Utilitarian Online Learning from Open-World Soft Sensing

Heng Lian[1], Yu Huang[2], Xingquan Zhu[3], Yi He[1*]
[1] William & Mary, Williamsburg, VA, USA
[2] Seagate Technology, Fremont, CA, USA
[3] Florida Atlantic University, Boca Raton, FL, USA
{hlian01, yihe}@wm.edu, yu.1.huang@seagate.com, xzhu3@fau.edu

2024 IEEE International Conference on Data Mining (ICDM) | 979-8-3315-0668-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDM59182.2024.00031

***Abstract*—Data-driven soft sensing enables to monitor and control complex industrial processes in real-time. Whereas recent data stream mining algorithms bolster predictive modeling on soft sensing data, which increment in volume and vary in feature dimensions, they operate mainly in *closed-world* settings, where all class labels must be known beforehand. This is restrictive in practical applications like semiconductor manufacturing, where new wafer defect types emerge dynamically in unforeseeable manners. This study aims to advance online algorithms by allowing learners opt to abstain from make prediction at certain costs. Our key idea is to establish a universal representation space aligning feature dimensions of incoming points while delineating a geometric shape underpinning them. On this shape, we minimize the region spanned by points of known classes through optimizing the trade-off between empirical risk and abstention cost. Theoretical results rationalize our universal representation learning design. We benchmark our approach on six datasets, including one real-world dataset of wafer fault-diagnostics collected through chip manufacturing lines in Seagate. Experimental results substantiate the effectiveness of our proposed approach, demonstrating superior performance over six state-of-the-art rival models. Code and datasets are openly accessible via an anonymous link: *https://github.com/X1aoLian/OWSS*.**

***Index Terms*—Data Streams, Online Learning, Open-World Learning, Optimal Rejection, Soft Sensing, Industrial Machine Learning**

I. Introduction

Soft sensing abounds in smart factories [1], allowing for real-time monitoring of multivariate dynamics through mapping easy-to-measure auxiliary parameters, such as temperature, pressure, or flow rates, into descriptive but hard-to-measure features like reaction rates or material properties [2]. Particularly in semiconductor manufacturing, traditional physical sensors are often inadequate for capturing the complex and time-varying interplay among materials with precision and scalability [3]. Soft sensing excels in modeling and controlling key chemical and structural features of material amalgamation at various stages such as metal deposition, photoresist coating, lithography, and etching [4]–[6].

Recently, soft-sensing models bias towards data-driven fashion [2], [7] due to their superior modeling accuracy and little reliance on expert knowledge. For example, Seagate's wafer manufacturing fault-diagnostic task can be casted into a classification paradigm, as shown in Figure 1, using soft-sensory features to predict wafer defects [8], [9]. Despite being effective, existing learning methods mainly fail short in respecting two unique traits of soft sensing data. First, the data are generated in real time and at scale [10], where the continuous data influx over time can overwhelm traditional methods with memory and computation overheads [11]–[13]. Second, the feature space describing soft-sensing data can vary and is open [14], [15]. To wit, after some periods, old sensors will wear out or become obsolete, causing pre-existing features to fade away; while, new sensors are deployed, continuously presenting new features to the classifier [16], [17].

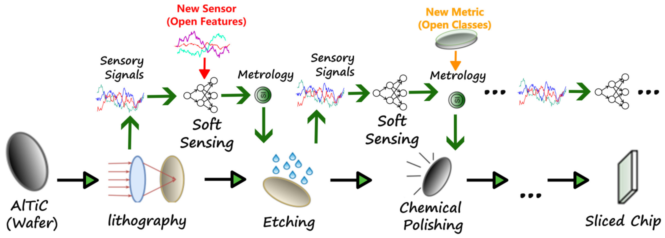


Fig. 1: Seagate wafer manufacturing pipeline using soft-sensing. Metrology for hard-to-measure metrics (target classes) are induced from raw sensory signals (features). Detailed results are reported in Section VI.

To tame the streaming and dynamic nature of soft-sensing data, one may consider adapting the recent Utilitarian Online Learning (UOL) models [15], which overcome two primary challenges. First, for new features that just emerge, they are not described by sufficient number of data points, thus initializing new model will end up with *weak* classifier. Second, for old features that fade away, ignoring the classifier trained on them results in missing information and a significant waste of previous data collection and processing efforts. UOL addresses these issues by establishing feature correlation, so as to 1) expedite new model convergence through optimal parameter initialization, and 2) leverage missing model information via old feature reconstruction. Alas, despite thriving in data stream analytics [16], [18]–[23], most UOL studies postulate a *closed-world* classification setting, which is restrictive in practice.

Specifically, the data streams in soft-sensing are likely to introduce data points belonging to new and unknown classes.

[*] Corresponding author: yihe@wm.edu

2374-8486/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDM59182.2024.00031

Such classes are often not predefined. For example, during semiconductor production, some wafer defect types can be entirely novel, such as those introduced by material impurities in electroplating and lithography [4], [5]. Given the rapid advancements in material science and the sophisticated interplay of materials, enumerating all possible defect types before starting the learning process is next to impossible. Predicting such novel defective patterns into known classes will incur substantial economic loss [6].

Surprisingly, although supporting varying feature spaces and detecting new classes [24] are two important aspects in streaming data analytics, no method has been developed to support the both. This study aims to fill this critical gap by exploring a new learning problem, which we term *Open-World Soft Sensing* (OWSS). Unlike a traditional classifier tending to make overly confident yet incorrect predictions on unknown points, our OWSS builds online learner that can *abstain* from decision-making in the rounds that instances from unknown classes are most likely to emerge. The idea conveys practical implications; for instance, the cost of abstention in wafer production, which may be that of additional physical tests or deferring to human inspection, is often more acceptable than letting suspicious chips flow into critical systems, such as medical diagnosis, hedge funds, or weapon control.

A major challenge lies in balancing the trade-off between empirical risk and abstention cost in an online fashion [25], [26]. In particular, the online learner, which observes and predicts each instance only once without back-tracking, may behave in two distinct ways: it can be *aggressive*, focusing solely on abstention cost and categorizing all instances into known classes; or it can be *conservative*, concentrating only on empirical risk and classifying most instances as unknown. In both cases, the online learner tends to make larger number of erroneous predictions compared to its hindsight optimal competitor, incurring considerable *regret* [11], [27]. To address this problem, we propose to optimize the trade-off by optimizing a bi-objective function, which minimizing the larger cost in an alternating manner until an equilibrium is reached. However, this optimization necessitates that instances from an open and varying feature space must be organized into a geometric structure [28], [29] where they share the same dimension. Note, data points arriving at different time steps may exist in disparate feature sets, making it challenging to measure their distances directly. we draw insights from UOL studies [15], [16] to discover a universal feature representation that consolidates all emerging feature information. This allows us to identify the geometric structure in the universal space by mapping incoming data points onto it, enabling consistent distance measurements across uniform dimensions.

Specific contributions of this paper are as follows:

1) This is the first study to explore the OWSS problem, where the online learner faces an input sequence living in open feature space and carrying unknown classes.
2) We propose a novel online algorithm to tackle the OWSS problem, which provably enjoys a tight generalization risk bound by learning the universal feature space in incremental fashions. Our algorithm and its theoretical analysis are presented in Sections IV and V, respectively.
3) Extensive experiments are conducted on five benchmark datasets and one real-world dataset from Seagate, which is for defect wafer diagnostics in semiconductor manufacturing lines. Our algorithm outperforms six state-of-the-art competitors, with results documented in Section VI.

II. Related Work

A. Data-Driven Soft Sensing Approaches

Soft sensors become increasingly integral in smart factories for process monitoring, quality prediction, and other critical industrial functions [1], [2]. Traditional soft sensing techniques were developed in knowledge-based fashions [30], [31], requiring domain experts to engineer a set of rules mapping raw and auxiliary time-series to descriptive soft-sensing features. Recent advances in big data analytics and increased computational capabilities have revolutionized the sensing paradigm, spurring a stride of studies exploring machine learning and data mining algorithms for soft sensing.

