Anomaly Detection in Non-stationary Data: Ensemble based Self-Adaptive OCSVM

Zahra Ghafoori, Sarah M. Erfani, Sutharshan Rajasegarar*, Shanika Karunasekera, and Christopher A. Leckie
Department of Computing and Information Systems, The University of Melbourne,
Melbourne, Australia
Email: {ghafooriz,sarah.erfani,karus,caleckie}.edu.au
*School of Information Technology, Deakin University
Melbourne, Australia
Email: sutharshan.rajasegarar@deakin.edu.au

***Abstract*—With the emergence of data streaming applications that produce large data in motion, anomaly detection in non-stationary environments has become a major research focus. The high volume of the data besides its unknown and unstable behaviour over time, limits the application of traditional anomaly detection methods that have been designed for stationary data. Moreover, basic assumptions of many existing works in the adaptive anomaly detection domain, such as the availability of labelled data over time or dealing with a known type of change in the data, are not valid for real-life applications. In this paper, we propose an *unsupervised* ensemble based anomaly detection method using One Class Support Vector Machines (OCSVMs). The proposed method is able to detect potential changes in the distribution of the normal data and adapts itself accordingly, *without requiring any external feedback*, e.g., ground truth labels. Moreover, it is able to automatically select a proper set of recent instances during learning phases, and a proper set of models during prediction phases by identifying active concepts. We evaluate our proposed method against state-of-the-art adaptive anomaly detection methods that can be applied in an unsupervised manner, on both real and synthetic non-stationary data. The results show that with considerably lower computational cost, our method outperforms the other methods.**

I. Introduction

Many existing anomaly detection (AD) methods assume that the underlying distribution of normal data is static and enough data is available for training an accurate model with acceptable generalisation error [1]. However, streaming applications, like monitoring systems, continuously generate large volumes of data that may evolve over time. Storing the past data and processing it with multiple passes is often difficult or impossible due to its memory requirements and high processing demand. Even if it is possible, the stored data may not be a good representative of future normal patterns. As a result, a model that is trained on only a part of the data suffers from any changes in the unknown underlying distribution that generates the normal data. These problems can be categorized as follows: *1)* ***Concept evolution:*** considering the normal data as a concept, we define evolution as a situation in which new aspects of the concept appear over time, which renders the current model less accurate. In this case, the model is blind on a part of the concept that increases the false negative rate (FNR) regarding the target (normal) class, and a mechanism is needed to detect and learn these new aspects of the concept. *2)* ***Concept transformation:*** a concept may transform gradually (*concept drift*) or suddenly (*concept change*) over time that can turn anomalous patterns into normal ones, and vice versa. This may affect both the false positive rate (FPR) and FNR in either way. In this case, even providing the whole data set does not guarantee high accuracy, because changes in normal behaviour might not be detected by processing the whole data set as a single batch.

Concept evolution and transformation bring new challenges to AD methods. First, these methods should be equipped with an adaptation strategy. This challenge has been dealt with using two major strategies in the data classification context [2]: (1) incremental or online learning by updating an existing model on each new available instance of data, and (2) training a new model on a fixed or variable size sliding-window of recent data (weighted or unweighted), which is done continuously or after detecting a change point. Each strategy has its own pros and cons, and it is not trivial to design a method that works for all possible situations. For example, the former case suffers if the concept changes rapidly and suddenly. In the latter case, finding an optimal window size is problematic [2]. Second, any strategy that is adopted to cope with concept evaluation or transformation should only be dependent on a limited set of recent data instances, as it is not possible to store and process all previous data. Thus, the challenge of choosing the right set of recent instances plays a vital role in the final accuracy. Finally, it is often impossible to label streaming or time-series data due to its high volume and dynamic nature, which limits the usage of any adaptive supervised AD method [3, 4, 5, 6] in practice.

Moshtaghi et al. [7] have proposed an unsupervised online adaptive AD method to cope with the aforementioned challenges. However, their method uses a continuous retraining strategy, which means it has high computational cost. Moreover, the complexity and dimensionality of the data set can affect the accuracy of their proposed method. Tan et al. [8] proposed a window-based adaptive AD method to reduce computation, but finding an optimal size for the sliding window and access to a validation set or a clean set of positive samples limits their proposed method.

In this paper, we propose a new window based unsupervised adaptive AD method, called Ensemble-based Self-Adaptive OCSVM (ESA-OCSVM), to deal with the aforementioned challenges in the following ways:

1) ESA-OCSVM applies a novel unsupervised change point detection method to avoid continuous learning. This change detection mechanism is independent of any external feedback, e.g., a validation set or any statistical change point detection test on the raw data. ESA-OCSVM keeps the track of recent instances using a sliding window of fixed size. It can decide which data from the sliding window is related to a current concept and should be used for training a new model, which obviates the need for finding an optimal size for the sliding window.
2) Our proposed method is unsupervised in the adaptation phase, because our recently proposed unsupervised parameter estimation method for OCSVMs [9], namely Quick Model Selection (QMS), is used to train each model in a timely manner and without any need for labeled data.
3) ESA-OCSVM is memory efficient, as any model that is trained on a few thousand instances provides a sparse solution. In other words, the model is only dependent on a small subset of the training instances, called support vectors, and the rest of the training set can be forgotten. This feature makes it possible to create an ensemble to deal with both concept evolution and concept transformation at the same time, which can be used to detect anomalies in a timely manner.

