$\ell_{1,2}$-Norm and CUR Decomposition based Sparse Online Active Learning for Data Streams with Streaming Features

Zhong Chen[1], Yi He[2], Di Wu[3], Liudong Zuo[4], Keren Li[5], Wenbin Zhang[6], and Zhiqiang Deng[7]

[1]School of Computing, Southern Illinois University
[2]School of Data Science, William & Mary
[3]College of Computer and Information Science, Southwest University
[4]Computer Science Department, California State University Dominguez Hills
[5]Department of Mathematics, University of Alabama at Birmingham
[6]School of Computing and Information Sciences, Florida International University
[7]Department of Civil and Environmental Engineering, Louisiana State University

2024 IEEE International Conference on Big Data (BigData) | 979-8-3503-6248-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/BigData62323.2024.10825278

***Abstract*—Aiming at learning from a sequence of data instances over time, online learning has attracted increasing attention in the big data era. As two important variants, sparse online learning has been extensively explored by facilitating sparse constraints for online models such as truncated gradient, ℓ_1-norm regularization, ℓ_1-ball projection, and regularized dual averaging; while online active learning aims to build an online prediction model with a limited number of labeled instances, deploying the so called query strategies to select informative instances over time. However, most existing studies consider sparse online learning or online active learning with fixed feature spaces, whereby in real practice the features may be dynamically evolved over time. To the end, we propose a novel unified one-pass online learning framework named OASF for simultaneously online active learning and sparse online learning tailored for data streams described by open feature spaces, where new features can emerge constantly, and old features may be vanished over various time spans. Specifically, we technically develop an effective online CUR matrix decomposition based on the $\ell_{1,2}$ mixed norm constraint for simultaneously selecting important up-to-date samples in a sliding window and facilitating stable and meaningful features in open feature spaces over time. If the loss function is simultaneously Lipschitz and convex, a sub-linear regret bound of our proposed algorithm is guaranteed with. Extensive experiments that are conducted with multiple streaming datasets have demonstrated the effectiveness of the proposed OASF compared with state-of-the-art online active learning and sparse online learning methods.**

***Index Terms*—online learning, active learning, CUR matrix decomposition, streaming sparse learning, $\ell_{1,2}$ mixed-norm**

I. Introduction

Advanced information technologies have enhanced the ability to collect, store, integrate, and analyze a large amount of data, introducing the emergence of new challenges for data mining and machine learning techniques [2]. In the big data era, data streams are ubiquitous, for example, the amount of data captured by sensors, smart phones, and other digital technologies has skyrocketed [28], [29]. Such streaming data coming from diverse domains such as financial and web applications are characterized by high velocity, large volume, infinite length and concept drift, providing a real-time description of our communities, cities, and natural and societal environments that constantly evolve. Traditional algorithms achieved remarkable performance on static datasets based on the rigorous assumption that the training and test sets come from the same distribution and their statistical properties will be unchanged over time [53]. Nevertheless, in streaming scenarios, these static characteristics no longer hold but are more likely evolving in an unpredictable way. To analyze the streaming data with varying patterns, many Online Learning (OL) algorithms [1] have been investigated to enable real time decision-making process, thereby making learning efficient, scalable and adaptable when processing incoming instances on-the-fly. In summary, the online algorithms [19], [26], [39], [45], [49], [51], [53], [58], [64] are more efficient and comfortable, comparing with the batch offline learning algorithms, to retrain any existing model with new receiving data instances.

However, most existing studies [1] consider online learning with fixed feature spaces, whereby in real practice the features may be dynamically evolved over time. Though some OL algorithms [7] can handle an incremental sample space, where the instances of training data emerge one after the other and are processed in a single pass, all data instances are posited to reside in a *fixed* feature space. Thus, this assumption may not hold in practice, leading to the traditional OL methods failing to deal with streaming data with dynamically evolved features. To wit, consider an urban disaster monitoring system aided by OL, where streaming data are sent from crowd-sensing devices such as smart phones and sensor kits/sites that scatter across a geographically wide region in real time. Fixing the set of features to be used in a prior is next to impossible for two reasons. First, new users join the sensing effort would commit data collected by their own devices (e.g., a new-brand cellphone), thus introducing new sensory features. Second, as users may stop sending data for reasons like battery exhaustion

or network malfunction, any pre-existing features can become unobserved in later time snapshots. In this study, we coin such data inputs as *streaming data in open feature spaces* (SDOFS).

To enable learning in an SDOFS setting, we propose a novel OL approach that encourages a sparse model solution and promotes an active learning strategy, termed sparse Online Active Learning for data streams with Streaming Features (OASF, for short). Our key idea is two-fold. First, to tame the feature space dynamics, we tailor an online passive-aggressive (PA) program based on the margin-maximum principle. The proposed PA program reweighs the learning weights at each iteration only if the increment or decrement of feature space would incur prediction loss. Second, to encourage model sparsity, we impose an $\ell_{1,2}$-norm constraint on the PA program and solve it with a closed-from solution. We leverage an incremental matrix with memory of the learned feature weights and apply the proximal operator on it to stabilize the resultant sparse solution. In the matrix, the learning weights of a feature should be set to zero consistently over the memory length, if the feature is irrelevant. We show that our OASF enjoys a closed form solution, which lends itself amenable for implementation. Theoretical and empirical studies are carried out to substantiate the effectiveness of our proposed method.

Specific contributions of this paper are as follows:

1) We explore the sparse active online learning problem in open feature spaces, with key challenge imposed by the emergence of new features and the disappearance of old features over time.
2) A new algorithm termed OASF is proposed to yield sparse model solution in the wildly evolving data environment. Our solution enjoys a closed form thus computational efficiency. Details are given in Section III.
3) Theoretical analysis substantiates the sub-linear regret bound of OASF. Main results are in Section IV.
4) Extensive experiments demonstrate the superiority of our proposal over four the state-of-the-art OL competitors in error rates, documented in Section V.

The rest of the paper is organized as follows. The related work is discussed in Section II. Our proposed method and the theoretical regret bound analysis are elaborated in Sections III and IV, respectively. The experimental results are reported in Section V. The conclusion and future work are summarized in Section VI.

II. Related Work

We relate our proposed OASF approach to two research threads: *online learning in open feature spaces*, which performs OL in the same data environment as we do, and *sparse active online learning*, which aims to reduce the OL model dimension and study query strategy but mostly in fixed feature spaces, while ours are open. Note, we are aware of another thread of studies termed *online streaming feature selection* (OSFS) [43], [44], [46], [47], which are seemingly similar to our study but essentially different in two aspects. First, OSFS allows incremental feature input but posits a *fixed instance space*, i.e., all instances are given in advance before feature selection starts. In our setting, however, the data instances are presented one after the other like normal streams. Second, OSFS decouples feature selection and learning, i.e., it selects a set of highly relevant features at first and then trains and evaluates a predictive model on it in an offline fashion. Our setting is more challenging as we need to perform active learning and feature selection jointly in an online fashion. Due to the clear disparities, we do not discuss nor compare with any OSFS studies in this paper.

A. Online Learning in Open Feature Spaces

Combining online learning and streaming feature selection, Zhang et al. [7] propose an online learning with streaming features algorithm (OLSF) and its two variants to enable learning from trapezoidal data streams with infinite training instances and increasing features. Learning with incremental and decremental features is crucial but rarely studied, particularly when the data comes like a stream and thus it is infeasible to keep the whole data for optimization [8], [9], [11], [12], [40]. To address the issue, Hou and Zhou [40] study this challenging problem and present the OPID approach. OPID attempts to compress important information of vanished features into functions of survived features, and then expand to include the augmented features. To handle capricious data streams with an arbitrarily varying feature space, He et al. [8] develop an online learning with capricious data streams algorithm (OCDS) by training a learner based on a universal feature space that includes the features appeared at each iteration. Hou et al. [9] propose a Feature Evolvable Streaming Learning (FESL) paradigm, where old features would vanish and new features would occur. Rather than relying on only the current features, FESL attempts to recover the vanished features and exploit it to improve performance. To explore online learning from data streams with an varying feature space, He et al. [11] propose the OVFM method to model the complex joint distribution underlying mixed data with Gaussian copula, where the observed features with arbitrary marginals are mapped onto a latent normal space.

However, few online learners [48] thus have been tailored for the feature correlations among old feature, survival features, and new features when introducing sparsity in the online models. The main challenge lies in that there is no good mechanisms to fairly consider the interactions among these time-evolving features when they are described by different feature spaces, which is the gap that our OASF attempt to explore and fulfill. To that end, we leverage the proximal operator of the $\ell_{1,2}$ mixed norm and show that it can be computed in a closed form by applying the well-known soft-thresholding operator to each column of the matrix, addressing the challenges in feature-evolving data streams. OASF employs $\ell_{1,2}$-norm as the distance metric for the suffered loss, and solves the optimal solution by a non-greedy algorithm, which has a closed-form solution in each iteration. This mechanism retains OASF desirable properties for handling feature-evolving data streams and is robust to outliers as well.