Seminal data-driven soft-sensing techniques include using Auto-Encoders for deep feature representation extraction and addressing issues of missing data [32], [33]. Likewise, Convolutional Neural Networks are opted for processing grid-like data to capture local feature dynamics or for transforming signals in the frequency domain [34], [35]. Recurrent Neural Networks are leveraged for variables exhibiting strong temporal dependencies, respecting sequential nonlinearity of industrial processes [36], [37]. To model correlations among soft sensors, recent advances exploit Transformers [9] and Graph Neural Networks [8]. Despite such array of data-driven soft-sensing models, there remains considerable scope to meet the dynamic needs of industrial processes. Existing studies often struggle to adapt to the fast-paced changes inherent in these settings, where new sensory features and previously unseen defect types can frequently arise. To keep these models effective, a repetitive cycle of data collection, model training and testing, and industrial deployment becomes necessary whenever there is a change in feature or label sets, leading to substantial inefficiencies in terms of time and resources previously invested. Desirably, a more agile model that can continuously learn and adapt to these ever-changing environments without the need for retraining would revolutionize the approach to soft sensing in industrial applications.

B. Utilitarian Online Learning

Online Learning (OL) enables predictive modeling from streaming data by building local empirical risk minimizers for future inputs at each time step [13], [27]. Utilitarian Online Learning (UOL) [15] extends OL by relaxing the assumption of a fixed feature space, allowing for a more adaptable learning environment. While UOL may seem related to the problem of concept-drift [38], where statistical properties of features evolve while the feature count per instance remains fixed they notably differ. Early appearance of UOL study can be traced back to [39], with successive explorations [16], [19]–[21], [40]–[43] ever since. Existing UOL studies mainly strive to solve two ill conditions incurred by the feature space dynamics. First, the emergence of new features that lack sufficient data for training results in slow convergence, with the learning coefficients boiling down to educated guesses, leading to predictive bias and errors [19], [42]. Second, when previously observed features become unobserved, the fact that the trained models cannot use past coefficients means that predictions may rely on less informative features with high missing rates [16], [20].

To overcome these challenges, UOL research has pursued three main learning strategies. The first leverages *passive-aggressive* learners [44] to redistribute learning coefficients from existing to new features within a margin-maximization regime [19], [41], [45], [46]. The second involves joint learning of the predictive model and *feature correlation* through online alternating steps to jump-start the learning of new features and reconstruct unobserved ones, thus expediting convergence [16], [20], [43], [47]. The third manages an *ensemble* of weak learners for each feature, updating or discontinuing learners based on their performance in estimating corresponding feature statistics [23], [39]. Nonetheless, most UOL studies focus on *binary* classification only, not accounting for the emergence of an unknown, new class within a streaming continuum. A more recent UOL study [48] considered this setting, but it assumes that instances of the new class are known immediately with ground-truth labels, and the difficulty is to resolve the one-shot learning challenge. Thus, the setting of [48] diverges from our OWSS problem, where instances from new classes are not known a priori, and the learner must discern whether a misclassification pertains to an existing class or an unknown one. It is noteworthy that recent studies addressing label scarcity [22], [42] do not fully address our OWSS challenge as well, because their focus remains on generating pseudo-labels to expedite online semi-supervised learners, but the classes of pseudo-labels remained binary.

C. Open-World Machine Learning

Conventional machine learning approaches typically adhere to a closed-world assumption, where the class labels remain constant across training and testing phases [25], [49], [50]. Open World Learning (OWL) challenges this paradigm by identifying test samples that likely originate from novel classes [26]. Representative applications of OWL have been predominantly domain-specific. In computer vision, Open-Set Recognition frameworks [25], [49], [51] use a one-vs-rest scheme to detect unseen objects without label assignment on the images housing them. Similarly, in natural language processing, Open Class classification approaches [52], [53] employ clustering techniques to discern unknown classes as anomalies within established document groups. These OWL methods are tailored to the characteristics of their respective data modalities. Moreover, the majority of OWL studies presume an offline training setting.

Notably, *Learning with Rejection* (LwR) models [54], [55] sometimes viewed as a subset of OWL do not align with the unique demands of our OWSS problem. LwR incorporates a rejection mechanism that effectively sidelines data instances at high risk of misclassification, with the presumption that misclassifications predominantly occur *between* two classes, such as data points residing within hard margins. Unfortunately, LwR relies on traditional classifiers that default an unlimited region volume for each known class, neglecting the fact that misclassifications may also occur *beyond* the seemingly correct side of the boundary. The further a point is from a decision boundary, the more confidently an LwR classifier will predict its class, even if the point is an outlier. Such high-confidence mis-classifications are a critical flaw when encountering unknown classes. Furthermore, the current OWL studies are predicated on the availability of a complete feature set beforehand, incompatible with our setting. In our OWSS problem, the feature set specified at the initial stages of learning could transform into an entirely different feature space over time [16], [21], rendering distance-based (e.g., clusters or margins) methods impractical.

III. Preliminary

Problem Statement. Let $\{\mathbf{x}_t, y_t\}_{t=1}^T$ be an input sequence of length T, where a point $\mathbf{x}_t := [f_1, \ldots, f_{d_t}]^\top \in \mathbb{R}^{d_t}$ arriving at t-th step is a d_t-dimensional feature vector. In an open feature space, the dimension of $\mathbf{x}_t$ can increment (*i.e.,* $d_t > d_{t-1}$) or decrement (*i.e.,* $d_t < d_{t-1}$), due to the new feature emerging and old feature vanishing, respectively. Denoted by $\mathcal{Y}_k$ and $\mathcal{Y}_u$ the labels of known and unknown classes, respectively. To ease notation, we define an instance space $\mathcal{X} = \{\mathbb{R}^{d_1} \cup \ldots \cup \mathbb{R}^{d_t}\}$ that records emerged features up to any t-th round. Any feature f_i before this round thus must be a member of $\mathcal{X}$.

At each round, the learner h_t receives a data point $\mathbf{x}_t$ and makes an immediate decision - whether this point should learned or abstained. An adversary observes the decision and reveal the true label y_t, returning a feedback by the following rule. If the prediction decision is made, the feedback is an instantaneous risk $\ell(h_t) = \ell(y_t, h_t(\mathbf{x}_t))$ revealing the discrepancy between the prediction and the ground-truth label, gauged by the adversary; otherwise, the learner incurs a cost of abstention $c(\mathbf{x}_t)$. Based on the two types of feedback, the learner updates to h_{t+1} and gets ready for the next round. Our goal is to find a series of learners $h_1, \ldots, h_T$ that minimizes the cumulative loss over T rounds, defined as follows.

$$\min_{h_1,\ldots,h_T} \sum_{t=1}^{T} \begin{cases} \ell(h_t) & \text{if } h_t \text{ predicts but } h_t(\mathbf{x}_t) \neq y_t \\ c(\mathbf{x}_t) & \text{if } h_t \text{ abstains} \end{cases}, \tag{1}$$

where we will follow prior art [56] to assume throughout that the abstention cost $c(\mathbf{x}_t)$ is a known constant and $c(\mathbf{x}_t) < \ell(h_t)$ if $y_t \in \mathcal{Y}_u$, which makes sense because otherwise an optimal yet trivial solution of Eq. (1) would be to never abstain. The designs of the learner h_t and the abstain loss $c(\mathbf{x}_t)$ are elaborated in detail in Section IV-B.