In the next two sections, we review the related work and present the relevant background and our problem statement. Our ESA-OCSVM method is proposed in Section IV, and evaluated against the traditional hyper-plane batch OCSVM [10], and state-of-the-art adaptive and unsupervised AD methods in Section V.

II. Related Work

Existing online [7] and periodic-retraining based [8, 11, 12, 13] adaptive AD methods are either dependent on having feedback on the actual labels of all or a part of the seen instances, or adopt a continuous learning (i.e., blind) strategy. However, manually labelling the instances of a data stream is quite expensive and is not guaranteed in real-world applications. Moreover, continuous learning, even if is implemented efficiently, does not provide any interpretation about potential change points in the data that can be even more important for analysis.

In this section, we briefly review the existing unsupervised and semi-supervised adaptive AD methods, and interested readers are referred to [14, 2, 15, 16] for more comprehensive surveys on anomaly detection and supervised drift adaptation. Zhang et al. [13] and Krawczyk et al. [12] proposed OCSVM-based approaches to deal with concept drift. They applied a continuous retraining strategy to utilize temporal correlation between recent data and keep an up-to-date model. However, retraining in the latter case was performed on weighted data while the former case used a fixed-size sliding window of recent samples. Since Krawczyk et al. [12] fed the classifier with only normal instances during the retraining phase, the reported results are valid only when a nearly clean set of positive samples is always available for retraining. Masud et al. [11] proposed a batch-incremental ensemble based method that used a clustering approach to learn a new model on every non-overlapping window of recent data. Anomaly detection was done as a byproduct of clustering, and a labelled validation set of recent data was required to check the validity of the current ensemble of models. Tan et al. [8] employed a window-based algorithm called Half-Space Tree adaptive (HSTa) that builds a HST forest using positive samples that fall inside each non-overlapping sliding window. HSTa forgets any previously trained HST after observing all the instances of the current window. Continuous retraining, finding an optimal size for the sliding window and access to a validation set or a clean set of positive samples limits the applicability of all these methods.

In contrast to the aforementioned window based methods, Moshtaghi et al. [7] proposed a method called Forgetting Factor Iterative Data Capture Anomaly Detection (FFIDCAD) that employed a continuous learning strategy to incrementally estimate an elliptical boundary that covers the normal data. This online AD method is unsupervised and its performance is independent from the actual labels of the instances. While FFIDCAD only requires one pass over each instance of data and works fairly well on simple data structures, its performance decreases for some complicated or high-dimensional data sets. Another limitation of this method is its sensitivity to a burst of anomalies: 1) the burst causes a false alarm, because the alarm is triggered when a fixed number of consecutive points are detected as anomalies, and 2) the adopted blind learning strategy treats the burst as normal instances and comes up with an invalid boundary after observing the burst.

III. Problem Statement and Background

In this section, we provide a formal statement of the problem, and briefly review the hyper-plane OCSVM method for anomaly detection as the basis of our proposed method.

A. Problem Statement

Given a data set D comprising observations $X_i \in R^d$, where $i = 1, 2, ...$, as feature vectors that are captured in a time-ordered manner, the aim is to analyse this data in the order it is received to detect and mark anomalies and normal observations using -1 and $+1$ labels, respectively. An initial part of the data set D^0 of size W_T is captured and available at the beginning, while the remainder of the data will arrive in batches of size W_B over time (see Figure 1). The first part of the data set D^0 is used for learning an initial model M^0. *Note that there is no labelled data available either during the initial training phase (stabilization period) or later when the data arrives in batches.*

The process or the underlying distribution that generates the normal data (i.e., the concept) is unknown and non-

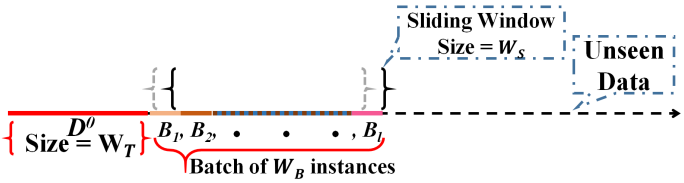

Fig. 1. Illustration of the initial data set D^0, batches and the sliding window for a given data stream.

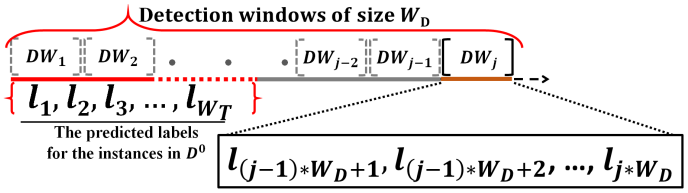

Fig. 2. Illustration of the detection windows on the consecutive predicted labels.

stationary, which means the data is prone to concept evolution or transformation:

1) Considering P_t as the underlying process that generates the normal data until time t, there may be new emerging patterns generated by P_{t+1} from time $t+1$, where $P_t \neq P_{t+1}$.
2) P_t may gradually or suddenly turn into P_{t+1} ($P_t \neq P_{t+1}$), with or without a transition period.