B. Sparse Online Learning

The goal of sparse online learning is to induce sparsity in the weights of online learning algorithms [17], ensuring the prediction model only contains a limited size of active features. The existing solutions for sparse online learning can be categorized into two main groups: truncation gradient based methods and regularized dual averaging based methods. The former group follows the general idea of subgradient descent with truncation. For example, Langford et al. [18] propose a simple yet efficient modification of the standard stochastic gradient via truncated gradient (TG) to achieve sparsity in online learning. Duchi and Singer [20] further propose a forward-backward splitting (FOBOS) algorithm to solve the sparse online learning problems. However, with high-dimensional streaming data, the TG and FOBOS methods suffer from slow convergence and high variance due to heterogeneity in feature sparsity. To the end, Ma and Zhang [21] introduce a stabilized truncated stochastic gradient descent (STSGD) algorithm. Chen et al. [17] extend TG to cost-sensitive online learning via truncated gradient (CSTG) and further propose asymmetric truncated gradient (ATG) [60] for adaptive online learning. The latter group focuses on the dual averaging methods that can explicitly exploit the regularization structure. One representative method is the regularized dual averaging (RDA) proposed in [22], which learns the variables by solving a regularized optimization problem that involves the average of all past subgradients. Lee and Wright [23] further extend RDA to RDA+ by using a more aggressive truncation threshold. Ushio and Yukawa [24] propose the projection based regularized dual averaging (PDA) method to exploit a sparsity-promoting regularizer. Zhou et al. [25] propose an online algorithm GraphDA for graph-structured sparsity constraint problems.

C. Online Active Learning

Active learning has attracted the data mining and machine learning community in the past decades. This is because it served for important purposes to increase practical applicability of machine learning techniques, such as (i) to reduce annotation and measurement costs, (ii) to reduce manual labeling effort for experts and (iii) to reduce computation time for model training. Almost all of the current techniques focus on the classical pool-based approach, which is off-line by nature as iterating over a pool of unlabeled reference samples a multiple times to choose the most promising ones for improving the performance of the classifiers. For the online and streaming cases, the challenge is that the sample selection strategy has to operate in a fast, ideally single-pass manner. Some online active learning approaches have been proposed in connection with the paradigm of evolving models during the last decade. For example, Chu et al. [56] propose an unbiased online active learning to study selective labeling in data streams. Lu et al. [57] investigate a new online active learning algorithm (PAA) by adapting the PA algorithm in online active learning settings. Hao et al. [59] propose a second-order online active learning (SOAL) by fully exploiting both the first-order and second-order information. Krawczyk et al. [61] propose three improved active learning strategies based on learner uncertainty, dynamic allocation of budget over time and search space randomization for mining drifting data streams. Shan et al. [62] propose a new online active learning ensemble framework for drifting data streams based on a hybrid labeling strategy. Krawczyk et al. [63] propose an novel active learning approach based on ensemble algorithms that is capable of using multiple base classifiers during the label query process, and further improve the instance selection by measuring the generalization capabilities of the classifiers to better adapt to concept drifts. Liu et al. [65] develop an active learning framework (CogDQS) based on a dual-query strategy and Ebbinghaus's law of human memory cognition. Zhang et al. [66] propose a novel online active learning framework based on sample representativeness.

Unfortunately, none of these methods can be generalized to open feature spaces. Specifically, for a new feature, its weight either is initialized as zero, which can be interpreted as irrelevant, or is randomly initialized, which would require a sufficiently large number of instances to converge. Likewise, for an old feature becoming unobserved, no gradient information is available on its entry, thus its weight is not updated. Both cases lead to statistical bias. Our OASF approach outperforms the prior studies by leveraging a passive-aggressive (PA) learner that 1) apportions the weights from other existing features to a new feature for its better initialization and 2) redistributes the weight of an unobserved old feature to other features. Closed-form solutions are available for both cases, which lends our OASF an advantage of fast convergence and be integrative to the tailored sparsity constraints.

III. Proposed Methods

A. Problem Statement

We start with a typical SDOFS modeling. Write an input sequence $\{(\boldsymbol{x}_t, y_t) \mid t \in [T]\}$. Each data instance $\boldsymbol{x}_t \in \mathbb{R}^{d_t}$ received at the t-th round is a vector of d_t-dimension, associated with a true class label $y_t \in \{-1, +1\}$. We hereby follow prior art [8], [10] to restrict our interest in a binary classification problem, as multi-class setups can be trivially reduced to binary cases with One-vs-One or One-vs-Rest strategies [16].

At each round, the learner observes $\boldsymbol{x}_t$ and returns a prediction $\hat{y}_t = \text{sign}(\boldsymbol{w}_t^\top \boldsymbol{x}_t)$. The true label y_t is then revealed, and the learner suffers a risk if the prediction was incorrect, e.g., gauged by hinge loss $\ell_t(\boldsymbol{w}_t, (\boldsymbol{x}_t, y_t)) = \max(0, 1 - y_t(\boldsymbol{w}_t^\top \boldsymbol{x}_t))$. The learner then is updated to $\boldsymbol{w}_{t+1}$ based on the loss information and gets ready to the next round. Our goal is to find an updating strategy $\mathcal{A}$ that minimizes empirical risk and, more importantly, yields a sparse model solution over T rounds, namely:

$$\min_{\boldsymbol{w}_t \in \mathbb{R}^{d_t}} \mathbb{E}_{t \in [T]} \left[\ell_t(\boldsymbol{w}_t, (\boldsymbol{x}_t, y_t)) \right] + \|\boldsymbol{w}_t\|_0, \tag{1}$$

where the ℓ_0-norm counts the number of nonzero entries in weight vector $\boldsymbol{w}_t$.

The main challenge is imposed by the fact that the feature space is open and can be either decremental ($d_{t+1} \leq d_t$) or incremental ($d_{t+1} \geq d_t$), due to newly emerging features or unobserved old features, respectively. The survived features are represented by $\boldsymbol{x}_{t+1}^s = \boldsymbol{x}_t \cap \boldsymbol{x}_{t+1}$ or $\boldsymbol{w}_{t+1}^s = \boldsymbol{w}_t \cap \boldsymbol{w}_{t+1}$, the vanished features are represented by $\boldsymbol{x}_{t+1}^v = \boldsymbol{x}_t \setminus \boldsymbol{x}_{t+1}$ or $\boldsymbol{w}_{t+1}^v = \boldsymbol{w}_t \setminus \boldsymbol{w}_{t+1}$, and the new features are represented by $\boldsymbol{x}_{t+1}^n = \boldsymbol{x}_{t+1} \setminus \boldsymbol{x}_t$ or $\boldsymbol{w}_{t+1}^n = \boldsymbol{w}_{t+1} \setminus \boldsymbol{w}_t$.

B. Online Passive-Aggressive Feature Reweighing

If the feature dimension is decreased from the t-th round to the $(t+1)$-th round (i.e., $d_t \geq d_{t+1}$), then we decompose the instance $\boldsymbol{x}_t = [\boldsymbol{x}_t^s; \boldsymbol{x}_t^d]$ and the corresponding weight vector $\boldsymbol{w}_t = [\boldsymbol{w}_t^s; \boldsymbol{w}_t^d]$, where $\boldsymbol{x}_t^s \in \mathbb{R}^{d_{t+1}}$ is the vector with survival features and $\boldsymbol{x}_t^d \in \mathbb{R}^{d_t - d_{t+1}}$ is the vector with vanished features. That is, $\boldsymbol{x}_t^s$ and $\boldsymbol{w}_t^s$ have the same dimension as $\boldsymbol{x}_{t+1}$ and $\boldsymbol{w}_{t+1}$. Moreover, to make the model be robust to the noise, we use the soft-margin technique by introducing a slack variable ξ into the optimization problem. In this case, we extend the passive-aggressive (PA) algorithm to update $\boldsymbol{w}_{t+1}$ by solving the following optimization task:

$$\boldsymbol{w}_{t+1} = \underset{\boldsymbol{w} \in \mathbb{R}^{d_{t+1}}, \ell_{t+1} \leq \xi, \xi \in \mathbb{R}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}_t^s\|_2^2 + \mu\xi^2 \qquad (2)$$

where $\mu > 0$ is a penalty parameter that can tradeoff the rigidness and slackness of the online model. A larger value of μ implies a more rigid update step, and $\ell_{t+1} = \ell_{t+1}(\boldsymbol{w}, (\boldsymbol{x}_{t+1}, y_{t+1})) = \max(0, 1 - y_{t+1}(\boldsymbol{w}^T \boldsymbol{x}_{t+1}))$ is the loss at round $t+1$. Then, we derive the closed-form solution for the above equation in Theorem 1.

Theorem 1 (**Closed-form Solution of Eq. (2)**). *The closed-form solution for minimizing Eq. (2) is* $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t^s + \gamma_t y_{t+1} \boldsymbol{x}_{t+1}$, *where* $\gamma_t = \frac{\ell_{t+1}(\boldsymbol{w}_t^s, (\boldsymbol{x}_{t+1}, y_{t+1}))}{\|\boldsymbol{x}_{t+1}\|_2^2 + \frac{1}{2\mu}}$.