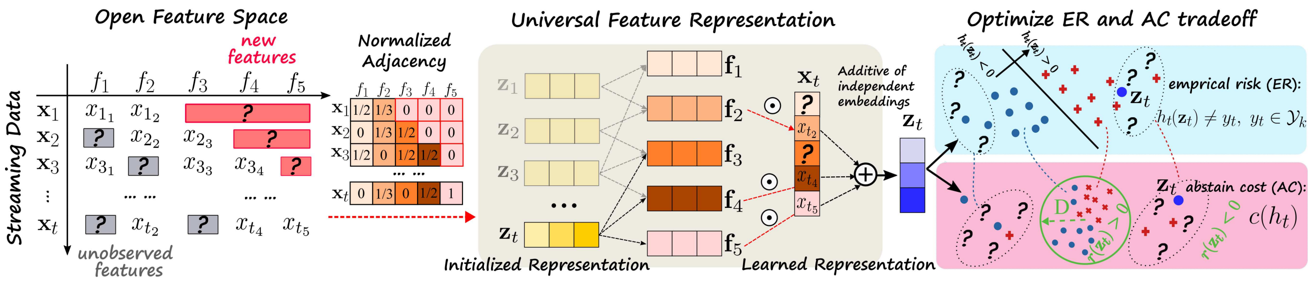

Open Feature Space
new features
Streaming Data
unobserved features
Normalized Adjacency
Universal Feature Representation
Initialized Representation
Learned Representation
Additive of independent embeddings
Optimize ER and AC tradeoff
emprical risk (ER):
$h_t(\mathbf{z}_t) \neq y_t,\ y_t \in \mathcal{Y}_k$
abstain cost (AC):
$c(h_t)$

Fig. 2: A bird's eyes view of our OWSS approach. At each time step, a data point $\mathbf{x}_t$ carrying possibly new and unobserved old features is encoded by a bipartite feature-instance graph, thereby projected onto a universal representation space as a universal feature representation $\mathbf{z}_t = \phi(\mathbf{x}_t)$ with aligned dimension (Section IV-A). An abstention function $r(\cdot)$ is learned through optimizing the tradeoff between empirical risk $\ell(h_t)$ and abstain cost $c(\mathbf{x}_t)$, determining whether the classifier h_t should predict $\mathbf{x}_t$ or simply abstain (Section IV-B).

IV. The OWSS Approach

A. Learn Universal Feature Representations

To tame the feature space dynamics, we aim to find a mapping $\phi: \mathbb{R}^{d_t} \mapsto \mathcal{U}$ that projects arriving instances onto the *universal feature space* $\mathcal{U} \in \mathbb{R}^k$, aligning their dimensions and generating universal feature representations. For any feature $f_i \in \mathcal{X}$ that has ever emerged up to the t-th round, we associate it with a feature embedding vector $\mathbf{f}_i \in \mathbb{R}^k$. We tailor objectives to learn feature embedding based on two intuitions. First, we desire the universal space $\mathcal{U}$ to increment by dynamically incorporating new feature embeddings, which capture the semantic meanings of the emerged features. However, as new feature embeddings are randomly initialized, a joint learning of both old and new features can be at risk of washing out the learned information in the old feature embeddings. We propose to disentangle the interplay among features, so as to update each embedding vector independently. To do that, we foster an *addition* relationship between the feature embeddings and data instances after mapping onto $\mathcal{U}$, namely $\mathbf{z}_t := \phi(\mathbf{x}_t) = \bigoplus_{i=1}^{d_t} \big(f_i \odot \mathbf{f}_i\big)$, where $\forall f_i \in \mathbf{x}_t$ denotes the features carried by $\mathbf{x}_t$. Denoted by $\oplus$ and $\odot$ are element-wise addition and product operators, respectively.

Second, we note that the *co-occurrence patterns* of features can reveal valuable information, beyond their semantic meanings. As the frequency of old features decreases over time in an open feature space, the co-occurrence of their combinations is expected to diminish as well. However, the persistent co-occurrence suggests their significance despite this decline in some cases. Revisit the soft-sensing data stream in semiconductor manufacturing; while some combinations of features are nearly impossible, some coincide with high possibility. To wit, a wafer cannot show defective signal in etch-condition check but pass in electroplating; rather, the two sensory features both indicate the quality of uniformity of wafer are mostly likely to fail or pass at once. To capture such feature co-occurrence patterns, we draw insights from geometry-aware representation learning [57] to model feature embeddings in a bipartite feature-instance graph, where co-occurrence features are immediate neighbours. The graph consists of two sets of nodes: the projected points $\{\mathbf{z}_t\}_{t=1}^{T}$ and the feature embeddings $\{\mathbf{f}_i\}_{i=1}^{|\mathcal{X}|}$ as shown in Figure 2. Message-passing in the graph is defined:

$$\min_{\Theta,\{\mathbf{f}_i\}} \frac{1}{T}\sum_{t=1}^{T} \Big\| \mathbf{z}_t - \mathrm{GCN}(\mathbf{A}, \{\mathbf{f}_i\}, \Theta) \Big\|_2^2, \tag{2}$$

where the matrix Θ parameterizes the graph and its (i,t)-th entry $\theta_{i,t}$ denotes learning weight of the t-th data point to be represented by the i-th feature embedding. The graph adjacency $\mathbf{A}$ encodes the feature co-occurrence, such that the immediate neighbors of $\mathbf{z}_t$ are the features carried by its original vector, namely $\mathbf{f}_i \in \mathcal{N}(\mathbf{z}_t)$ if $f_i \in \mathbf{x}_t$. In this work, we leverage graph convolutional network (GCN) with two hidden layers to implement the message passing, *i.e.,* $\mathrm{GCN}(\mathbf{A}, \{\mathbf{f}_i\}, \Theta) = \sigma\big(\mathrm{lap}(\mathbf{A})\mathbf{H}^{\top}\Theta\big)$, where $\sigma(\cdot)$ denotes a non-linear activation such as ReLU [58], $\mathrm{lap}(\cdot) = \mathbf{D}^{(-1/2)}(\cdot + \mathbf{I})\mathbf{D}^{(-1/2)}$ computes the graph Laplacian (with $\mathbf{D}$ and $\mathbf{I}$ being the degree and identify matrices, respectively), and $\mathbf{H} \in \mathbb{R}^{d_t \times k} := [\cdots, \mathbf{f}_i, \cdots]^{\top}$ is a matrix that stacks the feature embedding vectors neighboring $\mathbf{z}_t$.

Each node in GCN aggregates feature information from neighboring nodes in each layer, allowing nodes to represent both their own features and those of their local neighbors. With two layers, the GCN captures both direct and indirect feature relationships. In the first layer, feature nodes aggregate information from the point nodes that carry them, and for the second layer, these feature nodes can then indirectly exchange information with each other through the point nodes that previously aggregated their information, allowing for a more comprehensive understanding of feature co-occurrence patterns in an open feature space.

B. Optimally Abstain from Unknown Class

The dimension of any incoming instance in the universal space $\mathcal{U}$ becomes the same, thus making the distance measurement between points possible. This allows to uncover the geometric shape on which the regions covered by instances of known classes are tightly bounded. The learner h_t is realized by an abstain decision learner $r(\cdot)$, which investigates the shape and makes the abstain decision. This involves taking a projected point $\mathbf{z}_t$ and determining whether it belongs to a known or unknown class. This suggests to tailor an online binary classification task, we can envision that $r(\mathbf{z}_t) \geq 0$ if $y_t \in \mathcal{Y}_k$ and $r(\mathbf{z}_t) < 0$ if $y_t \in \mathcal{Y}_u$.

The problem now is how to train $r(\cdot)$ to achieve the trade-off between aggressive and conservative without observing the true label y_t. Our idea is to learn $r(\cdot)$ a kernel function such that, the larger the absolute value $|r(\mathbf{z}_t)|$, the more confidently the function approximates the location of point $\mathbf{x}_t$ on the geometric shape, falling into one of the following two cases: 1) the point belongs to the known-class region of minimum radius D or 2) the point is from an unknown class and far away from all the knowns. We formulate this idea into a bi-objective minimization problem as follows.

$$\min_{h_t,r} \sum_{t=1}^{T} \mathbb{1}_{[(r(\mathbf{z}_t)\geq 0)\wedge(\ell(h_t)>0)]} + \lambda \mathbb{1}_{[(r(\mathbf{z}_t)<0)\wedge(c(\mathbf{x}_t)>0)]}, \quad (3)$$

where we jointly train the online abstain decision leaner r_t and the adversary classifier returning the feedback. The indicator $\mathbb{1}_{[\cdot]}$ returns 1 if the argument is true and 0 otherwise. The parameter λ absorbs the scale difference between the empirical risk and abstention cost. Intuitively, the second term in Eq. (3) requires to minimize the rounds in which the learner abstains ($r(\mathbf{z}_t) < 0$), as the cost $c(\mathbf{x}_t) > 0$ holds $\forall t \in T$. The first term requires that, if the learner predicts ($r(\mathbf{z}_t) > 0$), the prediction $h_t(\mathbf{x}_t) = y_t \in \mathcal{Y}_k$. From a geometric perspective, optimizing Eq. (3) leads to a new representation space of input points, as shown in the rightmost panel of Figure 2, wherein the region of known classes (bounded by radius D) is tuned by including a maximal number of points over which the current learner h_t can make correct predictions. This results in an integer programming task, which provably has no effective solution in an online setting [59].