The learning task is to detect rare instances, which deviate from the current process that generates the normal data. Changes in the unknown underlying distribution of normal data may occur and make the model M^0 inaccurate at some point. Subsequently, a periodic retraining and adaptation strategy is needed to cope with the changes in the concept.

Next, we briefly review the hyper-plane OCSVM method for anomaly detection.

B. One-Class Support Vector Machines (OCSVMs)

The hyperplane OCSVM or ν-SVM [10] algorithm is a one-class classification method that finds a boundary around dense areas comprising the normal data [1]. In OCSVM, a training set of $X_i \in R^d$, where $i = 1, 2, ..., l$ feature vectors are projected to a potentially higher dimensional space using a feature map φ. Then, the algorithm finds a hyper-plane that separates the projected examples from the origin with the maximum possible margin. The primal quadratic problem that the OCSVM classifier solves is as follows:

$$\begin{aligned} \min_{\omega,\xi,\rho} \quad & \frac{1}{2}\|\omega\|^2 + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i, \\ \text{subject to} \quad & (\omega.\varphi(X_i)) \geq \rho - \xi_i, \end{aligned} \tag{1}$$

where $\omega \in R^d$ and $0 < \nu \leq 1$. In addition, $\xi_i \geq 0$ are slack variables that relax the problem constraints and allow some examples to fall outside the model boundary, which are treated as outliers. By using a kernel function φ, like a Gaussian kernel ($k(x, y) = e^{-\gamma\|x-y\|^2}$) with the kernel parameter γ, it is possible to apply the kernel trick and separate normal data points and outliers that are not linearly separable in the input space. After the training phase, the label of any unseen data x is simply predicted using the decision function $f(X) = sign((\omega.\varphi(X)) - \rho)$.

One of the important properties of OCSVMs is that the ν parameter is an upper bound on the fraction of outliers and a lower bound on the fraction of support vectors. In practice, this parameter and the kernel parameter γ are unknown. These parameters can be estimated using cross-validation, however this method is computationally expensive and needs a labelled training set. In contrast, QMS [9] combines K nearest neighbours (K-NN) and OCSVMs to estimate these parameters in an *unsupervised and timely* way. It uses the average distances to the K nearest neighbours for each training instance to estimate the data density around the instance. The γ parameter is set in a way that reflects the density of potential normal instances. The ν parameter is set based on the fraction of the training instances that have substantially lower density.

IV. Proposed Approach: Ensemble-based Self Adaptive OCSVM (ESA-OCSVM)

In this section, we explain the different elements of our proposed unsupervised adaptive AD method as follows: ***Parameter settings*** **-** For any training set, an estimation of the best OCSVM parameter settings $PS^* = \{\gamma^*, \nu^*\}$ is found in a *timely* manner and *without access to the actual labels* using the QMS method. ***Memory*** **-** A sliding window of size W_S keeps track of the most recent data and the older data outside this window is forgotten (see Figure 1). The size of this window can be adjusted by the user based on the memory constraints. ***Change detection*** **-** This module is designed to detect change points in an unsupervised manner based on the sequential analysis of the outlier percentage in consecutive batches, which is explained in more detail in Section IV-A. ***Learning*** **-** To cover all possible scenarios of changes in the concept, we propose to create an ensemble of models trained on non-overlapping subsets of the data instances. *One of our main contributions in this paper is proposing a novel method to select the data that should be used in training a new model when a change point is detected.* This process is explained in Section IV-B.

A. Change Point Detection

To prevent deterioration in the accuracy of anomaly detection in a non-stationary environment, our aim is to design a module to detect changes using the predicted labels. To this end, a *sequential* change point detection approach is applied on consecutive non-overlapping detection windows DW_j ($j = 1, 2, ...$) of size W_D. A given window DW_j represents predicted labels of the instances in the interval $\Delta_j = [(j-1) \times W_D + 1, j \times W_D]$. All the definitions so far are visualised in Figure 2.

Let O_j denotes the fraction of detected outliers in the detection window DW_j. During a transition between two

Algorithm 1 ESA-OCSVM: Change Point Detection Module

//initialization
set $\tilde{\mu}$, ε and h using D^0, $CS_0 = 0, flag = 0$
$j = ln = W_T/W_D$ //ln: index of last negative trend in CS_j
//for each detection window DW_j
loop
  $O_j = \sum_{i \in \Delta_j} (l_i == -1)/W_D$
  $CS_j = O_j - (\tilde{\mu} + \varepsilon) + CS_{j-1}$ //CUSUM test
  if $CS_j < 0$ **then**
    $CS_j = 0$
    $ln = j$
  end if
  if $CS_j > h$ **then**
    $flag = 1$, $CS_j = 0$, $changeIndex = ln$
  end if
  $j = j + 1$
end loop