If the feature dimension is increased from the t-th round to the $(t+1)$-th round (i.e., $d_t \leq d_{t+1}$), then we decompose the instance $\boldsymbol{x}_{t+1} = [\boldsymbol{x}_{t+1}^s; \boldsymbol{x}_{t+1}^n]$ and the corresponding weight vector $\boldsymbol{w}_{t+1} = [\boldsymbol{w}_{t+1}^s; \boldsymbol{w}_{t+1}^n]$, where $\boldsymbol{x}_{t+1}^s \in \mathbb{R}^{d_t}$ is the vector with survival features and $\boldsymbol{x}_{t+1}^n \in \mathbb{R}^{d_{t+1} - d_t}$ is the vector with newly-observed features. That is, $\boldsymbol{x}_{t+1}^s$ and $\boldsymbol{w}_{t+1}^s$ have the same dimension as $\boldsymbol{x}_t$ and $\boldsymbol{w}_t$. In this case, similarly, we extend the PA algorithm to update $\boldsymbol{w}_{t+1}$ by solving the following optimization task:

$$\boldsymbol{w}_{t+1} = \underset{\boldsymbol{w} = [\boldsymbol{w}^s; \boldsymbol{w}^n] \in \mathbb{R}^{d_{t+1}}, \ell_{t+1} \leq \xi, \xi \in \mathbb{R}}{\operatorname{argmin}} \tfrac{1}{2}\|\boldsymbol{w}^s - \boldsymbol{w}_t\|_2^2 + \tfrac{1}{2}\|\boldsymbol{w}^n\|_2^2 + \mu\xi^2 \qquad (3)$$

where $\mu > 0$ is a penalty parameter, and $\ell_{t+1} = \ell_{t+1}(\boldsymbol{w}, (\boldsymbol{x}_{t+1}, y_{t+1})) = \max(0, 1 - y_{t+1}(\boldsymbol{w}^T \boldsymbol{x}_{t+1})) = \max(0, 1 - y_{t+1}((\boldsymbol{w}^s)^T \boldsymbol{x}_{t+1}^s) - y_{t+1}((\boldsymbol{w}^n)^T \boldsymbol{x}_{t+1}^n))$ is the loss at round $t+1$. Then, we derive the closed-form solution for the above equation in Theorem 2.

Theorem 2 (**Closed-form Solution of Eq. (3)**). *The general update strategy is the closed-form solution of Eq. (3),* $\boldsymbol{w}_{t+1} = [\boldsymbol{w}_{t+1}^s; \boldsymbol{w}_{t+1}^n] = [\boldsymbol{w}_t + \gamma_t y_{t+1} \boldsymbol{x}_{t+1}^s; \gamma_t y_{t+1} \boldsymbol{x}_{t+1}^n]$, *where* $\gamma_t = \frac{\max(0, 1 - y_{t+1} \boldsymbol{w}_t^T \boldsymbol{x}_{t+1}^s)}{\|\boldsymbol{x}_{t+1}^s\|_2^2 + \|\boldsymbol{x}_{t+1}^n\|_2^2 + \frac{1}{2\mu}} = \frac{\ell_{t+1}(\boldsymbol{w}_t, (\boldsymbol{x}_{t+1}^s, y_{t+1}))}{\|\boldsymbol{x}_{t+1}^s\|_2^2 + \|\boldsymbol{x}_{t+1}^n\|_2^2 + \frac{1}{2\mu}} = \frac{\ell_{t+1}([\boldsymbol{w}_t; \boldsymbol{0}], (\boldsymbol{x}_{t+1}, y_{t+1}))}{\|\boldsymbol{x}_{t+1}\|_2^2 + \frac{1}{2\mu}}$, *and* $[\boldsymbol{w}_t; \boldsymbol{0}] \in \mathbb{R}^{d_{t+1}}$.

Hence, using the above two strategies with closed-form solutions, we can alternately update the online model in an SDOFS setup with widely evolving vanish, survival, and new features.

C. Memory-aware $\ell_{1,2}$-Norm Model Sparsifying

In this section, we consider the problem setting of the online binary classification task for SDOFS and present the OASF method to achieve sparse solution by leveraging the ℓ_1- and ℓ_2-mixed regularizer. We observe that when using the ℓ_2 norm as the regularization function, we obtain an all zeros vector if $\|\boldsymbol{w}\|_2 \leq \lambda$ (Theorem 3). The zero vector does not carry any generalization properties, which surfaces a concern regarding the usability of the these norms as a form of regularization. This seemingly problematic phenomenon can, however, be useful in the incremental online setting. In many applications, the set of weights can be grouped into subsets where each subset of weights should be dealt with uniformly. For example, in the sparse online learning problem for SDOFS, each sliding window is associated with a different weight vector $\boldsymbol{w}^l \in \mathbb{R}^{d_l}$ $(l = 1, 2, \cdots, L)$. The prediction for a new instance $\boldsymbol{x}$ is a vector $\langle \boldsymbol{w}^1, \boldsymbol{x} \rangle, \langle \boldsymbol{w}^2, \boldsymbol{x} \rangle, \cdots, \langle \boldsymbol{w}^L, \boldsymbol{x} \rangle$, where L is the length of a specific sliding window. The predicted class is the index of the inner-product attaining the largest of the L values, $\operatorname{argmax}_{l \in \{1,2,\cdots,L\}} \langle \boldsymbol{w}^l, \boldsymbol{x} \rangle$. Since all the weight vectors operate over the same instance space, in order to achieve a sparse solution, it may be beneficial to tie the weights corresponding to the input features. That is, we would like to employ a regularization function that tends to zero the row of weights $w_1^l, w_2^l, \cdots, w_{d_l}^l$ $(l = 1, 2, \cdots, L)$ simultaneously. In these circumstances, the nullification of the entire weight vector by the ℓ_2 regularization becomes a powerful tool.

Formally, let $\boldsymbol{W} \in \mathbb{R}^{d \times L}$ represent a $d \times L$ matrix where the l-th $(l = 1, 2, \cdots, L)$ column of the matrix is the weight vector $\boldsymbol{w}^l$, where d is the total number of all evolvable features. Thus, the i-th $(i = 1, 2, \cdots, d)$ row corresponds to the weight of the i-th feature with respect to all instances. The mixed $\ell_{1,2}$-norm of $\boldsymbol{W}$, denoted $\|\boldsymbol{W}\|_{\ell_{1,2}}$, is obtained by computing the ℓ_2-norm of each row of $\boldsymbol{W}$ and then applying the ℓ_1-norm to the resulting d dimensional vector, i.e., $\|\boldsymbol{W}\|_{\ell_{1,2}} = \sum_{i=1}^{d} \|\boldsymbol{w}_i\|_2$. Thus, in a mixed-norm regularized optimization problem, we seek the minimizer of the objective function,

$$f(\boldsymbol{W}) + \lambda \|\boldsymbol{W}\|_{\ell_{1,2}} \qquad (4)$$

where $f(\boldsymbol{W})$ is a loss function, we define specifically $f(\boldsymbol{W}) = \frac{1}{2}\|\boldsymbol{W} - \boldsymbol{W}_t\|_F^2$ in our study.

Given the specific variants of various norms, the model update for the $\ell_{1,2}$ mixed-norm is readily available. Let $\boldsymbol{w}^l \in \mathbb{R}^d$ denote the l-th $(l = 1, 2, \cdots, L)$ column of the matrix $\boldsymbol{W} \in \mathbb{R}^{d \times L}$, i.e., $\boldsymbol{W} = [\boldsymbol{w}^1, \boldsymbol{w}^2, \cdots, \boldsymbol{w}^L]$, and $\bar{\boldsymbol{w}}^i \in \mathbb{R}^L$ denote the i-th $(i = 1, 2, \cdots, d)$ row of the

matrix $\boldsymbol{W} \in \mathbb{R}^{d\times L}$, i.e., $\boldsymbol{W} = [\bar{\boldsymbol{w}}^1; \bar{\boldsymbol{w}}^2; \cdots; \bar{\boldsymbol{w}}^d]$. Analogously to the standard norm-based regularization, we let $\boldsymbol{W}_t = [[\boldsymbol{w}_{t-L+1}; \boldsymbol{0}], [\boldsymbol{w}_{t-L+2}; \boldsymbol{0}], \cdots, [\boldsymbol{w}_t; \boldsymbol{0}]] \in \mathbb{R}^{d\times L}$ be the incremental matrix with all good feature alignment, where $[\boldsymbol{w}_{t-l+1}; \boldsymbol{0}] \in \mathbb{R}^d$ and $\boldsymbol{w}_{t-l+1} \in \mathbb{R}^{d_{t-l+1}}$ $(l = 1, 2, \cdots, L)$, which can be obtained by online learning with decremental or incremental features or mixed features (Section III-B). For the $\ell_{1,2}$ mixed-norm, we need to solve the problem,

$$\min_{\boldsymbol{W}\in\mathbb{R}^{d\times L}} \{\frac{1}{2}\|\boldsymbol{W} - \boldsymbol{W}_t\|_F^2 + \lambda\|\boldsymbol{W}\|_{\ell_{1,2}}\} \tag{5}$$

where $\|\cdot\|_F^2$ is the Frobenius norm of a matrix and $\lambda > 0$ is the regularization parameter.