To circumvent the challenge, we instead optimize the surrogate loss [56] that convexifies Eq. (3), defined as a minimax problem:

$$\min_{h_t,r} \frac{1}{T}\sum_{t=1}^{T} \max\left\{\ell(h_t) + \frac{\lambda \cdot r(\mathbf{z}_t) + D}{2\lambda - 1},\ \lambda\Big(1 - \frac{r(\mathbf{z}_t) - D}{2\lambda - 1}\Big),\ 0\right\}, \quad (4)$$

where $\ell(h_t)$ is extended to the first term and $c(\mathbf{x}_t)$ is represented as the second term. $\lambda > 1/2$ is a tunable coefficient. As the value of λ goes larger, a smaller value of $r(\mathbf{z}_t)$ will suffice to hold **1)** the second term is more than 0, $\lambda - \lambda r(\mathbf{z}_t)/(2\lambda - 1) > 0$; **2)** the first term is smaller than the second one, $2\lambda r(\mathbf{z}_t)/(2\lambda - 1) - \lambda < 0$, *i.e.,* the second term of Eq. (4) will be minimized, which is w.r.t. the abstain decision function $r(\cdot)$ only. Note, minimizing the second term is equivalent to maximize $r(\mathbf{z}_t)$, encouraging the learner to make more predictions rather than abstention. In practice, one can chose to lower the value of λ if the abstention cost is less significant than misclassification. One such example would be in defective semiconductor detection, assuming chips for medical devices or weapon control, which has almost no tolerance to predictive errors and thus λ must be set small for spotlighting more suspicious patterns to avoid mistakes.

A constant D in the first and second terms of Eq. (4) defines the minimum radius of the known-class region. This is vital early in the learning process when $r(\cdot)$ may not perform optimally due to limited training data. An ineffective $r(\cdot)$, unable to correctly identify potentially unknown instances, leads to substantial empirical risk. This setup poses a challenge for optimization: the learner h_t struggles to discern whether the feedback is due to incorrect classifications within known classes or a failure to abstain. By constraining D, we direct the optimization to initially focus on lowering $r(\mathbf{z}_t)$, prompting more frequent abstention in the initial stages while ensuring accurate predictions for known classes. Additionally, a larger D decreases the likelihood that the second term becomes negative as $r(\mathbf{z}_t)$ increases. This encourages the model to make more predictions when empirical risk is low, making the first term of Eq. (4) negligible.

Algorithm 1: The proposed OWSS Algorithm

```
Initialize : $d_0 = 0$, radius $D$, coefficient $\lambda > 1/2$,
             GCN parameter $\Theta$, feature embedding
             matrix $\mathbf{H}_0 := [\cdots, \mathbf{f}_i, \cdots]^\top \in \mathbb{R}^{d_1 \times k}$,
             learner $h_t$
for $t = 1, \ldots, T$ do
    Receives $\mathbf{x}_t = [f_1, \ldots, f_{d_t}]^\top$ ;
    if $d_t \geqslant d_{t-1}$; // observe new feature $f_t$
     then
        Initialize feature embedding $\mathbf{f}_{d_t} \in \mathbb{R}^k$ ;
        Expend $\mathbf{H}_t \leftarrow [\mathbf{H}_{t-1} : \mathbf{f}_{d_t}]$ ;
    Map $\mathbf{z}_t \leftarrow \phi(\mathbf{x}_t) = \bigoplus_{i=1}^{d_t} (f_i \odot \mathbf{f}_i)$ ;
    if $r(\mathbf{z}_t) < 0$ ;        // learner abstains
     then Incurs $c(\mathbf{x}_t)$ ;
    else                              // learner predicts
    Suffer risk $\ell(h_t)$ if $h_t(\mathbf{z}_t) \neq y_t$  // adversary
        observes label $y_t$ and returns
        feedback /* Apply ADMM [60] for
        optimization                         */
    Optimize Eq. (2) w.r.t. $\Theta$ and $\{\mathbf{f}_i\}$, keeping $h_t$
     fixed ;
    Optimize Eq. (4) w.r.t.
```

The second term of Eq. (4) will stop to be minimized in two cases. 1) An increasing $r(\mathbf{z}_t)$ will hold $\lambda < \lambda \cdot r(\mathbf{z}_t)/(2\lambda - 1)$, making the second term negative. 2) More predictions means more misclassifications, accumulating empirical risk $\ell(h_t)$ and making $\ell(h_t) + 2\lambda r(\mathbf{z}_t)/(2\lambda - 1) - \lambda > 0$, letting the first term of Eq. (4) take over. The minimization of the first term will reduce both $\ell(h_t)$ and $r(\mathbf{z}_t)$, enhancing classification accuracy for known classes and encouraging abstention. Its optimization stops when the abstention cost of $r(\mathbf{z}_t)$ falls below a threshold, thereby activating the second term. Thus, optimization alternates between the two terms, shifting focus as one surpasses the other, until an equilibrium is reached and both terms fall below zero, concluding the optimization process. Main steps of our OWSS are summarized in Algorithm 1.

V. Theoretical Analysis

We analyze the generalization error bound to validate:

RQ1. *Will the incremental universal representation result in the minimization of the empirical risk?*

We let $h_1, \ldots, h_t \in \mathcal{H}$ be the online learner searched within a hypothesis space $\mathcal{H}$. Empirical risk for any learner is:

$$\epsilon_{\mathbb{R}^k}(h, f) = \mathbb{E}_{\mathbf{z}_t \sim \mathbb{R}^k}\left[|h(\mathbf{z}_t) - f(\mathbf{z}_t)|\right], \ \forall t \in [1, T],$$

where the error difference between $h(\cdot)$ and any other predictor $f(\cdot)$ is calculated. When $f(\cdot)$ indicates the real data-label distribution, the risk can be abbreviated as $\epsilon_{\mathbb{R}^k}(h)$. At t+1 time step, a new feature is observed and the universal space $\mathbb{R}_t^k$ starts to incorporate information carried by it, and being updated to $\mathbb{R}_{t+1}^k$. The risk of the learner of predicting instances from the old space $\mathbb{R}_t^k$ is defined as $\epsilon_{\mathbb{R}_t^k}(h)$, while its risk for instances in the new space $\mathbb{R}_{t+1}^k$ is defined as $\epsilon_{\mathbb{R}_{t+1}^k}(h)$. The optimal joint learner h^* causing the minimal errors in both spaces is defined $h^* = \text{argmin}_{h \in \mathcal{H}} \epsilon_{\mathbb{R}_t^k}(h) + \epsilon_{\mathbb{R}_{t+1}^k}(h)$. The combined error γ of the optimal learners is $\gamma = \epsilon_{\mathbb{R}_t^k}(h^*) + \epsilon_{\mathbb{R}_{t+1}^k}(h^*)$. We then have:

Theorem 1. *Denoted by $\epsilon_{\mathbb{R}_t^k}(h)$ and $\epsilon_{\mathbb{R}_{t+1}^k}(h)$ the empirical risks suffered by using h to predict samples in $\mathbb{R}_t^k$ and $\mathbb{R}_{t+1}^k$, respectively. Let $\mathcal{H}$ be a hypothesis space with VC dimension d. Write $|\mathbb{R}_t^k|$ and $|\mathbb{R}_{t+1}^k|$ as subsets of samples drawn from $\mathbb{R}_t^k$ and $\mathbb{R}_{t+1}^k$, respectively, both of size n. For any $\delta \in (0, 1)$, with probability at least $1 - \delta$,*

$$\begin{aligned} \epsilon_{\mathbb{R}_{t+1}^k}(h) \leq \epsilon_{\mathbb{R}_t^k}(h) &+ \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}\left(|\mathbb{R}_t^k|, |\mathbb{R}_{t+1}^k|\right) \\ &+ 4\sqrt{\frac{d \log(2n) + \log\left(\frac{2}{\delta}\right)}{4n}} + \gamma, \end{aligned} \tag{5}$$

where $d_{\mathcal{H}\Delta\mathcal{H}}$ gauges the H-divergence between two sets of samples.