Algorithm 2 ESA-OCSVM: Learning Module

//initialization
PS_0^*, ν_0^* //estimated using QMS on D^0
build M^0 on D^0 using PS_0^* parameter settings
delete PS_0^* and D^0
$E = \{M^0\}$, $index = 0$, $li = W_T$ //li: time-stamp of last learned instance
loop
  if $flag = 1$ **then**
    $flag = 0$, $changeStart = changeIndex \times W_D$
    if $li < changeStart$ **then**
      $index = index + 1$
      $li = ct$ //the current time-stamp
    end if
    D^{tmp} = instances in memory with time-stamp $\geq changeStart$
    PS_{tmp}^*, ν_{tmp}^* //estimated using QMS on D^{index}
    build M^{index} on D^{tmp}
    delete PS_{tmp}^* and D^{tmp}
    $E = E \cup M^{index}$
    if $index == N$ **then**
      $index = N - 1$, and remove the oldest model from E
    end if
  end if
end loop

successive data distributions P_t and P_{t+1} that occurs at an unknown point $t \in \Delta_{j=m}$, we expect a substantial number of normal instances to be labelled as outliers. Consequently, the mean value of the $\{O_j\}_{j=1}^{\infty}$ random sequence, gradually or suddenly, increases from $\tilde{\mu}$ to $\tilde{\mu} + \varepsilon$ at interval Δ_m. In our proposed method, this observation signals a change point, which leads to learning a new model.

Considering the dynamic nature of noise over different time periods and data sets, a robust approach for detecting trends in the univariate random sequence $\{O_j\}$ is needed to distinguish fluctuations in the anomaly rate from actual change points. To this end, we apply a nonparametric cumulative sum (CUSUM) test for statistical change point detection on $\{O_j\}$, which is formulated as follows [17, 18]:

$$O_j = \mu + \kappa_j I(j < m) + (\varepsilon + \eta_j) I(j \geq m), \qquad (2)$$

where $I(\cdot)$ is the indicator function and $\varepsilon > 0$. Random sequences $\{\kappa_j\}$ and $\{\eta_j\}$ are calculated on predicted labels $l_{i \in \Delta_j}$ before and after the change interval Δ_m, respectively, such that $E[\kappa_j] = E[\eta_j] = 0$, where E denotes the expected value. This implies that before the change interval Δ_m, the value of the cumulative sum defined as $CS_j = \max[0, O_j - (\tilde{\mu} + \varepsilon) + CS_{j-1}]$ stays around zero [19]. As a result, if CS exceeds a pre-defined threshold $h > 0$ at some point, a change point is detected. These steps are shown in Algorithm 1. Considerable work has been done to date to set the values of the $\tilde{\mu}$, ε and h parameters to get an acceptable detection rate and low false alarm rate. In the Empirical Evaluation Section we have used the initial observed data D^0 to set these parameters based on [19].

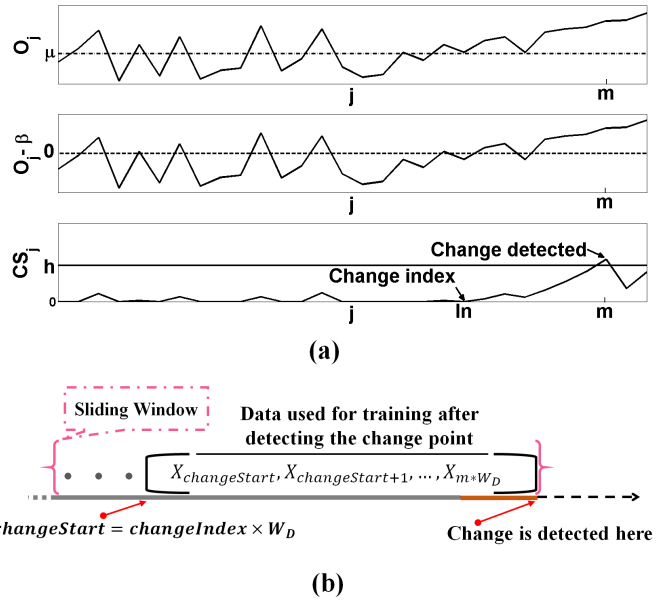

Fig. 3. Change detection and learning: (a) an example of a detected change that illustrates the important points, and (b) the corresponding selected data for training a new model.

B. Learning and Prediction

The learning procedure is explained in Algorithm 2. The main contribution of our method in learning and prediction is building an ensemble of models that is able to cope with both concept evolution and transformation problems in a single framework. This framework is able to automatically (1) select a related set of instances from the sliding window when a change is detected, and (2) choose a proper set of models from the ensemble for the purpose of doing anomaly detection. In case of concept evolution, building the ensemble makes it possible to remember previously seen patterns while learning new ones. In case of concept transformation, dependent to the pace of changes, old models are forgotten because the size of the ensemble is limited. In addition, as OCSVMs generate sparse solutions, all previously learned data can be forgotten, which means our learning and prediction methods are memory efficient and fast.

