0.9587/0.0006**†	0.9452/0.0009
MNIST	0.95	4000	1/3	**0.9143/0.0109**†	0.8874/0.0180	0.8941/0.0117	0.7046/0.0011
MNIST	0.95	0	1/3	0.9078/0.0052	0.8922/0.0080	**0.9153/0.0023**†	0.8911/0.0055
MNIST	0.95	4000	4/3	**0.9193/0.0082**†	0.8904/0.0162	0.9042/0.0066	0.7725/0.0065

The best mean result is highlighted in boldface. If EEOF significantly outperforms at least two variants, or a variant performs significantly better than EEOF (Wilcoxon rank sum test at 95% confidence level), it is marked with a † symbol.

TABLE VI
THE AVERAGE OF G-MEAN (MEAN/STD) IN 10 RUNS OF THE ABLATION STUDY ON REAL-WORLD DATA STREAMS.

Data Stream	EEOF	EOF	EEOF-Adapt	EEOF-Pred
Tweet Stream-A	**0.6444/0.0064**†	0.6239/0.0071	0.6475/0.0022	0.6200/0.0042
Tweet Stream-B	0.7245/0.0014	0.7005/0.0031	0.7258/0.0007	**0.7288/0.0016**†
Tweet Stream-C	**0.6205/0.0025**†	0.6035/0.0038	0.6151/0.0024	0.5980/0.0035
Tweet Stream-20 classes	**0.0586/0.0009**†	0.0353/0.0024	0.0581/0.0005	0.0568/0.0008
KDDCUP99	**0.8410/0.0005**†	0.8207/0.0045	0.8396/0.0006	0.2592/0.0003
Poker-hand	0.3956/0.0089	0.3518/0.0076	0.3420/0.0016	**0.4021/0.0087**

The best mean result is highlighted in boldface. If EEOF significantly outperforms at least two variants, or a variant performs significantly better than EEOF (Wilcoxon rank sum test at 95% confidence level), it is marked with a † symbol.

is still considerably slight. Hence, the default framework of EEOF is generally recommended to address scenarios involving the class emergence with a dominant amount, and diverse real-world data streams with class evolution.

VI. CONCLUSION

In this work, we propose a novel framework EEOF to address the class evolution problem and handle scenarios involving dominant emerging classes, which are not considered in the previous work but prevalent in real-world situations. Our proposed algorithm operates in an online learning manner with no need for storing past data. This framework is constructed in an ensemble of ensemble manner to enhance diversity within each OVA ensemble model and incorporates a novel adaptive model adaptation method that does not rely on any underlying assumptions about the data stream. Additionally, a confidence-triggered fallback mode is integrated to prevent performance degradation resulting from incorrect decisions regarding class disappearance. Comparative evaluations against the state-of-the-art method demonstrate that our algorithm achieves significantly better performance on both synthetic and real-world data streams, especially in scenarios featuring dominant emerging classes.

ACKNOWLEDGMENT

This work is partially funded by (UTS part), and National Natural Science Foundation of China (NSFC) under Grant 62250710682.

REFERENCES

[1] H. Xu, B. Liu, L. Shu, and P. Yu, "Open-world learning and application to product classification," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 3413–3419.

[2] S. Wang, L. L. Minku, and X. Yao, "A systematic study of online class imbalance learning with concept drift," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4802–4821, 2018.

[3] P. ZareMoodi, S. Kamali Siahroudi, and H. Beigy, "Concept-evolution detection in non-stationary data streams: a fuzzy clustering approach," *Knowledge and Information Systems*, vol. 60, no. 3, pp. 1329–1352, 2019.

[4] S. U. Din and J. Shao, "Exploiting evolving micro-clusters for data stream classification with emerging class detection," *Information Sciences*, vol. 507, pp. 404–420, 2020.

[5] E. R. de Faria, A. C. Ponce de Leon Ferreira Carvalho, and J. Gama, "Minas: multiclass learning algorithm for novelty detection in data streams," *Data Mining and Knowledge Discovery*, vol. 30, no. 3, pp. 640–680, 2016.

[6] A. Haque, L. Khan, and M. Baron, "Sand: Semi-supervised adaptive novel class detection and classification over data stream," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Feb. 2016.

[7] D.-W. Zhou, Y. Yang, and D.-C. Zhan, "Learning to classify with incremental new class," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 6, pp. 2429–2443, 2022.

[8] S. Hashemi, Y. Yang, Z. Mirzamomen, and M. Kangavari, "Adapted one-versus-all decision trees for data stream classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 5, pp. 624–637, 2009.

[9] Y. Sun, K. Tang, L. L. Minku, S. Wang, and X. Yao, "Online ensemble learning of data streams with gradually evolved classes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1532–1545, 2016.

[10] Z. Wang, Z. Kong, S. Changra, H. Tao, and L. Khan, "Robust high dimensional stream classification with novel class detection," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 2019, pp. 1418–1429.

[11] X. Li, Y. Zhou, Z. Jin, P. Yu, and S. Zhou, "A classification and novel class detection algorithm for concept drift data stream based on the cohesiveness and separation index of mahalanobis distance," *Journal of Electrical and Computer Engineering*, vol. 2020, p. 4027423, 2020.

[12] X. Zheng, P. Li, X. Hu, and K. Yu, "Semi-supervised classification on data streams with recurring concept drift and concept evolution," *Knowledge-Based Systems*, vol. 215, p. 106749, 2021.

[13] X. Mu, K. M. Ting, and Z.-H. Zhou, "Classification under streaming emerging new classes: A solution using completely-random trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1605–1618, 2017.

[14] T. Al-Khateeb, M. M. Masud, L. Khan, C. Aggarwal, J. Han, and B. Thuraisingham, "Stream classification with recurring and novel class detection using class-based ensemble," in *2012 IEEE 12th International Conference on Data Mining*, 2012, pp. 31–40.

[15] M. M. Masud, T. M. Al-Khateeb, L. Khan, C. Aggarwal, J. Gao, J. Han, and B. Thuraisingham, "Detecting recurring and novel classes in concept-drifting data streams," in *2011 IEEE 11th International Conference on Data Mining*, 2011, pp. 1176–1181.

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

[17] N. C. Oza and S. J. Russell, "Online bagging and boosting," in *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, T. S. Richardson and T. S. Jaakkola, Eds., vol. R3. PMLR, 04–07 Jan 2001, pp. 229–236.

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

[19] R. Cattral, F. Oppacher, and D. Deugo, "Evolutionary data mining with automatic rule generalization," *Recent Advances in Computers, Computing and Communications*, vol. 1, no. 1, pp. 296–300, 2002.

[20] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6.

[21] J. Kivinen, A. Smola, and R. Williamson, "Online learning with kernels," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2165–2176, 2004.

[22] P. K. Mallapragada, R. Jin, and A. Jain, "Non-parametric mixture models for clustering," in *Structural, Syntactic, and Statistical Pattern Recognition*, E. R. Hancock, R. C. Wilson, T. Windeatt, I. Ulusoy, and F. Escolano, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 334–343.

[23] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 977–979.