Remark. The bound suggested by Theorem 1 establishes a relationship between the H-divergence distance of two spaces $\mathbb{R}_t^k$ and $\mathbb{R}_{t+1}^k$ and the empirical risk difference made by the same learner on two spaces. Despite both spanning a k-dimensional space, $\mathbb{R}_{t+1}^k$ is evolved from $\mathbb{R}_t^k$ through integrating a new feature. This bound indicates that the additional risks associated with the universal representation are solely influenced by the distance between the original and the incremented spaces. The more closely the updated latent space, shaped by the feature inclusion, aligns with the original space, the fewer prediction errors the universal representation will incur. This reasoning aligns with our proposition that the emergence of new features can aid in acquiring superior learn representations, overweighing its possibility of introducing noise, thereby enhancing its discriminant power. We deduce that if this new feature evolves the universal representation space by less than $\frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathbb{R}_t^k, \mathbb{R}_{t+1}^k\right)$, our OWSS algorithm can enjoy an $\mathcal{O}(\sqrt{(1/n) d \log n})$ generalization risk that will diminish as more observed data points ($n \to \infty$), thus the answer for question **RQ1**.

VI. Experiments

A. Benchmark Setup

1) **Seagate Soft-sensing Dataset:** The dataset is generated from and provided by the Seagate manufacturing lines housed in Minnesota and Ireland factories, originally containing 1) high-dimensional time-series data sensed from various manufacturing processes, and 2) categorical variables relevant to the process descriptions. Figure 1 shows the manufacturing lines with soft-sensing capacity. To wit, an Al_2O_3-TiC (AlTiC for short) wafer undergoes a series of manufacturing stages, which encompass polishing, deposition, lithography, etching, and so on. The soft sensors spreading over these stages take in the raw auxiliary time-series and output in total 94 descriptive features. At each time snapshot, the observed data point represents the wafer quality condition described by those features, as new soft-sensory variables emerging and old features expiring.

Subsequent to each of these pivotal steps, the wafer is subjected to a rigorous quality control assessment, employing precision metrology tools to evaluate critical quality indicators (KQIs). These KQIs encompass a range of critical measurements, including wafer thickness, surface roughness, film thickness, adhesion strength, pattern alignment, and etch depth, among others. Each KQI serves as a class label, with certain internal threshold determining whether the AlTiC wafer passes or fails at the assessment. To uphold stringent data security standards, KQIs are anonymized through a numerical coding system. In our study, three KQIs are identified to proxy the two known and one unknown classes.

2) **Synthetic Datasets:** We selected five benchmark datasets to evaluate the generalizability of our method. These include musk, optdigits, and satimage from UCI [61], Reuter for document classification [62], and MNIST for handwritten digit classification [63]. For UCI datasets and MNIST with fixed feature dimensions, we follow prior studies [21], [50] to simulate our open-world streaming setting. We split each dataset into 10 chunks, where the i-th chunk contains $i \times 10\%$ of the features. The entire data matrices are shuffled for each repeated experiment to enable cross-validation. In Reuter, documents in Italian and Spanish are set as two known classes, while English documents serve as the unknown sample. In MNIST, digits 1, 2, …,7 represent seven known classes, and digits 8, 9, 10 are the unknowns. In satimage, four known classes are identified. The statistics of these datasets, along with Seagate, are shown in the table below. The last column shows the ratio of samples belonging to known and unknown classes. For example, in the satimage dataset, $(2|2|1|1) : 6$ means that the ratio among four known classes is $2:2:1:1$ and the ratio between known and unknown class is $6:6$.

No.	Dataset	# Samples	# Features	$(\mathcal{Y}_k) : \mathcal{Y}_u$
1	Seagate	11,800	94	$(3 \mid 3) : 1$
2	musk	2096	166	$(10 \mid 10) : 1$
3	optdigits	3919	64	$(3 \mid 3) : 1$
4	Reuter	5000	21,531	$(2 \mid 2) : 1$
5	satimage	6435	36	$(2 \mid 2 \mid 1 \mid 1) : 6$
6	MNIST	10,000	784	$(1 \mid \ldots \mid 1) : 3$

*3) **Evaluation Metric:*** We employ three metrics for our comparative study. First, we introduce the Known-Class Cumulative Accuracy (KCCA), which measures the classification accuracy for instances identified as belonging to known classes. Assuming m instances are predicted as coming from known classes by the learner, KCCA is calculated as

$$\text{KCCA} = \frac{1}{m}\sum_{t=1}^{m} [\![y_i = \hat{y}_i]\!], \ \forall \hat{y}_t \in \mathcal{Y}_{\text{k}},$$

The purpose of KCCA is to assess the expected accuracy of instances classified as known. This metric decreases if more instances from unknown classes are incorrectly identified as known, or if errors are made in predicting known-class instances. To evaluate our capability to detect instances from unknown classes, we utilize a confusion matrix:

	$\hat{y}_t \in \mathcal{Y}_u$	$\hat{y}_t \in \mathcal{Y}_k$
$y_t \in \mathcal{Y}_u$	True Positive (TP)	False Negative (FN)
$y_t \in \mathcal{Y}_k$	False Positive (FP)	True Negative (TN)

This matrix aids in calculating the New-Class Detection Recall (Recall $= TP/(TP + FN)$) and the F1-score (F1-Score $= 2TP/(2TP + FP + FN)$), where a higher Recall indicates effective detection of instances from the new class. It is important to note that both KCCA and Recall tend to be more favorable when the learner predicts fewer instances, typically marking the majority of the data sequence as unknown. In these cases, the F1-Score score is comprehensive and decreases when many instances are deemed unknown.

*4) **Compared Methods:*** OWSS pioneers the problem of online learning with open feature spaces and unknown classes. Benchmarking the comparative study is challenging, because the existing models cannot address our OWSS problem well. To level the comparison, we identify six state-of-the-art rival models, and two variants of our approach for ablation study.

They include: **1)** OCO [64] that draws a baseline, training a linear classifier on the input data sequence and learns each new feature from scratch. **2)** OSLMF [65] is an online learning method dealing with open feature space under a semi-supervised setting. It classifies all instances into known classes in a forcible manner. **3)** GraSSNet [8] requires a tailored graph neural network (GNN) for detecting defective wafers by learning the relationship among multiple sensory features and KQIs. **4)** ECOD [66] calculate the empirical cumulative distribution and tail probability for each data feature, offering a low-cost, parameter-free method without the need for distance calculations. **5)** LUNAR [67] enhances LOF using a graph neural network to model each data point and its neighbors. **6)** ORCA [50] requires a long period of pre-training on known classes, thus can make accurate predictions on known classes and avoid assigning samples of unknown classes to them based on an uncertainty-based adaptive margin maximization. **7)** OWSS-GR (Graph Representation) is used to validate the tightness of Theorem 1, so as to rationalize our universal representation learning method. OWSS-GR leverages the pre-trained GNN in GraSSNet to align the dimension of input sequence. **8)** OWSS-IF (Idle Features) pads missing features vectors with zero values instead of learning representations.

*5) **Experimental Setup:*** We benchmark all experiments on a machine equipped with an Intel Core i9-13700K CPU and an NVIDIA GeForce RTX 3090 GPU. We implement the compared methods ECOD and LUNAR based on a public pytorch-based libraries Pyod [68]. Regarding hyper-parameters, for the compared methods, we leverage the parameter sets reported in their corresponding references if available; for ours, we grid-search for the optimal parameter sets, which are detailed in our public repository along with the model structures.