Our ESA-OCSVM algorithm initialises an ensemble E by the first trained model M^0. The size of the ensemble E is limited to N models, which is adjusted based on available memory for storing the models. Every time a change point is detected, a new model is trained and added to the ensemble. When the ensemble is full, the oldest model is deleted. Figure 3(a) shows how the data is selected every time a change point is detected. No data before the last time that a negative

trend in CS is observed, indicated by the *change index* in Figure 3(a), should be used in training a new model. This interval for a given example is shown in Figure 3(b). The intuition behind this strategy is to choose a set of instances that with higher probability are related to the current concept. For example, a change in form of a concept drift, gradually increases the value of the CS, and requires a longer time to be detected. In contrast, an emerging pattern or a sudden change may increase this value abruptly, and can be detected sooner. In the former case, instances related to the new concept are distributed over a longer period of time. However, in the latter case, the most recent instances are highly related and the older data should not be used for training a new model. We propose that the elapsed time from the last observed negative trend in the value of CS up until the time that a change point is detected is also longer in the former case in comparison with the latter one. Thus, our heuristic uses the aforementioned trick to dynamically select a set of available instances that represent the new concept, and dispenses with finding an optimal sliding window size.

Another challenge to adapt with both concept evolution and transformation is to identify if a detected change is ended and the state is stable. This challenge is important, for consecutive alarms that arise because of an ongoing change, can cause several successive training. This problem misleads the learning and prediction processes by generating similar and inaccurate models. To solve this problem, we propose that the first negative trend in the value of CS after detecting a change point can be a good indicator of entering the stable mood. If no negative trend after the detected change point is observed, the training sets of a newly trained model and the most recent model in the ensemble overlap. In this case, this model is substituted by the new model. By applying this technique, the ensemble is managed automatically to avoid the stated problem.

To make sure that a proper set of models are selected for prediction, only models that can see at least half of the data are used. This technique makes our ESA-OCSVM able to select models that are highly related to a current concept.

In the next section, we evaluate our proposed ESA-OCSVM method alongside with the existing unsupervised adaptive AD methods for both concept evolution and transformation scenarios.

V. Empirical Evaluation

We evaluate our proposed ESA-OCSVM in comparison to the FFIDCAD [7] and HSTa [8] adaptive AD methods, because to the best of our knowledge, they are state-of-the-art methods that can be applied in an unsupervised manner. In addition to these methods, a continuous retraining strategy is applied for the OCSVM algorithm. This method, hereafter Basic Ensemble-based Adaptive OCSVM (BEA-OCSVM), uses all instances inside every non-overlapping sliding window of size W_S to create a fixed-size ensemble of OCSVM models trained on the recent instances. BEA-OCSVM is implemented to show that our proposed change detection and data selection algorithms substantially improve the accuracy in comparison with blind retraining of the OCSVM. Finally, a batch version of the OCSVM is applied when 20%, 40%, and 80% of the whole examined data sets, are available for training a new batch model. The aim of this experiment is to demonstrate that even being able to access and process all of the past data does not necessarily result in higher accuracy.

TABLE I
Summary of the data sets

Data set	#Features	#Instances
O-News-P	59	39,644
GSAD-S1	8	31,910
GSAD-S9	8	31,910
Shuttle	9	58,000
Forest	54	30,000
USPS	256	7,291
LG-S10	6	38,860
IBRL-S9	4	45,204
STB-S13	7	29,485
Banana	3	22,000

The comparison is made based on two measures: (1) the accuracy of the examined methods over time, and (2) the fraction of batches that have caused retraining. The last measure is important because it reflects the computational demand of the different methods in dealing with the concept evolution and transformation.

A. Experimental Setup

The parameters of the QMS method are set based on [9]. For simplicity, the size of the sliding window W_S and the initial training set W_T are considered equal as well as the size of the detection window W_D and batches W_B. The latter cases are set to 100 instances. The former cases, $W_S = W_T = 2,000$, and the size of the ensemble $N = 6$ are chosen according to [11], to show that the performance of our ESA-OCSVM is not strongly dependent to the choice of these values. Due to space limitations, we summarize our studies on the choice of these parameters as follows: the values should not be set very large because it converts our method to batch learning.

We let all the methods first see an incoming batch of $W_B = 100$ instances before predicting the corresponding labels of these instances. For FFIDCAD and HSTa, adaptation is performed continuously, i.e., by receiving every batch, while our proposed method adapts itself once a change point is detected. The default settings of FFIDCAD [7] and HSTa [8] are used. Like any score-based anomaly detection method, a limitation of HSTa is in setting a cut-off threshold for converting the scores to the predicted labels [20]. When no ground truth labels or knowledge about the fraction of outliers are available over time, and the prediction is performed in an ongoing manner rather than batch processing, finding the best cut-off threshold is even more challenging. To handle this problem for each data set, *we assume that the actual labels of the initial training set D^0 are available for HSTa to find an effective cut-off threshold*, which is then fixed for the rest of the data stream. For each examined data set in our experiments, the