This problem is equivalent to

$$\min_{\boldsymbol{W}=[\bar{\boldsymbol{w}}^1;\bar{\boldsymbol{w}}^2;\cdots;\bar{\boldsymbol{w}}^d]\in\mathbb{R}^{d\times L}} \sum_{i=1}^{d}\{\frac{1}{2}\|\bar{\boldsymbol{w}}^i - \bar{\boldsymbol{w}}_t^i\|_2^2 + \lambda\|\bar{\boldsymbol{w}}^i\|_2\} \tag{6}$$

where $\bar{\boldsymbol{w}}_t^i$ is the i-th row of $\boldsymbol{W}_t$. It is immediate to see that the problem given in Eq. (5) is decomposable into d separate problems of dimension L in Eq. (6), each of which can be solved by the procedures described in the following Theorem 3. The end result of solving these types of mixed-norm problems is a sparse matrix with numerous zero rows. In this way, OASF can not only alleviate the curse of dimensionality by the incremental learning strategy, but also promote the sparsity of decremental and incremental features by considering feature correlations over time. Hence, OASF has a big potential to improve the prediction performance compared with most existing methods.

Theorem 3 (Closed-form Solution of OASF)**.** *The closed-form solution of the following ℓ_2-norm minimization: $\bar{\boldsymbol{w}}_\star^i = \operatorname{argmin}_{\bar{\boldsymbol{w}}^i\in\mathbb{R}^L}\{\frac{1}{2}\|\bar{\boldsymbol{w}}^i - \bar{\boldsymbol{w}}_t^i\|_2^2 + \lambda\|\bar{\boldsymbol{w}}^i\|_2\}$, where $i = 1, 2, \cdots, d$, is:*

$$\bar{\boldsymbol{w}}_\star^i = \begin{cases} \boldsymbol{0} & \text{if} \quad \|\bar{\boldsymbol{w}}_t^i\|_2 \le \lambda \\ (1 - \frac{\lambda}{\|\bar{\boldsymbol{w}}_t^i\|_2})\bar{\boldsymbol{w}}_t^i & \text{if} \quad \|\bar{\boldsymbol{w}}_t^i\|_2 > \lambda \end{cases} \tag{7}$$

Remark 1: It is worth noting that the ℓ_2 regularization results in a zero weight vector under the condition that $\|\bar{\boldsymbol{w}}_t^i\|_2 \le \lambda$. This condition is rather more stringent for sparsity than the condition for ℓ_1 (where a weight is sparse based only on its value, while here, sparsity happens only if the entire weight vector has ℓ_2-norm less than λ), so it is unlikely to hold in high dimensions. However, it does constitute a very important building block when using a mixed ℓ_1/ℓ_2-norm as the regularization function.

In summary, the pseudo codes of the proposed OASF method are present in Algorithm 1.

D. Online Active Learning through CUR Decomposition

The CUR decomposition [54] provides a low-rank approximation to a data matrix $\boldsymbol{W} \in \mathbb{R}^{d\times L}$. In particular, CUR decomposes the data matrix $\boldsymbol{W}$ into the form of a product of three matrices as $\boldsymbol{W} \approx \boldsymbol{CUR}$, where $\boldsymbol{C} \in \mathbb{R}^{d\times c}$, $\boldsymbol{U} \in \mathbb{R}^{c\times r}$, and $\boldsymbol{R} \in \mathbb{R}^{r\times L}$ ($c < L$ and $r < d$). Unlike other low-rank approximations such as Singular Value Decomposition (SVD), CUR extracts $\boldsymbol{C}$ and $\boldsymbol{R}$ as small numbers of the column and row vectors of $\boldsymbol{W}$, respectively. In other words, $\boldsymbol{C}$ and $\boldsymbol{R}$ are subsets of c columns and r rows of the original data matrix $\boldsymbol{W}$, respectively. This property helps practitioners to interpret the result more easily than that in the case of SVD.

Algorithm 1 The OASF Algorithm

Online input: streaming instance $\boldsymbol{x}_{t+1}$; true label y_{t+1}; regularization parameter λ, penalty parameter μ, and sliding window size L.
Online output: sparse solution, $\boldsymbol{w}_{t+1}$.

Initialization: $\boldsymbol{w}_0 = \boldsymbol{0} \in \mathbb{R}^{d_0}$;
for $t = 0, 1, \ldots, T-1$ **do**
  receive $\boldsymbol{x}_{t+1} \in \mathbb{R}^{d_{t+1}}$;
  if $(d_t \ge d_{t+1})$ **then**
    predict $\hat{y}_{t+1} = \text{sign}((\boldsymbol{w}_t^s)^T\boldsymbol{x}_{t+1})$ and receive $y_{t+1} \in \{-1, +1\}$;
    suffer loss $\ell_t(\boldsymbol{w}_t) = \ell_{t+1}(\boldsymbol{w}_t^s, (\boldsymbol{x}_{t+1}, y_{t+1}))$;
    update $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t^s + \gamma_t y_{t+1}\boldsymbol{x}_{t+1}$, where $\gamma_t = \frac{\ell_{t+1}(\boldsymbol{w}_t^s, (\boldsymbol{x}_{t+1}, y_{t+1}))}{\|\boldsymbol{x}_{t+1}\|_2^2 + \frac{1}{2\mu}}$;
    sparse update $\boldsymbol{w}_{t+1} = \operatorname{argmin}_{\bar{\boldsymbol{w}}^i\in\mathbb{R}^L}\{\frac{1}{2}\|\bar{\boldsymbol{w}}^i - \bar{\boldsymbol{w}}_{t+1}^i\|_2^2 + \lambda\|\bar{\boldsymbol{w}}^i\|_2\}(i = 1, 2, \cdots, d_{t+1})$ through Eq. (7);
  else if $(d_t \le d_{t+1})$ **then**
    predict $\hat{y}_{t+1} = \text{sign}([\boldsymbol{w}_t; \boldsymbol{0}]^T\boldsymbol{x}_{t+1})$ and receive $y_{t+1} \in \{-1, +1\}$;
    suffer loss $\ell_t(\boldsymbol{w}_t) = \ell_{t+1}([\boldsymbol{w}_t; \boldsymbol{0}], (\boldsymbol{x}_{t+1}, y_{t+1}))$;
    update $\boldsymbol{w}_{t+1} = [\boldsymbol{w}_{t+1}^s; \boldsymbol{w}_{t+1}^n] = [\boldsymbol{w}_t + \gamma_t y_{t+1}\boldsymbol{x}_{t+1}^s; \gamma_t y_{t+1}\boldsymbol{x}_{t+1}^n]$, where $\gamma_t = \frac{\ell_{t+1}([\boldsymbol{w}_t; \boldsymbol{0}], (\boldsymbol{x}_{t+1}, y_{t+1}))}{\|\boldsymbol{x}_{t+1}\|_2^2 + \frac{1}{2\mu}}$;
    sparse update $\boldsymbol{w}_{t+1} = \operatorname{argmin}_{\bar{\boldsymbol{w}}^i\in\mathbb{R}^L}\{\frac{1}{2}\|\bar{\boldsymbol{w}}^i - \bar{\boldsymbol{w}}_{t+1}^i\|_2^2 + \lambda\|\bar{\boldsymbol{w}}^i\|_2\}(i = 1, 2, \cdots, d_{t+1})$ through Eq. (7);
  end if
end for