B. Results and Findings

We extrapolate from experimental results in Tables I and II and Figures 3 and 4 to answer the research questions (RQs).

RQ2. *How does our* OWSS *compare with the state-of-the-art models in terms of empirical performance?*

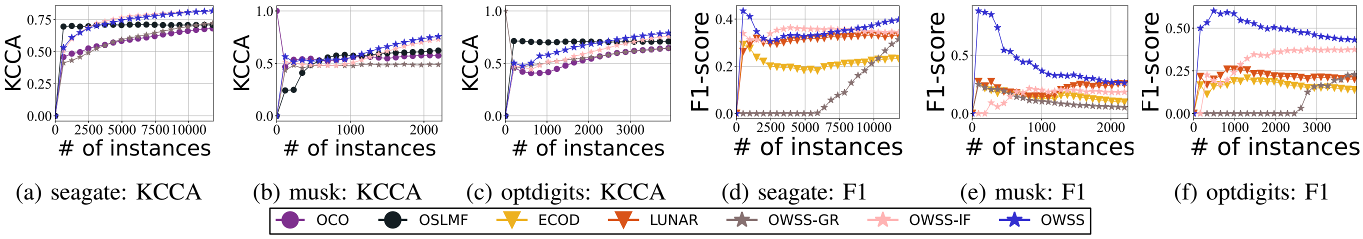
We make seven observations to answer this question. First, our OWSS achieve the best performance in all three metrics on six benchmark datasets. Our method can abstain from making predictions on most unknown-class instances with 82.5% Recall on average across all six five datasets. Meanwhile, the average F1-Score with 45.9% ensure the accurate detection is not at the high cost of not predicting a large amount of know-class instances. On average, OWSS outperforms six rival models (i.e., OCO, OSLMF, ECOD, and ORCA) by 22.3%, 44.4%, and 14.4% in terms of KCCA, Recall, and F1-Score, respectively. Second, compared to linear methods aiming at classifying known classes like OCO, OWSS enjoys significantly higher KCCA, such as Seagate dataset as shown in Figure 3a, indicating that our proposed universal representation learning can capture non-linear interplays among soft-sensory features, onto which the data points are projected for better linear separability. Such KCCA superiority of OWSS can also be observed in other datasets, reflecting our model's improved performance after excluding the interference of unknown classes. Third, we note that the performance of our OWSS is on a par with OSLMF with a 20.9% KCCA difference (OWSS is higher). Despite that OSLMF also aligns time-varying feature dimensions in a latent space, it cannot single out instances from unknown classes, thereby incurring additional empirical risk by erroneously predict unknown-class points into known classes. Forth, compared to the offline outlier detection algorithms ECOD and LUNAR, which detect unknown classes as outliers and thus cannot classify known classes, our method not only accurately classifies known classes but also achieves an average improvement of 59.5% and 18.2% in F1 and Recall, respectively.

Fifth, GraSSNet is also adept at modeling nonlinear soft-sensory feature interactions in the Seagate dataset, powered by the strong representation capability of GNN architecture; however, GraSSNet cannot perform well in streaming settings, resulting in inferior KCCA performance compared to online models, underperforming OCO, OSLMF, and our OWSS by 21.6%, 26.2%, and 37.1%, respectively. The variant OWSS-GR with this GNN architecture fails to converge and detect

TABLE I: Comparative results (mean ± standard deviation) on all four datasets in 3 metrics, averaged from 10 repeats. Not applicable (N/A) model on specific settings are indicated and justified in the footnote below table. Best results are bold.

Dataset	Metric	OCO	OSLMF	GraSSNet	ECOD	LUNAR	ORCA	OWSS-GR	OWSS-IF	OWSS
Seagate	KCCA	.681 ± .004	.708 ± .006	.421 ± .001	N/A‡	N/A‡	.326 ± .000	.722 ± .013	.814 ± .002	**.817 ± .003**
	Recall	N/A*	N/A*	N/A*	.177 ± .000	.272 ± .003	.500 ± .000	.201 ± .016	.448 ± .007	**.452 ± .005**
	F1-Score	N/A	N/A	N/A	.232 ± .000	.329 ± .005	.251 ± .000	.313 ± .015	.343 ± .003	**.396 ± .002**
musk	KCCA	.574 ± .004	.622 ± .081	.392 ± .002	N/A	N/A	.459 ± .000	.487 ± .001	.728 ± .005	**.756 ± .002**
	Recall	N/A	N/A	N/A	.108 ± .000	.402 ± .004	.500 ± .000	.029 ± .009	.843 ± .033	**.921 ± .006**
	F1-Score	N/A	N/A	N/A	.097 ± .000	.254 ± .011	.214 ± .000	.052 ± .003	.184 ± .019	**.262 ± .002**
optdigits	KCCA	.645 ± .005	.708 ± .018	.437 ± .001	N/A	N/A	.312 ± .000	.664 ± .001	.758 ± .022	**.789 ± .010**
	Recall	N/A	N/A	N/A	.109 ± .000	.170 ± .007	.500 ± .000	.145 ± .000	.786 ± .019	**.871 ± .033**
	F1-Score	N/A	N/A	N/A	.138 ± .000	.200 ± .006	.332 ± .000	.228 ± .000	.374 ± .011	**.432 ± .028**
Reuter	KCCA	.798 ± .001	.687 ± .040	N/A	N/A	N/A	.325 ± .000	N/A†	.827 ± .004	**.934 ± .001**
	Recall	N/A	N/A	N/A	.124 ± .000	.144 ± .009	.500 ± .000	N/A†	.501 ± .005	**.889 ± .001**
	F1-Score	N/A	N/A	N/A	.168 ± .000	.192 ± .013	.292 ± .000	N/A†	.337 ± .004	**.442 ± .002**

* For OCO, OSLMF, and GraSSNet, N/A is indicated in Recall and F1 as they cannot detect unknown classes and classify all instances as known, resulting in zero Recall. Removing these values prevents skewing average performance calculations, avoiding an unfair disadvantage for OCO, OSLMF, and GraSSNet. For ECOD and LUNAR, N/A is also indicated for the same reason as they can only detect unknown class and cannot classify known classes.
‡ For ECOD and LUNAR, N/A is indicated in KCCA as they only detect known classes as outliers, without the capability to distinguish known classes.
† OWSS-GR fails to work on the Reuter dataset due to its high feature dimension and sparsity, resulting in over-smoothed node representations, which has been also discussed in [69]. For the same reason we remove its extremely low values of KCCA, Recall, and F1 to prevent skewed comparison.



(a) seagate: KCCA (b) musk: KCCA (c) optdigits: KCCA (d) seagate: F1 (e) musk: F1 (f) optdigits: F1

Fig. 3: Performance trends in terms of KCCA and F1-Score of seven methods from three datasets.

the unknown class on datasets with small column (e.g., musk and optdigits) in online fashion with small recall and f1-score values, as shown in Table I and Figure 3e and 3f. Besides, GraSSNet cannot work on datasets with large features (e.g., Reuter), as its parameters increase exponentially as the number of features increases. Sixth, although ORCA possesses the modeling capability to detect new classes, it requires massive offline pretraining over known-class samples to let the model warm up to capture tight boundaries. In online setting, ORCA fails to converge and ends up with 0.5 recall on average, meaning that it either treats all instances as unknown class, or it will make prediction on all data points, thereby incurring the worst KCCA performance across all datasets. Seventh, the consistent performance of our method OWSS on five benchmark datasets and that on Seagate attests to its effective generalization across various domains.