TABLE II
THE AUC OF OUR ESA-OCSVM IN COMPARISON WITH THE OTHER METHODS ON THE LAST 80%

Data set	**ESA-OSCVM**		BEA-OSCVM		OCSVM		FFIDCAD		HSTa	
	Avg	std	Avg	std	Avg	std	Avg	std	Avg	std
O-News-P	0.974	0.003	0.972	0.000	0.927	0.003	0.636	0.001	**0.990**	0.006
GSAD-S1	**0.989**	0.002	0.978	0.001	0.966	0.001	**0.989**	0.001	0.972	0.006
GSAD-S9	**0.993**	0.002	0.975	0.001	0.966	0.001	0.990	0.001	0.950	0.006
Shuttle	**0.996**	0.001	0.996	0.000	0.989	0.001	0.949	0.001	**0.996**	0.002
Forest	0.980	0.002	0.965	0.000	0.794	0.003	0.786	0.001	**0.993**	0.002
USPS	**0.981**	0.014	0.926	0.000	0.720	0.010	0.588	0.014	0.670	0.020
LG-S10	**0.981**	0.002	0.823	0.007	0.660	0.003	0.980	0.001	0.835	0.034
IBRL-S9	**0.985**	0.003	0.772	0.011	0.579	0.005	0.978	0.001	0.815	0.014
STB-S13	0.983	0.002	0.694	0.006	0.567	0.021	**0.988**	0.001	0.802	0.047
Banana	**0.930**	0.005	0.924	0.004	0.803	0.006	0.882	0.005	0.884	0.013
Avg	**0.979**	0.004	0.903	0.003	0.797	0.005	0.877	0.003	0.891	0.015

cut-off threshold is set to a value that maximises the accuracy on the initial data set D^0.

For our ESA-OCSVM, the first model M^0 is trained on D^0, which includes $2,000$ instances. Then, $\{O_j\}_{j=1}^{20}$ is computed for each of the 20 detection windows of size $W_D = 100$. The values of $\tilde{\mu}$, ε, and h are set to the average of $\{O_j\}_{j=1}^{20}$, 0.1 and 5 times its corresponding standard deviation, respectively [19]. These settings detect any increase in $\tilde{\mu}$ equal to or greater than 10% of the standard deviation of $\{O_j\}_{j=1}^{20}$.

To evaluate all methods over time, the evaluation is performed after observing 20%, 40%, and 80% of the examined data sets, respectively on the rest 80%, 60%, and 20% data. We use the Receiver Operating Characteristic (ROC) curve and the corresponding Area Under the Curve (AUC) to evaluate the accuracy as it is insensitive to the class imbalance problem. The reported results were averaged over 500 runs.

B. Data Sets

We ran our experiments on nine real and one synthetic data sets. Table I summarises the dimensions of the data sets.

The first five real data sets are extracted from the UCI Machine Learning Repository[1], namely Gas Sensor Array Drift (GSAD)[2], Online News Popularity (O-News-P), Forest Cover Type (Forest), and Shuttle. The other four real data sets are sensor measurements from IBRL[3], Le Genepi (LG) and St Bernard (STB) environmental monitoring systems[4], and USPS[5] data set.

For the sensor-type data sets, sensors with non-stationary behaviour are selected as follows: sensors 1 and 9 from GSAD, sensor 9 from IBRL, sensor 10 from LG, and sensor 13 from STB, and only the non-missing measurements are used. For the Forest and USPS data sets, the emergence of the different classes in the data is manipulated to create synthetic drift. Since the Forest data set is very large, which slows down the batch OCSVM considerably, a part of this data set is selected randomly and used in the experiments.

[1]https://archive.ics.uci.edu/ml/datasets.html
[2]The measurements from two sensors of this data set are used.
[3]http://db.csail.mit.edu/labdata/labdata.html
[4]http://lcav.epfl.ch/page-86035-en.html
[5]https://www.otexts.org/1577

We also generated a synthetic Banana data set comprising 11 states, each of which have $2,000$ normal instances and about 6° rotational drift in comparison with their nearby states. All the data sets are scaled in the range $[0, 1]$. For all data sets, 5% anomalies are perturbed around the normal data by adding or subtracting uniform noise. In this way we know the actual labels and we are able to evaluate the methods.

C. Results and Discussion

Tables II to IV report the accuracy of the examined methods over time. Our proposed ESA-OCSVM is stable over time and outperforms all the other methods on average. For the cases where another method has a higher accuracy, the average difference is marginal (0.009). The findings confirm that the FFIDCAD method works well on simple data sets like GSAD, LG, STB, and IBRL, but for the rest of the data sets that are more complicated and have higher dimensions, its accuracy decreases significantly. Although for the O-News-P, Shuttle and Forest data sets that have minor emerging patterns, the HSTa method have worked very well, its accuracy for the rest of the data sets is lower and have larger variance over time.

