IISE TRANSACTIONS
2017, VOL. 49, NO. 11, 1–17
https://doi.org/10.1080/24725854.2017.1347984

Taylor & Francis
Taylor & Francis Group

Ensemble-Bayesian SPC: Multi-mode process monitoring for novelty detection

Marcelo Bacher and Irad Ben-Gal

Department of Industrial Engineering, Tel Aviv University, Tel Aviv, Israel

ABSTRACT

We propose a monitoring method based on a Bayesian analysis of an ensemble-of-classifiers for Statistical Process Control (SPC) of multi-mode systems. A specific case is considered, in which new modes of operations (new classes), also called "novelties," are identified during the monitoring stage of the system. The proposed Ensemble-Bayesian SPC (EB-SPC) models the known operating modes by categorizing their corresponding observations into data classes that are detected during the training stage. Ensembles of decision trees are trained over replicated subspaces of features, with class-dependent thresholds being computed and used to detect novelties. In contrast with existing monitoring approaches that often focus on a single operating mode as the "in-control" class, the EB-SPC exploits the joint information of the trained classes and combines the posterior probabilities of various classifiers by using a "mixture-of-experts" approach. Performance evaluation on real datasets from both public repositories and real-world semiconductor datasets shows that the EB-SPC outperforms both conventional multivariate SPC as well as ensemble-of-classifiers methods and has a high potential for novelty detection including the monitoring of multimode systems.

ARTICLE HISTORY
Received 28 September 2015
Accepted 30 May 2017

KEYWORDS
TSPC; ensemble; subspaces; bayes; novelty detection

1. Introduction

The need for new methods for the monitoring of complex processes that generate high-dimensional data, which occur, for example, in systems with a large number of sensors or attributes, has been continuously growing in recent years (Chandola *et al.*, 2007; Kenett and Zacks, 2014). One research area that addresses this problem is Multivariate Statistical Process Control (MSPC), which contains a large body of related works (see, e.g., Chiang *et al.* (2001), Ben-Gal and Singer (2004), and Montgomery (2008)). Many MSPC methods face the dimensionality challenge by first applying dimensionality reduction techniques, such as Principal Component Analysis (PCA) or partial least squares, and then monitoring the data-rich systems on smaller dimensions. Often, these methods follow the SPC convention, in which the data collected during the normal operating mode of the process is used to learn a representative "in-control" statistical model, while during the monitoring stage, outliers that deviate from the learned "in-control" model are used for anomaly detection.

A slightly different challenge arises when monitoring a multi-mode system, in which the occurrence of new data patterns might represent a new mode that formerly has not been present during the learning stage. Detecting new data classes not represented by the "in-control" statistical model is also known as *novelty detection* (see, e.g., Chandola *et al.* (2007)). Although the term *novelty detection* has been used interchangeably with *anomaly detection* and *outlier detection*, the latter two terms are often related to sporadic abnormal data samples that are caused by white noise or "special cause" factors resulting in noisy spikes. Often, they do not address a situation where an *entirely* new and internally correlated data class appears during the monitoring stage. Thus, although anomaly detection and outlier detection methods find use in practical conventional process monitoring, these methods are less efficient for the monitoring of multi-mode systems, particularly when the system modes are correlated over some subsets of high-dimensional datasets. The reasons for such inefficiencies include the following:

1. They occur because the novelties (i.e., the new system modes), unlike conventional anomalies, are not well represented by the SPC "white noise assumption."
2. Applying dimensionality reduction techniques *before* the novelties are present in the data can result in a lower-dimension projection where the novelties cannot be detected.
3. The high computational effort required in order to find the subspace where the novelty is correlated, which grows exponentially in the data dimension.