RQ3. *How effective is the tradeoff between empirical risk and abstention cost optimized?*

Figure 4 depicts the data distribution of the Seagate dataset, with samples from two known classes (blue and red) and one unknown class (green). As shown in its left panel, known-class samples in the universal space can be separated by a linear boundary when the model is tasked solely with the classification of known classes. Both blue and red points situated further from this boundary are considered to be classified with greater confidence. However, when samples from the unknown class are likewise mapped into this space, these green points are dispersed throughout the space. Notably, those that are positioned far from the boundary are also classified by the model into known classes with high confidence, resulting in an increased empirical risk. Our OWSS method refines this approach by optimally abstaining points from the unknown class. As depicted in the right panel, instead of a single linear classification boundary, our method creates tighter boundary for each known class, where the points from two known classes in the universal space are gathered to form two clusters, with sample density increasing closer to the centers, indicated by progressively darker colors. As a result, a sample is considered to possess higher confidence only as it nears the center. When samples from the unknown class are reconstructed in this space, these green points find themselves isolated outside the two clusters, thus abstained. Compared to the OWSS-IF variant, OWSS slightly outperform by 4.2% on average which we can also observe from Figure 3a, 3b, and 3c. However, OWSS-IF outperforms OCO and OSLMF by 10.7% and 10.1%, respectively, which substantiates the effective optimization of the trade-off.

RQ4. *What is the impact of the number of known classes?*

We leverage the results from satimage and MNIST to answer this question, documented in Table II reduced from Table 1 because OSLMF and GraSSNet are tailored for binary classification and can only be extended to multiclass settings using one-vs-one or one-vs-all strategies [71]. Such strategies mainly decompose class combinatorics have no impact on evaluating the challenging of delineating and bounding the regions spanned by known-class data points, as our OWSS does in Figure 4. Due to page limits, we precluded the comparison with them and reduce Table II. In comparison with

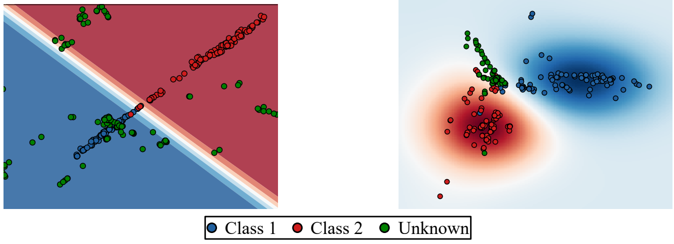


Fig. 4: Isomap visualization [70] of data points before (left) *vs.* after (right) our universal representation on Seagate dataset. *Left*: Binary classifier forcibly predicts unknown points (green) into the known class regions 1 (blue) & 2 (red). *Right*: Our OWSS model minimizing the volume of two known-class regions abstains from predicting unknowns. The unknown green points lean toward the red ones because the red points represents defective wafers, and all unknowns are likely defective, making them align more with the class 2 (defective) rather than the class 1 (qualified).

TABLE II: Results on MNIST and Satimage, reduced from Table I, as OSLMF and GraSSNet are binary classifiers.

	Metric	OCO	ORCA	OWSS
satimage	KCCA	.339 ± .015	.259 ± .000	**.553 ± .012**
	Recall	N/A	.500 ± .000	**.912 ± .026**
	F1-Score	N/A	.382 ± .000	**.601 ± .020**
MNIST	KCCA	.447 ± .034	.083 ± .000	**.708 ± .019**
	Recall	N/A	.500 ± .000	**.908 ± .068**
	F1-Score	N/A	.215 ± .000	**.623 ± .035**

OCO and ORCA, we can first observe that ORCA still presents a terrible performance with its randomness under no pre-train setting. Second, while our OWSS experiences a decrease in KCCA values with the increasing number of known classes, it still exhibits an improvement of 15% compared to the OCO on average. Third, withing increasing number of knonw classes, our OWSS remains the detection ability, proved by recall values of 91.2% and 90.8% on satimage and MNIST, respectively. This is because, in the presence of more classes, our method still tends to learn the tight boundary for each known class, and abstain any data points with low confidence, as evident from reduced F1-Score values. Despite the decrease, our model still achieves an average of 61.2% F1-Score values on two datasets, indicating its efficacy. In general, for a multi-class problem new experimental results validate that our proposed OWSS approach still remains its effectiveness and outperforms two competitors OCO and ORCA.

VII. Conclusion

This paper explored a new learning problem of *Open-World Soft Sensing* (OWSS), where predictive models are built upon input sequences characterized by varying feature dimensions and potential emergence of unknown classes. Our key idea to solve OWSS is to construct a universal representation space, on which the model learns to abstain from decision-making when presented data points that may represent unknowns. We tailor an objective function that optimizes the tradeoff between such abstention cost and empirical risk, arriving equilibrium when the regions spanned by know-class points on the universal space are tightly bounded. Theoretical and experimental results substantiate the effectiveness of our proposal.

Acknowledgement

This work has been supported in part by the National Science Foundation (NSF) under Grant Nos. IIS-2441449, IIS-2236578, IOS-2446522, IIS-2236579, IIS-2302786 and IOS-2430224, and also supported in part by the Commonwealth Cyber Initiative (CCI).

References

[1] Y. Jiang, S. Yin, J. Dong, and O. Kaynak, "A review on soft sensors for monitoring, control, and optimization of industrial processes," *IEEE Sensors Journal*, vol. 21, no. 11, pp. 12 868–12 881, 2020.

[2] Q. Sun and Z. Ge, "A survey on deep learning for data-driven soft sensors," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 5853–5866, 2021.

[3] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault detection and diagnosis in industrial systems*. Springer Science & Business Media, 2000.

[4] K. H. Yeoh, K.-H. Chew, T. L. Yoon, R. Rusi, and D. Ong, "Strain-tunable electronic and magnetic properties of two-dimensional gallium nitride with vacancy defects," *Journal of Applied Physics*, vol. 127, no. 1, 2020.

[5] A. Azarov, A. Galeckas, C. Mieszczyński, A. Hallén, and A. Kuznetsov, "Effects of annealing on photoluminescence and defect interplay in zno bombarded by heavy ions: Crucial role of the ion dose," *Journal of Applied Physics*, vol. 127, no. 2, 2020.

[6] N. Shankar and Z. Zhong, "Defect detection on semiconductor wafer surfaces," *Microelectronic engineering*, vol. 77, no. 3-4, pp. 337–346, 2005.

[7] L. Ren, Z. Meng, X. Wang, L. Zhang, and L. T. Yang, "A data-driven approach of product quality prediction for complex production systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 6457–6465, 2020.

[8] Y. Huang, C. Zhang, J. Yella, S. Petrov, X. Qian, Y. Tang, X. Zhu, and S. Bom, "Grassnet: Graph soft sensing neural networks," in *Big Data*, 2021, pp. 746–756.

[9] J. Yella, C. Zhang, S. Petrov, Y. Huang, X. Qian, A. A. Minai, and S. Bom, "Soft-sensing conformer: A curriculum learning-based convolutional transformer," in *BigData*, 2021, pp. 1990–1998.

[10] S. Petrov, C. Zhang, J. Yella, Y. Huang, X. Qian, and S. Bom, "Ieee bigdata 2021 cup: Soft sensing at scale," in *Big Data*, 2021, pp. 5780–5785.

[11] N. Cesa-Bianchi and F. Orabona, "Online learning algorithms," *Annual Review of Statistics and Its Application*, 2021.

[12] C. C. Aggarwal, *Data streams: models and algorithms*. Springer, 2007, vol. 31.

[13] H. B. McMahan, "A survey of algorithms and analysis for adaptive online learning," *Journal of Machine Learning Research*, vol. 18, no. 1, pp. 3117–3166, 2017.

[14] H. M. Gomes, J. Read, A. Bifet, J. P. Barddal, and J. Gama, "Machine learning for streaming data: state of the art, challenges, and opportunities," *ACM SIGKDD Explorations Newsletter*, vol. 21, no. 2, pp. 6–22, 2019.

[15] Y. He, C. Schreckenberger, H. Stuckenschmidt, and X. Wu, "Towards utilitarian online learning – a review of online algorithms in open feature space," in *IJCAI*, 2023.

[16] B.-J. Hou, L. Zhang, and Z.-H. Zhou, "Learning with feature evolvable streams," *NeurIPS*, vol. 30, 2017.

[17] C. Zhang, J. Yella, Y. Huang, X. Qian, S. Petrov, A. Rzhetsky, and S. Bom, "Soft sensing transformer: hundreds of sensors are worth a single word," in *BigData*, 2021, pp. 1999–2008.

[18] J. B. Gomes, M. M. Gaber, P. A. Sousa, and E. Menasalvas, "Mining recurring concepts in a dynamic feature space," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 95–110, 2013.