We conduct the Wilcoxon signed-rank test on the findings of Tables II to IV to identify the statistical significance of the differences between the accuracy of our ESA-OCSVM and the rest of the methods over time. In each comparison, the aim is to investigate to what extent the null hypothesis H_0, which indicates that there is no difference between the first and second methods in terms of their accuracy, can be rejected. For each comparison, the test returns the sum of positive ranks of the first method (R^+), the sum of negative ranks of the first method (R^-), and the p-value. These statistics gives us sufficient evidence to reject or accept the null hypothesis H_0. The p-value represents the lowest level of significance of a hypothesis that results in a rejection. For all the comparisons in this study the significance level α is set to 0.05. A p-value less than α indicates that the null hypothesis H_0 can be rejected.

Table V demonstrates that the null hypothesis H_0 can be rejected in all the comparisons, as the corresponding p-values are less than the significance level $\alpha = 0.05$. In other words, our ESA-OCSVM significantly outperforms the rest of the

TABLE III
THE AUC OF OUR ESA-OCSVM IN COMPARISON WITH THE OTHER METHODS ON THE LAST 60%

Data set	**ESA-OSCVM**		BEA-OSCVM		OCSVM		FFIDCAD		HSTa	
	Avg	std	Avg	std	Avg	std	Avg	std	Avg	std
O-News-P	0.974	0.003	0.973	0.000	0.953	0.001	0.635	0.002	**0.994**	0.004
GSAD-S1	**0.989**	0.002	0.975	0.001	0.960	0.001	0.988	0.001	0.972	0.005
GSAD-S9	**0.992**	0.002	0.969	0.001	0.960	0.001	0.989	0.001	0.942	0.004
Shuttle	0.997	0.001	0.996	0.000	0.990	0.001	0.950	0.001	**0.998**	0.001
Forest	0.981	0.003	0.977	0.001	0.914	0.002	0.783	0.001	**0.999**	0.001
USPS	**0.981**	0.016	0.934	0.001	0.793	0.009	0.553	0.015	0.711	0.025
LG-S10	**0.980**	0.002	0.798	0.009	0.732	0.005	0.979	0.001	0.835	0.038
IBRL-S9	**0.984**	0.003	0.728	0.015	0.528	0.001	0.972	0.002	0.785	0.014
STB-S13	0.984	0.002	0.724	0.007	0.755	0.011	**0.987**	0.001	0.801	0.044
Banana	**0.928**	0.006	0.921	0.004	0.828	0.006	0.881	0.006	0.882	0.011
Avg	**0.979**	0.004	0.900	0.004	0.841	0.004	0.872	0.003	0.892	0.015

TABLE IV
THE AUC OF OUR ESA-OCSVM IN COMPARISON WITH THE OTHER METHODS ON THE LAST 20%

Data set	**ESA-OSCVM**		BEA-OSCVM		OCSVM		FFIDCAD		HSTa	
	Avg	std	Avg	std	Avg	std	Avg	std	Avg	std
O-News-P	0.976	0.005	0.977	0.001	0.970	0.000	0.641	0.003	**0.999**	0.001
GSAD-S1	0.973	0.007	0.932	0.001	0.869	0.001	**0.977**	0.002	0.922	0.013
GSAD-S9	**0.976**	0.006	0.908	0.004	0.869	0.001	0.973	0.002	0.826	0.039
Shuttle	0.996	0.001	0.995	0.000	0.990	0.000	0.950	0.001	**0.998**	0.001
Forest	0.982	0.004	0.984	0.001	0.968	0.001	0.744	0.002	**1.000**	0.000
USPS	**0.994**	0.014	0.996	0.001	0.892	0.006	0.546	0.027	0.875	0.048
LG-S10	**0.978**	0.004	0.754	0.014	0.871	0.006	0.975	0.001	0.878	0.036
IBRL-S9	**0.984**	0.005	0.722	0.016	0.548	0.006	0.966	0.003	0.878	0.013
STB-S13	0.983	0.003	0.579	0.004	0.552	0.004	**0.984**	0.001	0.577	0.062
Banana	0.931	0.010	0.917	0.007	**0.932**	0.010	0.884	0.010	0.911	0.021
Avg	**0.977**	0.006	0.876	0.005	0.846	0.004	0.864	0.005	0.886	0.023

methods. Moreover, our proposed method has higher rank in comparison with the rest of the methods. It significantly improves both batch OCSVM and BEA-OCSVM with a very small p-value, which confirms our earlier statements about the unreliability of applying batch learning and the blind retraining strategy for OCSVMs in non-stationary environments.

The measurements of sensors in IBRL, LG, and STB change rapidly, and we can categorise them as data sets including concept transformation, while in the rest of the data sets, new patterns emerge over time. Our method demonstrated superior performance for all the data sets, hence, we conclude that the proposed algorithms can be used in both the concept evolution and transformation cases.