A proposed approach to tackle the monitoring challenge in high-dimensional data was the use of an ensemble of classifiers (e.g., Wang *et al.* (2003)). This approach was supported by several studies in recent years, showing that ensemble-based anomaly detection techniques that use subsets of data attributes can lead to a better performance in comparison with the classic approaches (e.g., Aldrich and Auret (2013); Aggarwal and Sathe (2017)). Nevertheless, Fernandez-Delgado *et al.* (2014) showed empirically that although Rotation Forest (Rodriguez and Kuncheva, 2006) and Random Forest (Breiman, 2001) ensembles are high-performing classifiers when trained with full-class information, when these ensemble models are applied with the presence of novelties, they can result in a high misclassification rate and a poor novelty detection performance. One explanation for such a poor performance level lies in the

CONTACT Irad Ben-Gal bengal@tau.ac.il
Color versions of one or more of the figures in this article can be found online at www.tandfonline.com/uiie.

characteristics of the classification trees that form the ensemble forest. Classification trees partition the feature space into hyperplanes, generating areas with low data density levels (Breiman *et al.*, 1984) that become sparser as the dimensionality grows (Duda *et al.*, 2001). Moreover, another limitation worth noting is that most of the existing ensemble-based novelty detection approaches use a "single-class classification" paradigm and either treat multiple known classes as a single artificial class (Upadhyaya and Singh, 2012) or learn a single-class classification model for each class independently (Tax and Duin, 2008).

This article follows the above studies and presents a new *novelty detection* framework based on an ensemble of classifiers, each of which undergoes training using subsets of data attributes. The framework combines Bayesian inference SPC and accordingly is called Ensemble-Bayesian SPC (EB-SPC). In particular, we use Random Forest (Breiman, 2001) and Rotation Forest (Rodriguez and Kuncheva, 2006) as the ensemble methods to classify the data samples. Additionally, we incorporate heuristics combining mixture-of-experts (Jacobs *et al.*, 1991) and bootstrapped confidence intervals (Efron and Tibshirani, 1997) to define statistical limits similarly to the Hotelling T^2 monitoring method. The proposed EB-SPC overcomes the difficulty of computing the density estimations over several data classes that is often limited by the data dimensionality (Scott and Sain, 2005). Moreover, whereas conventional SPC-based novelty detection methods detect novelty instances by measuring a distance value of the test instance from the "normal" instances, this work presents a relatively simple, yet promising, heuristic that exploits the *joint* information simultaneously obtained by all of the classes to estimate the statistical boundaries derived from basic probabilistic concepts. We show that the proposed approach achieves a better performance in detecting new classes of data, in which the underlying assumption is that the classes, in general, are characterized by relatively compact densities. We show that the proposed combination of ensemble outputs boosts the conventional use of Rotation Forest and Random Forest classifiers to detect anomalies and elevates their usage for novelty detection in multi-mode process monitoring. Furthermore, the approach combines the probability outputs of the ensemble to generate class-dependent thresholds and then uses them to detect novelties. By doing so, the proposed EB-SPC extends the current scope of SPC charts for multi-class data environments beyond the known use of ensemble-based classifiers. Moreover, it extends classic mechanisms for ensemble of scores in novelty detection problems, such as voting or averaging (Aggarwal and Sathe, 2017), which make the decision in multi-class datasets more complex and challenging. Finally, the EB-SPC controls the statistical limits for each class, while estimating them by relying solely on the known classes, and thus extends the internal validation paradigms on novelty detection (Marques *et al.*, 2015). Experimental results on both real-world datasets as well as publicly available ones show that the EB-SPC method outperforms both state-of-the-art SPC approaches as well as ensemble-of-classifiers approaches in most of the analyzed cases.

The rest of this article is organized as follows. Section 2 introduces a general background on state-of-the-art methods for process monitoring and novelty detection related to the proposed EB-SPC. Section 3 presents the proposed monitoring framework in detail. It explains how the ensemble outputs are combined to generate a class-dependent threshold from the class samples. Section 4 provides a comparative study and experimental results based on standard repository datasets and on real-world data from a semiconductor fab. Section 5 summarizes the conclusions and outlines directions for future work.