[19] Q. Zhang, P. Zhang, G. Long, W. Ding, C. Zhang, and X. Wu, "Online learning from trapezoidal data streams," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2709–2723, 2016.

[20] Y. He, B. Wu, D. Wu, E. Beyazit, S. Chen, and X. Wu, "Online learning from capricious data streams: a generative approach," in *IJCAI*, 2019.

[21] H. Lian, J. S. Atwood, B. Hou, J. Wu, and Y. He, "Online deep learning from doubly-streaming data," in *ACMMM*, 2022.

[22] D. Wu, S. Zhuo, Y. Wang, Z. Chen, and Y. He, "Online semi-supervised learning with mix-typed streaming features," in *AAAI*, vol. 37, no. 4, 2023, pp. 4720–4728.

[23] C. Schreckenberger, Y. He, S. Lüdtke, C. Bartelt, and H. Stuckenschmidt, "Online random feature forests for learning in varying feature spaces," in *AAAI*, vol. 37, no. 4, 2023, pp. 4587–4595.

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

[25] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1757–1772, 2012.

[26] T. E. Boult, S. Cruz, A. R. Dhamija, M. Gunther, J. Henrydoss, and W. J. Scheirer, "Learning and the unknown: Surveying steps toward open world recognition," in *AAAI*, vol. 33, no. 01, 2019, pp. 9801–9807.

[27] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

[28] A. B. Goldberg, M. Li, and X. Zhu, "Online manifold regularization: A new learning setting and empirical study," in *ECML-PKDD*, 2008, pp. 393–407.

[29] X. Wang, Z. Tu, Y. Hong, Y. Wu, and G. Shi, "No-regret online learning over riemannian manifolds," *NeurIPS*, vol. 34, pp. 28 323–28 335, 2021.

[30] B. Lin, B. Recke, J. K. Knudsen, and S. B. Jørgensen, "A systematic approach for soft sensor development," *Computers & chemical engineering*, vol. 31, no. 5-6, pp. 419–425, 2007.

[31] Y. Zhou, L. Chang, and B. Qian, "A belief-rule-based model for information fusion with insufficient multi-sensor data and domain knowledge using evolutionary algorithms with operator recommendations," *Soft Computing*, vol. 23, no. 13, pp. 5129–5142, 2019.

[32] Y. Huang, Y. Tang, and J. VanZwieten, "Prognostics with variational autoencoder by generative adversarial learning," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 1, pp. 856–867, 2021.

[33] F. Guo, W. Bai, and B. Huang, "Output-relevant variational autoencoder for just-in-time soft sensor modeling with missing data," *Journal of Process Control*, vol. 92, pp. 90–97, 2020.

[34] K. Wang, C. Shang, L. Liu, Y. Jiang, D. Huang, and F. Yang, "Dynamic soft sensor development based on convolutional neural networks," *Industrial & Engineering Chemistry Research*, vol. 58, no. 26, pp. 11 521–11 531, 2019.

[35] X. Yuan, S. Qi, Y. A. Shardt, Y. Wang, C. Yang, and W. Gui, "Soft sensor model for dynamic processes based on multichannel convolutional neural network," *Chemometrics and Intelligent Laboratory Systems*, vol. 203, p. 104050, 2020.

[36] W. Ke, D. Huang, F. Yang, and Y. Jiang, "Soft sensor development and applications based on lstm in deep neural networks," in *IEEE Symposium Series on Computational Intelligence*, 2017, pp. 1–6.

[37] A. Narayan, B. S. Mishra, P. S. Hiremath, N. T. Pendari, and S. Gangisetty, "An ensemble of transformer and lstm approach for multivariate time series data classification," in *Big Data*, 2021, pp. 5774–5779.

[38] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, 2014.

[39] B. Wenerstrom and C. Giraud-Carrier, "Temporal data mining in dynamic feature spaces," in *ICDM*, 2006.

[40] C. Hou, L.-L. Zeng, and D. Hu, "Safe classification with augmented features," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 9, pp. 2176–2192, 2018.

[41] E. Beyazit, M. Hosseini, A. Maida, and X. Wu, "Learning simplified decision boundaries from trapezoidal data streams," in *ICANN*, 2018, pp. 508–517.

[42] Y. He, X. Yuan, S. Chen, and X. Wu, "Online learning in variable feature spaces under incomplete supervision," in *AAAI*, vol. 35, no. 5, 2021, pp. 4106–4114.

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

[44] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[45] Y. Liu, X. Fan, W. Li, and Y. Gao, "Online passive-aggressive active learning for trapezoidal data streams," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[46] S. Gu, Y. Qian, and C. Hou, "Learning with incremental instances and features," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[47] Z.-Y. Zhang, P. Zhao, Y. Jiang, and Z.-H. Zhou, "Learning with feature and distribution evolvable streams," in *ICML*, 2020, pp. 11 317–11 327.

[48] C. Hou, S. Gu, C. Xu, and Y. Qian, "Incremental learning for simultaneous augmentation of feature and class," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[49] A. Bendale and T. Boult, "Towards open world recognition," in *CVPR*, 2015, pp. 1893–1902.

[50] K. Cao, M. Brbic, and J. Leskovec, "Open-world semi-supervised learning," in *ICLR*, 2022.

[51] K. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Towards open world object detection," in *CVPR*, 2021, pp. 5830–5840.

[52] G. Fei, S. Wang, and B. Liu, "Learning cumulatively to become more knowledgeable," in *KDD*, 2016, pp. 1565–1574.

[53] N. Ma, A. Politowicz, S. Mazumder, J. Chen, B. Liu, E. Robertson, and S. Grigsby, "Semantic novelty detection in natural language descriptions," in *EMNLP*, 2021, pp. 866–882.

[54] H. Mozannar and D. Sontag, "Consistent estimators for learning to defer to an expert," in *ICML*, 2020, pp. 7076–7087.

[55] C. Cortes, G. DeSalvo, C. Gentile, M. Mohri, and S. Yang, "Online learning with abstention," in *ICML*, 2018, pp. 1059–1067.

[56] C. Cortes, G. DeSalvo, and M. Mohri, "Learning with rejection," in *ALT*, 2016, pp. 67–82.

[57] K. Klemmer, N. S. Safir, and D. B. Neill, "Positional encoder graph neural networks for geographic data," in *AISTATS*, 2023, pp. 1379–1389.

[58] Y. Zhang, H. Zhu, Z. Meng, P. Koniusz, and I. King, "Graph-adaptive rectified linear unit for graph neural networks," in *WWW*, 2022, pp. 1331–1339.

[59] X. Li, C. Sun, and Y. Ye, "Simple and fast algorithm for binary integer and online linear programming," *NeurIPS*, vol. 33, pp. 9412–9421, 2020.

[60] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[61] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[62] T. Reuter and P. Cimiano, "Event-based classification of social media streams," in *ICMR*, 2012, pp. 1–8.

[63] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[64] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *ICML*, 2003, pp. 928–936.

[65] D. Wu, S. Zhuo, Y. Wang, Z. Chen, and Y. He, "Online semi-supervised learning with mix-typed streaming features," in *AAAI*, 2023.

[66] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. Chen, "Ecod: Unsupervised outlier detection using empirical cumulative distribution functions," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[67] A. Goodge, B. Hooi, S.-K. Ng, and W. S. Ng, "Lunar: Unifying local outlier detection methods via graph neural networks," in *AAAI*, vol. 36, 2022, pp. 6737–6745.

[68] Y. Zhao, Z. Nasrullah, and Z. Li, "Pyod: A python toolbox for scalable outlier detection," *Journal of Machine Learning Research*, vol. 20, no. 96, pp. 1–7, 2019. [Online]. Available: http://jmlr.org/papers/v20/19-011.html

[69] K. Ding, Y. Li, J. Li, C. Liu, and H. Liu, "Feature interaction-aware graph neural networks," *arXiv preprint arXiv:1908.07110*, 2019.

[70] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[71] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, vol. 44, no. 8, pp. 1761–1776, 2011.