Table VI shows the percentage of batches that have caused retraining in our proposed ESA-OCSVM for the examined data sets. The average, highest and lowest values of the percentages reported by Table VI for our ESA-OCSVM are 30%, 67.6%, and 1.4% of batches, respectively. For the rest of the adaptive methods, 100% of batches have caused retraining, which entails substantially higher computational cost. For the Shuttle data set with $58,000$ instances, many unnecessary updates are performed by the rest of the methods, while our method performed retraining only in 1.4% of batches and still has the highest accuracy. For IBRl-S9 and STB-S13 that have higher dynamicity, the percentage of batches that have caused retraining are still more than 30% less than the rest of the methods. The results are well correlated with the potential number of changes in the data, because: sensors 1 and 9 of GSAD have minor drifts over time; USPS, Forest and Banana data sets have a few number of synthetic drifts; O-News-P is a data set gathered through 2 years and includes the features of published articles of a journal, which is prone to a small level of change; while the Shuttle data set seems nearly static (the accuracy of batch training on this data set over the different time periods is reasonably stable and high). In contrast, the measurements of the aforementioned sensors in IBRL, LG, and STB change rapidly.

TABLE V
THE RESULTS OF WILCOXON TEST FOR OUR ESA-OCSVM IN COMPARISON WITH THE OTHER METHODS

ESA-OCSVM vs.	R^+ (our method)	R^- (others)	P-value
BEA-OCSVM	420.5	14.5	0.000011
OCSVM	464.0	1.0	0.000002
FFIDCAD	404.5	30.5	0.000058
HSTa	389.4	45.5	0.000283

VI. CONCLUSION

In this paper, we proposed a novel ensemble based method for non-stationary environments when performing anomaly detection using the OCSVM method. This method utilized the

TABLE VI
THE PERCENTAGE OF THE BATCHES THAT CAUSE RETRAINING IN OUR ESA-OCSVM

Data set	%	Data set	%
O-News-P	26.8	USPS	13.2
GSAD-S1	5.6	LG-S10	35.7
GSAD-S9	7.6	IBRL-S9	67.6
Shuttle	1.4	STB-S13	67.4
Forest	23.9	Banana	27.9

observed fraction of outliers to detect change points, independently from any external feedback (e.g., ground truth labels or statistical analysis of the raw data) during the operational phase. A novel method is also proposed to select only highly related instances of the sliding window, which removes the need for finding an optimal window size for each individual data set. Experimental evaluation showed that our adaptive OCSVM significantly improves the traditional batch OCSVM method as well as the existing adaptive methods that can be applied in an unsupervised manner.

REFERENCES

[1] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, October 2004.

[2] 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, April 2104.

[3] J. Gao, W. Fan, J. Han, and S. Y. Philip, "A general framework for mining concept-drifting data streams with skewed distributions," in *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)*. SIAM, 2007, pp. 3–14.

[4] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2009, pp. 139–148.

[5] S. Chen and H. He, "Sera: selectively recursive approach towards nonstationary imbalanced stream data mining," in *Proceedings of the 2009 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2009, pp. 522–529.

[6] T. R. Hoens and N. V. Chawla, "Learning in non-stationary environments with class imbalance," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2012, pp. 168–176.

[7] M. Moshtaghi, C. Leckie, S. Karunasekera, J. C. Bezdek, S. Rajasegarar, and M. Palaniswami, "Incremental elliptical boundary estimation for anomaly detection in wireless sensor networks," in *Proceedings of the 11th International Conference on Data Mining (ICDM)*. Vancouver, Canada: IEEE, July 2011, pp. 467–476.

[8] S. C. Tan, K. M. Ting, and T. F. Liu, "Fast anomaly detection for streaming data," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 22. Citeseer, 2011, p. 1511.

[9] Z. Ghafoori, S. Rajasegar, S. M. Erfani, S. Karunasekera, and C. A. Leckie, "Unsupervised parameter estimation for one-class support vector machines," in *Proceedings of the 20th Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, April 2016.

[10] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computing*, vol. 13, no. 7, pp. 1443–1471, July 2001.

[11] M. M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham, "Classification and novel class detection in data streams with active mining," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2010, pp. 311–324.

[12] B. Krawczyk and M. Woźniak, "One-class classifiers with incremental learning and forgetting for data streams with concept drift," *Soft Computing*, pp. 1–14, October 2014.

[13] Y. Zhang, N. Meratnia, and P. Havinga, "Adaptive and online one-class support vector machine-based outlier detection techniques for wireless sensor networks," in *Proceedings of the IEEE 23rd International Conference on Advanced Information Networking and Applications (WAINA) Workshops/Symposia*. IEEE, May 2009, pp. 990–995.

[14] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, July 2009.

[15] C. O'Reilly, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Anomaly detection in wireless sensor networks in a non-stationary environment," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1413–1432, 2014.

[16] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly detection in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, no. 4, pp. 34–40, 2008.

[17] T. Peng, C. Leckie, and K. Ramamohanarao, "Proactively detecting distributed denial of service attacks using source ip address monitoring," in *Proceedings of Networking 2004*, vol. 3042. Springer, 2004, pp. 771–782.

[18] E. Brodsky and B. S. Darkhovsky, *Nonparametric methods in change point problems*. Germany: Springer, 1993.

[19] N. Ye, "Univariate control charts," in *Data mining: theories, algorithms, and examples*. CRC Press, 2013, ch. 16.

[20] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 159–170, 2010.