2. Background and related work

Data-driven monitoring of complex systems has been studied in detail in recent decades (Montgomery, 2008). Approaches that aim at detecting abrupt changes in complex systems have been analyzed under the rubric of anomaly or novelty detection. In general, anomaly detection methods aim at detecting data observations that considerably deviate with respect to "regular" data samples (Aggarwal, 2015). Among anomaly detection techniques, one can find the classic Hotelling T^2 method and its derivatives (see, e.g., Chiang *et al.* (2001) and Kenett and Zacks (2014)), the Gaussian Mixture Models (Bishop, 2006), the Minimum Volume Sets (see, e.g., Park *et al.* (2010)), and some combinations of the above-mentioned approaches (e.g., Ge and Song (2013)). Other popular methods make use of Support Vector Machines (SVMs; Vapnik (1998)) such as OC-SVM (Schölkopf *et al.*, 2000), Support Vector Data Description (SVDD; Tax and Duin (2004)), or, again, other combinations derived from the above-mentioned approaches (Ge and Song, 2013). Well-documented surveys on anomaly detection can be found in the literature (Chandola *et al.*, 2007; Ben-Gal, 2010; Pimentel *et al.*, 2014). Most of the mentioned techniques address the challenge of monitoring a system as a "one-class" classification problem, where data samples are assumed to stem from a single and unknown probability distribution. This assumption constitutes a limitation for more complex processes where data samples from different distributions are better represented by a multi-class dataset.

Methods for monitoring systems that generate multi-class datasets are less common in the literature of classic SPC and are often based on pattern recognition, classification, and clustering algorithms coupled with statistical scoring schemes (see, e.g., Upadhyaya and Singh (2012)). Given a data sample gathered from the monitored process, the conventional approach is to classify the sample into one of the known classes that were learned during the training stage based on its statistical score (see Tax and Duin (2008)). If the score value (e.g., the likelihood) falls below some confidence threshold, the observed sample is labeled as an "abnormal" data sample. Several approaches have been proposed to address anomaly detection in multi-class datasets, such as the Bayesian Monitoring Statistic (BMS) based on Bayesian soft classification (Ge and Song, 2013). In the BMS method, classic statistics such as T^2 and Squared Prediction Error (SPE; see Chiang *et al.* (2001)) are obtained using the "in-control" data samples and then mapped into probability values for each data class. Then, a Bayesian combination of the probabilities is used to detect unexpected observations. Bodesheim *et al.* (2013) proposed a related approach that uses a classifier for novelty detection; the main idea was to map all of the training samples of each data into a single point in a null space kernel representation. In Bodesheim *et al.* (2015), an extension of their previous approach for a multi-class problem was presented, where multiple normal data classes and novelty detection are

jointly treated within a single model. Another approach that addressed the challenge of detecting abnormal samples in multi-class datasets was introduced by Jumutc and Suykens (2014) and named Multi-Class Supervised Novelty Detection (SND). In SND, the authors proposed an SVM-like algorithm that obtains decision functions for each class respectively while keeping the data description compact, while enhancing the probability of novelty detection, when outliers within the training dataset are present. Lazzaretti *et al.* (2016) proposed to use the SVDD, as originally presented by Tax (2001), to model each normal class seperately, while extending the objective function to include negative examples (i.e., observations that should be labeled as novelties). At the test stage, data samples that do not belong to any normal class are tagged as novelties by this approach.