Since the $\boldsymbol{R}$ has been determined by the $\ell_{1,2}$ constraint (r rows of $\boldsymbol{W}$ will be zero vectors in Section III-C), which imposes sparse rows of the incremental matrix $\boldsymbol{W} \in \mathbb{R}^{d\times L}$. For the selection of $\boldsymbol{C}$, the optimization problem is defined as follows

$$\min_{\boldsymbol{X}\in\mathbb{R}^{L\times L}} \frac{1}{2}\|\boldsymbol{W} - \boldsymbol{WX}\|_F^2 + \eta\sum_{i=1}^{L}\|\boldsymbol{X}_{(i)}\|_2, \tag{8}$$

where $\boldsymbol{X} \in \mathbb{R}^{L\times L}$ is the parameter matrix, and $\eta > 0$ is a regularization parameter. Given the matrix $\boldsymbol{W}$, $\boldsymbol{W}_{(i)} \in \mathbb{R}^{1\times L}$ and $\boldsymbol{W}^{(i)} \in \mathbb{R}^{d\times 1}$ denote the i-th row vector and i-th column vector of $\boldsymbol{W}$, respectively. Similarly, given a set of indices $\mathcal{I}$, $\boldsymbol{W}_{\mathcal{I}}$ and $\boldsymbol{W}^{\mathcal{I}}$ denote the submatrices of $\boldsymbol{W}$ containing only $\mathcal{I}$ rows and columns, respectively. The term $\|\boldsymbol{X}_{(i)}\|_2$ induces $\boldsymbol{X}_{(i)}$ to be a zero vector, where $\boldsymbol{X}_{(i)} \in \mathbb{R}^{1\times L}$ is the i-th row vector of $\boldsymbol{X}$. The regularization constant η controls the degree of sparsity of the parameter matrix $\boldsymbol{X}$. If $\boldsymbol{X}_{(i)} = \boldsymbol{0}$ is a zero

vector, the corresponding column of the data matrix $\boldsymbol{W}^{(i)}$ can be considered as an unimportant column for problem (8). On the other hand, $\boldsymbol{W}^{(i)}$ is important when the corresponding $\boldsymbol{X}_{(i)}$ is a nonzero vector. Therefore, we can select columns $\boldsymbol{C}$ as $\boldsymbol{W}^{\mathcal{I}}$, where $\mathcal{I} \subseteq [L] = \{1, 2, \cdots, L\}$ represents the indices corresponding to the nonzero row vectors of $\boldsymbol{X}$. Hence, the proposed OASF algorithm can select more informative instances in the sliding window incrementally.

E. Coordinate Descent

Problem (8) can be solved by using the coordinate descent [55]. The algorithm iteratively updates each parameter vector $\boldsymbol{X}_{(i)}$ corresponding to each row of the parameter matrix $\boldsymbol{X}$ until $\boldsymbol{X}$ converges. Then, the following equation is used to update $\boldsymbol{X}_{(i)} \in \mathbb{R}^{1\times L}$:

$$\boldsymbol{X}_{(i)} = \begin{cases} \boldsymbol{0} & \text{if} \quad \|\boldsymbol{u}_i\|_2 \leq \eta \\ (1 - \frac{\eta}{\|\boldsymbol{u}_i\|_2})\boldsymbol{u}_i & \text{if} \quad \|\boldsymbol{u}_i\|_2 > \eta \end{cases} \tag{9}$$

where $\boldsymbol{u}_i \in \mathbb{R}^{1\times L}$ is computed as follows:

$$\boldsymbol{u}_i = \frac{(\boldsymbol{W}^{(i)})^\top}{\|\boldsymbol{W}^{(i)}\|_2}(\boldsymbol{W} - \sum_{j=1, j\neq i}^{L} \boldsymbol{W}^{(j)}\boldsymbol{X}_{(j)}). \tag{10}$$

where $\boldsymbol{W}^{(i)} \in \mathbb{R}^{d\times 1}$ denote the i-th column vector of $\boldsymbol{W}$. Algorithm 2 shows the pseudocode of coordinate descent. The inner loop (lines 3–4) performs Equation (9) to update each row of $\boldsymbol{X}$, and the outer loop (lines 2–5) repeats the update process until $\boldsymbol{X}$ converges. The computation cost of Equation (10) is $\mathcal{O}(L^2 d)$ time. Therefore, Equation (9) also requires $\mathcal{O}(L^2 d)$ time. Equation (9) can be modified to have $\mathcal{O}(Ld)$ time by updating the CUR every L rounds.

Algorithm 2 The CUR Decomposition Algorithm

1: $[L] = \{1, 2, \cdots, L\}$, $\boldsymbol{X} \leftarrow \boldsymbol{0} \in \mathbb{R}^{L\times L}$;
2: **repeat**
3: **for** $i \in [L]$ **do**
4: Update $\boldsymbol{X}_{(i)}$ by Equation (9);
5: **end for**
6: **until** $\boldsymbol{X}$ converges.

IV. Theoretical Analysis

Clearly, for the online update of decremental features, the regret of OASF can be bounded by $\mathcal{O}(\sqrt{T})$ as the conventional online gradient descent with fixed feature space. Here, we introduce Lemma 1 and derive the regret bound of OASF with incremental features in Theorem 4.

Lemma 1. *Let $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_T, y_T)$ be a sequence of training instances, where $\boldsymbol{x}_t \in \mathbb{R}^{d_t}$, $d_t \leq d_{t+1}$, and $y_t \in \{-1, +1\}$ for all $t \in [T]$. Let the learning rate γ_t for the online learning with incremental features. Then, the following bound holds for any $\boldsymbol{w} \in \mathbb{R}^{d_T}$ ($d_1 \leq d_2 \leq \cdots \leq d_t \leq d_T \leq d_T$), $\sum_{t=0}^{T-1} \gamma_t(2\ell_{t+1}([\boldsymbol{w}_t; \boldsymbol{0}], (\boldsymbol{x}_{t+1}, y_{t+1})) - \gamma_t\|\boldsymbol{x}_{t+1}\|_2^2 - 2\ell_{t+1}(\Pi_{\boldsymbol{w}_{t+1}}\boldsymbol{w}, (\boldsymbol{x}_{t+1}, y_{t+1}))) \leq \|\boldsymbol{w}\|_2^2$, where $\Pi_{\boldsymbol{w}_{t+1}}\boldsymbol{w} = \Pi_{\boldsymbol{x}_{t+1}}\boldsymbol{w} \in \mathbb{R}^{d_{t+1}}$ is the sub-vector of $\boldsymbol{w}$ and has the same dimension as $\boldsymbol{w}_{t+1}$ and $\boldsymbol{x}_{t+1}$.*

Theorem 4 (Regret Bound of OASF)**.** *Let $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_T, y_T)$ be a sequence of training instances, where $\boldsymbol{x}_t \in \mathbb{R}^{d_t}$, $d_t \leq d_{t+1}$, $y_t \in \{-1, +1\}$, and $\|\boldsymbol{x}_t\|_2^2 \leq R^2$ ($R > 0$) for all $t \in [T]$. Let the learning rate $\gamma_t = \frac{\ell_{t+1}([\boldsymbol{w}_t; \boldsymbol{0}], (\boldsymbol{x}_{t+1}, y_{t+1}))}{\|\boldsymbol{x}_{t+1}\|_2^2 + \frac{1}{2\mu}}$ for the OASF with online learning with incremental features. Then, the following regret bound $\mathcal{R}_T(\boldsymbol{w})$ holds for any $\boldsymbol{w} \in \mathbb{R}^{d_T}$ ($d_1 \leq d_2 \leq \cdots \leq d_t \leq \cdots \leq d_{T-1} \leq d_T$), $\mathcal{R}_T(\boldsymbol{w}) = \sum_{t=0}^{T-1} \ell_{t+1}([\boldsymbol{w}_t; \boldsymbol{0}], (\boldsymbol{x}_{t+1}, y_{t+1})) - \sum_{t=0}^{T-1} \ell_{t+1}(\Pi_{\boldsymbol{w}_{t+1}}\boldsymbol{w}, (\boldsymbol{x}_{t+1}, y_{t+1})) \leq \sqrt{T}(\frac{\|\boldsymbol{w}\|_2}{2} + U_T) + (\frac{1}{2\mu} + R^2)\|\boldsymbol{w}\|_2^2$, where $U_T = \sqrt{\sum_{t=0}^{T-1} \ell_{t+1}^2(\Pi_{\boldsymbol{w}_{t+1}}\boldsymbol{w}, (\boldsymbol{x}_{t+1}, y_{t+1}))}$.*

Remark 2: Theorem 4 indicates that the regret bound of OASF is upper bounded by a sub-linear bound plus $(\frac{1}{2\mu} + R^2)\|\boldsymbol{w}\|_2^2$. If we assume that for any $\boldsymbol{w} \in \mathbb{R}^{d_T}$, we have $\|\boldsymbol{w}\|_2^2 \leq C^2$ ($C > 0$), we can obtain that $\mathcal{R}_T(\boldsymbol{w}) \leq \sqrt{T}(\frac{C}{2} + U_T) + (\frac{1}{2\mu} + R^2)C^2$, which implies that the regret bound of OASF enjoys $\mathcal{O}(\sqrt{T})$. Hence, the average regret bound of OASF is $\mathcal{O}(\frac{1}{\sqrt{T}})$, which will converge to zero as the number of streaming samples $T \to \infty$.

V. Experiments

A. Datasets and Evaluation Metrics

Eight real-world streaming datasets are utilized in the experiments. Table I summarizes the corresponding number of samples and features for each dataset. These datasets are also utilized as real-world streaming benchmarks in many state-of-the-art studies for mining data streams. We follow the same protocol of prior studies [7], [8] to simulate the streaming feature dynamics, where the later inputs tend to carry incrementally more features and decrementally less features. We split the original datasets into twenty chunks, where in the i-th ($i = 1, 2, \cdots, 10$) chunk only the first $i \times 10\%$ features would be retained, i.e., the first data batch will retain the first 10% features and so forth. In the i-th ($i = 11, 12, \cdots, 20$) chunk only $(21-i)\times 10\%$ features would be retained. All the datasets are implemented with 10% outliers for the experiments. We use dynamic classification error rate and running time as the comparison metrics.

B. Competing Algorithms

In the experiments, we compare OASF with four state-of-the-art online learning algorithms for data streams with streaming features: OLSF [7], OPID [10], OCDS [8], and SOAL [59].

C. Experimental Settings

We implement OASF in MATLAB. The MATLAB implementations of OLSF [7], OPID [10], OCDS [8], and

[1]http://archive.ics.uci.edu/ml/datasets.php
[2]https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

TABLE I
SUMMARY OF THE DATASETS USED IN THE EXPERIMENTS.

Dataset	#Samples	#Features	#Classes
dvd books[2]	$2,000$	$473,857$	2
internetads[2]	$1,960$	$1,558$	6
MITFace[2]	$6,977$	361	2
musk[2]	$3,062$	166	2
nslkdd[2]	$14,000$	122	2
spambase[2]	$4,601$	56	2
splice[2]	$1,000$	60	2
usps1all[2]	$7,291$	256	2

SOAL [59] are conducted from existing studies. For a fair comparison, the same experimental setup is applied to all algorithms. After the preliminary studies, we set the length of the sliding window by $L = 100$, and regularized parameters are tested by $\lambda = 20$, $\mu = 10$, $\eta = 1$, and $L = 100$ for OASF. All other parameter values are determined based upon the recommendations in existing studies. One hundred independent runs for each dataset are performed, and the average result of each method is reported. We perform all experiments on a Windows machine with a 3.7-GHz Intel Core processor and 64.0-GB main memory.

D. Dynamic Error Rate Comparisons

As shown in Fig. 1, we investigate the dynamic classification error rate of all algorithms with the progression of a data stream. For these eight data streams, the online average error rate curves of OASF consistently dominate the corresponding curves of other algorithms without much variation. The superiority of OASF over others is evident on the "dvd-books" (1st row, left panel), "internetads" (1st row, right panel), "spambase" (3rd row, right panel), and "splice" (4th row, left panel) streaming datasets. This indicates that OASF are able to capture the underlying structure of varied feature spaces associated with the ever-evolving distributions of streaming data.

E. Dynamic Running Time Comparisons

Fig. 2 presents the online average running time of one hundred independent runs for all methods on those eight datasets. The average time consumptions of SOAL, OVFM and OASF are approximately 2-3 times that of the most efficient online algorithm OLSF, making the proposed methods relatively fast to process high-throughput data streams. In addition, the average time consumed by OASF is still competitive compared with the second-order sparse online active learning method SOAL. These results validate the efficiency of OASF compared with state-of-the-art methods.

F. Parameter Sensitivity Analysis

To run OASF one needs to specify a set of parameters λ, μ, η, and L. Taking the high-dimensional dataset "internetads" as the example, we summarize the performance of OASF using the grid search. In Fig. 3, we compare the dynamic average error rates when varying these parameters. It is evident that the performances of OASF are relatively stable without much variation and the curve is flat when λ is in the range of $[0.1, 10]$. However, the average error rates of OASF are fluctuated with a general "n" shape as proper μ is vital to determine performance of the model. Similarly, the performances of OASF are relatively stable without much variation when η and L are veried in a relatively wide range. Overall, OASF is relatively robust to parameters μ and η but is somewhat sensitive to parameters λ and L.

VI. CONCLUSION

In this paper, we focus on a general and challenging setting - online learning from SDOFS with dynamically vanished, survived and new features over time by proposing OASF. By leveraging the power of the $\ell_{1,2}$-norm constraint, we exploit sparse 'non-zero' weights of the memory-aware matrix, resulting in truly sparse solutions in this complex prediction problem. We further utilize the CUR decomposition based online active learning to select informative instances in the sliding window over time. We theoretically prove the regret bound of the proposed OASF method with a sub-linear setup. Experiments on multiple benchmark datasets demonstrate the effectiveness of the proposed OASF method over three advanced state-of-the-art online methods. As part of future work, we plan to improve the stability of OASF by incorporating ensemble learning strategies. Also, we may introduce a general and adaptive robust loss function for OASF to address the challenges in SDOFS with noise. Another potential direction is to investigate local adaptive OASF for streaming data in open feature spaces with concept drifts.

ACKNOWLEDGMENT

This work has been supported in part by the National Science Foundation (NSF) under Grant Nos. IIS-2236578, IIS-2441449, and IOS-2446522.

REFERENCES

[1] S.C. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," Neurocomputing, 459, pp. 249–289, 2021.
[2] M. De Lange and T. Tuytelaars, "Continual prototype evolution: Learning online from non-stationary data streams," In CVPR, pp. 8250–8259, 2021.
[3] C. Li, X. Wang, W. Dong, J. Yan, Q. Liu, and H. Zha, "Joint active learning with feature selection via cur matrix decomposition," IEEE Trans. Pattern Anal. Mach., 41(6), pp. 1382–1396, 2018.
[4] Z. Chen, Z. Fang, J. Zhao, W. Fan, A. Edwards, and K. Zhang, "Online density estimation over streaming data: A local adaptive solution," In IEEE BigData, pp. 201–210, 2018.
[5] C. Schreckenberger, Y. He, S. Ludtke, C. Bartelt, and H. Stuckenschmidt, "Online random feature forests for learning in varying feature spaces," In AAAI, pp. 4587–4595, 2023.
[6] Z. Chen, H. Zhan, V. Sheng, A. Edwards, and K. Zhang, "Projection dual averaging based second-order online learning," In ICDM, pp. 51–60, 2022.
[7] Q. Zhang, P. Zhang, G. Long, W. Ding, C. Zhang, and X. Wu, "Online learning from trapezoidal data streams," IEEE Trans. Knowl. Data Eng., 28(10), pp. 2709–2723, 2016.
[8] Y. He, B. Wu, D. Wu, E. Beyazit, S. Chen, and X. Wu, "Online learning from capricious data streams: a generative approach," In IJCAI, pp. 2491–2497, 2019.

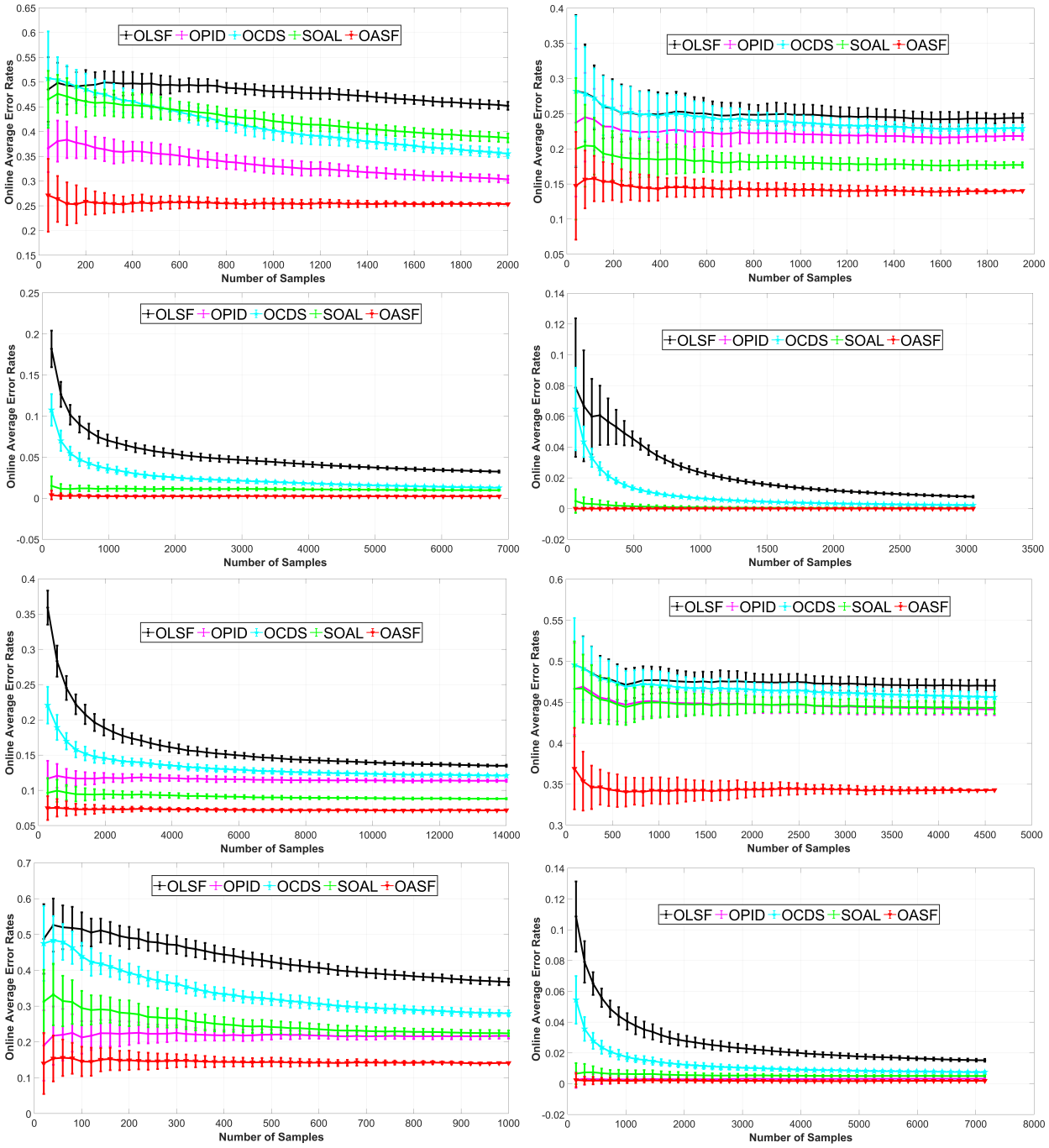

Fig. 1. Dynamic learning curves in terms of online error rates of all competing online algorithms.

[9] B.J. Hou, L. Zhang, and Z.H. Zhou, "Learning with feature evolvable streams," IEEE Trans. Knowl. Data Eng., 33(6), pp. 2602–2615, 2021.

[10] C. Hou, and Z.H. Zhou, "One-pass learning with incremental and decremental features," IEEE Trans. Pattern Anal. Mach., 40(11), pp. 2776–2792, 2017.

[11] Y. He, J. Dong, B.J. Hou, Y. Wang, and F. Wang, "Online learning in variable feature spaces with mixed data," In ICDM, pp. 181–190, 2021.

[12] Y. He, X. Yuan, S. Chen, and X. Wu, "Online learning in variable feature spaces under incomplete supervision," In AAAI, pp. 4106–4114, 2021.

[13] J. Liu, and J. Ye, "Efficient l1/lq norm regularization," arXiv preprint arXiv:1009.4766, 2010.

[14] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l 1-ball for learning in high dimensions," In ICML, pp. 272–279, 2008.

[15] E. Manzoor, H. Lamba, and L. Akoglu, "xstream: Outlier detection in feature-evolving data streams," In KDD, pp. 1963–1972, 2018.

[16] E. Beyazit, J. Alagurajah, and X. Wu, "Online learning from data streams with varying feature spaces," In AAAI, pp. 3232–3239, 2019.

[17] Z. Chen, Z. Fang, W. Fan, A. Edwards, and K. Zhang, "CSTG: An effective framework for cost-sensitive sparse online learning," In SDM, pp. 759–767, 2017.

[18] J. Langford, L. Li, and T. Zhang, "Sparse online learning via truncated gradient," J. Mach. Learn. Res., 10, pp. 777–801, 2009.

[19] Z. Chen, V. Sheng, A. Edwards, and K. Zhang, "An effective cost-

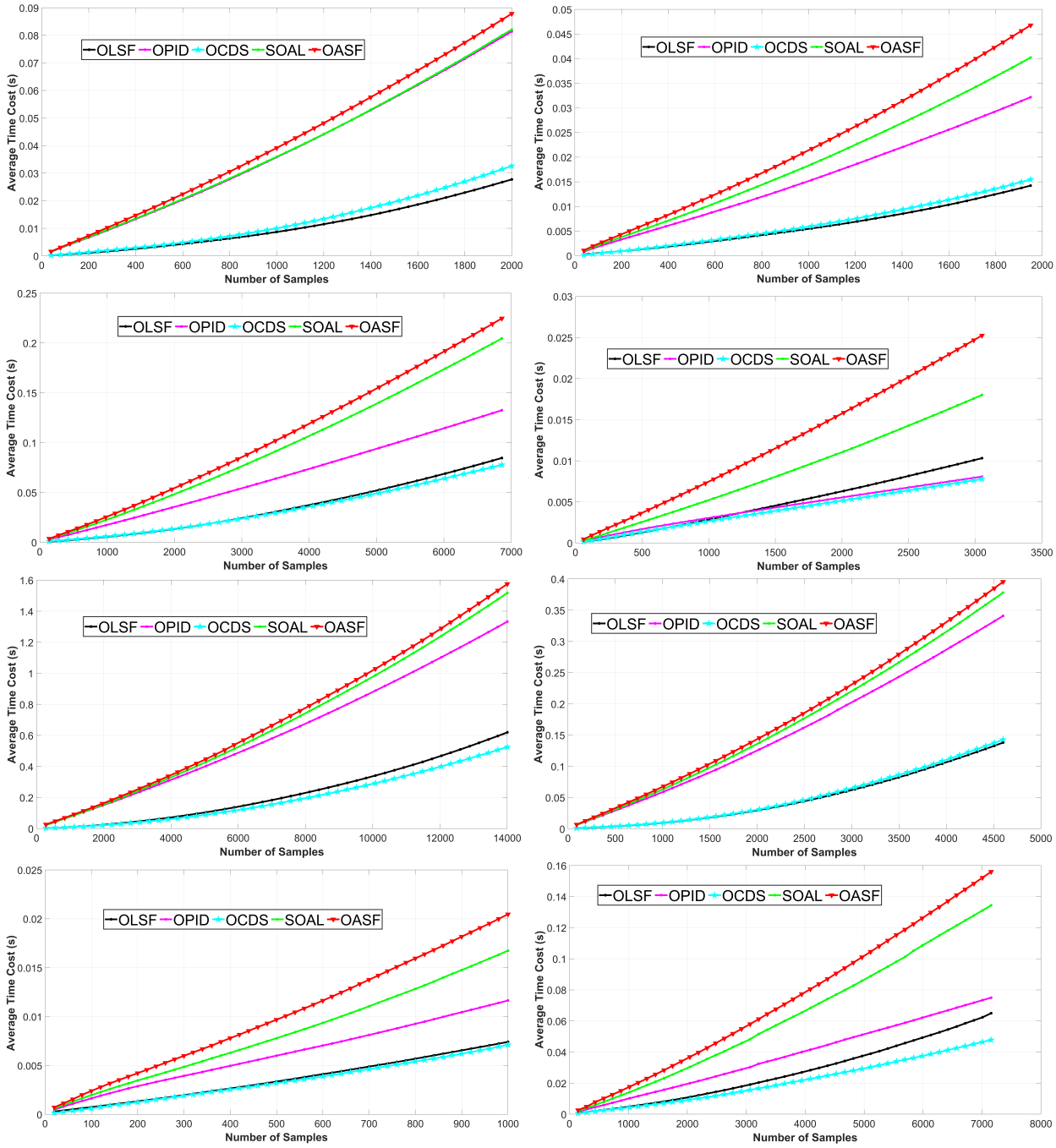

Fig. 2. Dynamic learning curves in terms of online running time (seconds) of all competing online algorithms.

sensitive sparse online learning framework for imbalanced streaming data classification and its application to online anomaly detection," Knowl. Inf. Syst., 65(1), pp. 59–87, 2023.

[20] J. Duchi, and Y. Singer, "Efficient online and batch learning using forward backward splitting," J. Mach. Learn. Res., 10, pp. 2899–2934, 2009.

[21] Y. Ma, and T. Zheng, "Stabilized sparse online learning for sparse data," J. Mach. Learn. Res., 18(1), pp. 4773–4808, 2017.

[22] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," J. Mach. Learn. Res., 11, pp. 2543–2596, 2010.

[23] S. Lee, and S.J. Wright, "Manifold identification in dual averaging for regularized stochastic online learning," J. Mach. Learn. Res., 13(55), pp. 1705–1744, 2012.

[24] A. Ushio, and M. Yukawa, "Projection-based regularized dual averaging for stochastic optimization," IEEE Trans. Signal Process., 67(10), pp. 2720–2733, 2019.

[25] B. Zhou, F. Chen, and Y. Ying, "Dual averaging method for online graph-structured sparsity," In KDD, pp. 436–446, 2019.

[26] Z. Chen, Z. Fang, V. Sheng, J. Zhao, W. Fan, A. Edwards, and K. Zhang, "Adaptive robust local online density estimation for streaming data," Int. J. Mach. Learn. Cybern., 12(6), pp. 1803–1824, 2021.

[27] N. Hurley, and S. Rickard, "Comparing measures of sparsity," IEEE Trans. Inf. Theory, 55(10), pp. 4723–4741, 2009.

[28] J. Gama, and M.M. Gaber, "Learning from data streams: processing techniques in sensor networks," Springer Science & Business Media, 2007.

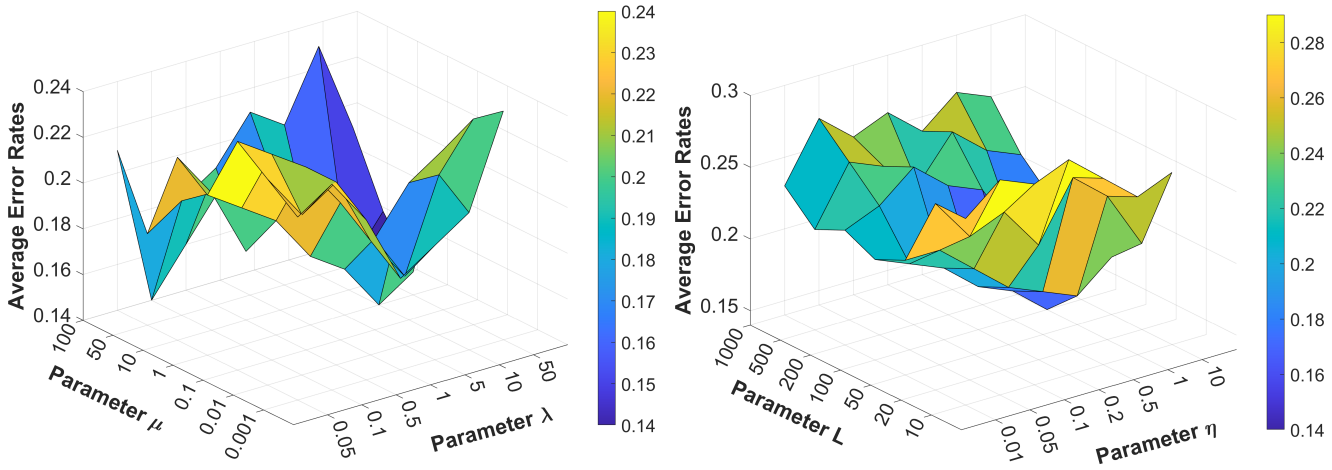

Fig. 3. The sensitivity analysis of parameters λ and μ (left: η and L) of OASF on the "internetads" dataset.

[29] S. Nittel, "Real-time sensor data streams," SIGSPATIAL Special, 7(2), pp. 22–28, 2015.

[30] Q. Shi, and M. Abdel-Aty, "Big data applications in real-time traffic operation and safety monitoring and improvement on urban expressways," Transp. Res. Part C Emerg., 58, pp. 380–394, 2015.

[31] J. Pardo, F. Zamora-Martinez, and P. Botella-Rocamora, "Online learning algorithm for time series forecasting suitable for low cost wireless sensor networks nodes," Sens., 15(4), pp. 9277–9304, 2015.

[32] C.C. Aggarwal, "Data streams: models and algorithms," Springer, 2007.

[33] S. Shalev-Shwartz, "Online learning and online convex optimization," Found. Trends Mach. Learn., 4(2), pp. 107–194, 2012.

[34] H. Yu, M.J. Neely, and X. Wei, "Online convex optimization with stochastic constraints," In NeurIPS, pp. 1427–1437, 2017.

[35] D. Leite, P. Costa, and F. Gomide, "Evolving granular neural networks from fuzzy data streams," Neural Netw., 38, pp. 1–16, 2013.

[36] Y. Meng, C. Jiang, T.Q. Quek, Z. Han, and Y. Ren, "Social learning based inference for crowdsensing in mobile social networks," IEEE Trans. Mob. Comput., 17(8), pp. 1966-1979, 2017.

[37] Z. Pan, H. Yu, C. Miao, and C. Leung, "Crowdsensing air quality with camera-enabled mobile devices," In AAAI, 31(2), pp. 4728–4733, 2017.

[38] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," IEEE Commun. Surv. Tutor., 21(3), pp. 2419–2465, 2019.

[39] Z. Chen, Y. He, D. Wu, ,H. Zhan, V. Sheng, and K. Zhang, "Robust sparse online learning for data streams with streaming features," In SDM, pp. 181–189, 2024.

[40] B.J. Hou, L. Zhang, and Z.H. Zhou, "Learning with feature evolvable streams," In NeurIPS, 33(6), pp. 1416–1426, 2017.

[41] Q. Li, S. Shah, A. Nourbakhsh, X. Liu, and R. Fang, "Hashtag recommendation based on topic enhanced embedding, tweet entity data and learning to rank," In CIKM, pp. 2085–2088, 2016.

[42] X. Liu, Q. Li, A. Nourbakhsh, R. Fang, M. Thomas, K. Anderson, R. Kociuba, M. Vedder, S. Pomerville, R. Wudali, and R. Martin, "Reuters tracer: A large scale system of detecting & verifying real-time news events from twitter," In CIKM, pp. 207–216, 2016.

[43] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, "Online feature selection with streaming features," IEEE Trans. Pattern Anal. Mach., 35(5), pp. 1178–1192, 2012.

[44] K. Yu, W. Ding, and X. Wu, "LOFS: A library of online streaming feature selection," Knowl.-Based Syst., 113, pp. 1–3, 2016.

[45] Z. Chen, Z. Fang, V. Sheng, A. Edwards, and K. Zhang, "CSRDA: Cost-sensitive regularized dual averaging for handling imbalanced and high-dimensional streaming data," In ICBK, pp. 164–173. 2021.

[46] D. You, R. Li, S. Liang, M. Sun, X. Ou, F. Yuan, L. Shen, and X. Wu, "Online causal feature selection for streaming features," IEEE Trans. Neural Netw. Learn. Syst., 2021.

[47] H. Li, X. Wu, Z. Li, and W. Ding, "Group feature selection with streaming features," In ICDM, pp. 1109–1114, 2013.

[48] Z.Y. Zhang, P. Zhao, Y. Jiang, and Z.H. Zhou, "Learning with feature and distribution evolvable streams," In ICML, pp. 11317–11327, 2020.

[49] Z. Chen, "Robust sparse online learning through adversarial sparsity constraints," In SmartCloud, pp. 42–47, 2024.

[50] Y. Liu, X. Fan, W. Li, and Y. Gao, "Online passive-aggressive active learning for trapezoidal data streams," IEEE Trans. Neural Netw. Learn. Syst., 2022.

[51] Z. Chen, V. Sheng, A. Edwards, and K. Zhang, "Cost-sensitive sparse group online learning for imbalanced data streams," Mach. Learn., 113(7), pp. 4407–4444, 2024.

[52] B. Bejar, I. Dokmanic, and R. Vidal, "The fastest $\ell_{1,\infty}$ prox in the west," IEEE Trans. Pattern Anal. Mach., 44(7), pp. 3858–3869, 2021.

[53] Z. Chen, Z. Fang, J. Zhao, W. Fan, A. Edwards, and K. Zhang, "Online density estimation over streaming data: A local adaptive solution," In IEEE BigData, pp. 201–210, 2018.

[54] M.W. Mahoney, and P. Drineas, "CUR matrix decompositions for improved data analysis," Proc. Natl. Acad. Sci., 106(3), pp.697–702, 2009.

[55] J. Bien, Y. Xu, and M.W. Mahoney, "CUR from a sparse optimization viewpoint," In NeurIPS, pp. 217–225, 2010.

[56] W. Chu, ,M. Zinkevich, ,L. Li, , A. Thomas, and B. Tseng, "Unbiased online active learning in data streams," In KDD, pp. 195–203, 2019.

[57] J. Lu, P. Zhao, and S.C. Hoi, "Online passive-aggressive active learning," Mach. Learn., 103, pp. 141–183, 2016.

[58] Z. Chen, H. Zhan, V. Sheng, A. Edwards, and K. Zhang, "Proximal cost-sensitive sparse group online learning," In IEEE BigData, pp. 495–504, 2022.

[59] S. Hao, J. Lu, P. Zhao, C. Zhang, S.C. Hoi, and C. Miao, "Second-order online active learning and its applications," IEEE Trans. Knowl. Data Eng., 30(7), pp. 1338–1351, 2017.

[60] Z. Chen, "Adaptive sparse online learning through asymmetric truncated gradient," In IEEE BigDataService, pp. 44–51, 2024.

[61] B. Krawczyk, B. Pfahringer, and M. Wozniak, "Combining active learning with concept drift detection for data stream mining," In IEEE BigData, pp. 2239–2244, 2018.

[62] J. Shan, H. Zhang, W. Liu, and Q. Liu, "Online active learning ensemble framework for drifted data streams," IEEE Trans. Neural Netw. Learn. Syst., 30(2), pp. 486–498, 2018.

[63] B. Krawczyk, and A. Cano, "Adaptive ensemble active learning for drifting data stream mining," In IJCAI, pp. 2763–2771, 2019.

[64] D. Wu, S. Zhuo, Y. Wang, Z. Chen, and Y. He, "Online semi-supervised learning with mix-typed streaming features," In AAAI, pp. 4720–4728, 2023.

[65] S. Liu, , S. Xue, ,J. Wu, ,C. Zhou, , J. Yang, , Z.Li, and J. Cao, "Online active learning for drifting data streams," IEEE Trans. Neural Netw. Learn. Syst., 34(1), pp. 186–200, 2023.

[66] K. Zhang, S. Liu, and Y. Chen, "Online active learning framework for data stream classification with density-peaks recognition," IEEE Access, 11, pp. 27853–27864, 2023.