The constant increase in the complexity of systems and processes requiring monitoring has pushed the frontiers of monitoring strategies to consider combinations of classic statistical approaches with machine learning techniques to cope with the sharp increase in the dimensionality and complexity of the generated data (Aldrich and Auret, 2013). To address these challenges, ensemble-of-classifiers have found extensive use in novelty detection applications, showing several advantages over methods that use a single model. First, ensemble-of-classifier methods often achieve a significant improvement in prediction accuracy (Kuncheva, 2004; Fernandez-Delgado *et al.*, 2014), especially when high-dimensional overlapping clusters are present in the data and the learned datasets are highly imbalanced (Byon *et al.*, 2010). Second, building an ensemble-of-classifiers is more efficient than building a single model, in particular when the ensembles are trained over different subsets of data attributes that are referred to as "subspaces" as in Random Forest (Breiman, 2001) or Rotation Forest (Rodriguez and Kuncheva, 2006). Lastly, the nature of the ensemble-of-classifiers allows its implementation to scalable and online classification tasks of large databases (Wang *et al.*, 2003), which has gained attention in modern process monitoring applications. Recently, Aggarwal and Sathe (2017) summarized relevant works, where the authors reviewed methods for "ensemble of subspaces" that addresses the problem of anomaly detection. Other ensemble-based approaches use an ensemble-of-classifiers trained over subspaces of data classification trees to identify novel observations in data streams, such as the Accuracy-Weighted Ensemble (AWE) proposed by Wang *et al.* (2003) and based on C4.5 classification tree (Quinlan, 1993). The AWE approach uses the classification error values from each ensemble element at the training stage to produce a class-dependent probability threshold to detect unexpected data samples in the monitoring stage. Similarly, Masud *et al.* (2009) proposed MineClass to detect abnormal data samples and identify novelties based on the *k*-Means (MacQueen, 1967) clustering algorithm. MineClass was later extended in Masud, Gao, Khan, Han, and Thuraisingham (2011) to incorporate time constraints at the training phase and update the ensembles for data stream applications. Masud, Gao, Khan, Han, and Thuraisingham, (2011) and Al-Khateeb *et al.* (2016) have also proposed a method to deal with recurring classes, a special case of novelty detection, where a class may appear and later disappear in the data stream. Recently, De Faria *et al.* (2016) proposed a new approach called MINAS to detect novel classes. MINAS uses an ensemble of micro-clusters trained over each data class and, similar to MineClass, detects novel observations if these exceed a threshold computed as the hyper-radius of the clusters.

Motivated by previous works, this article presents an ensemble-based SPC that combines a Bayesian inference approach for novelty detection and it is named EB-SPC. The proposed framework learns the data dependencies in the monitored process and the underlying class distributions without relying on prior assumptions about the data distribution. It trains ensembles of classifiers over the known data classes generated from the monitored process. It applies both Rotation Forest and Random Forest ensembles and uses their outputs to generate Bayesian posterior thresholds to signal novelties. First, the ensemble is used to compute a conditional probability distribution for each class and to estimate a low control threshold for each class. Then the EB-SPC combines the low-density regions over all of the trained classes based on their posterior conditional probabilities. A consequence of this approach is that if low posterior class probabilities are computed from each ensemble, their combination by the EB-SPC generates an *overall* low posterior probability threshold for that class, resulting in an improved novelty detection performance. Many of the existing ensemble-based approaches for novelty detection use voting scores or classification errors as their weighting factors in a linear combination function (e.g., AWE (Wang *et al.*, 2003), MineClass (Masud *et al.*, 2009), and T-Norm (Tax and Duin, 2008)). However, the proposed EB-SPC combines the ensemble outputs over all classes based on a joint posterior probability of the class. In other words, a main contribution of the proposed approach is not in the use of an ensemble-based approach for novelty detection but, rather, the use of an SPC scheme to combine the outputs of various ensembles to detect novelties that emerge during the operating stage of the process and do not match the previously learned classes. Finally, let us note that although the proposed scheme is independent from the applied ensemble, the EB-SPC makes use of the intrinsic characteristics of the Classification and Regression Tree (CART) approach, originally proposed by Breiman *et al.* (1984), as the base classifier in the ensemble. CART partitions the feature space to refined hyperplanes during the training phase. Such partitioning is appealing since a new data sample is classified by the corresponding CART leaf; i.e., in a feature space hyperplane, independent of the density of the class in that region. The next section details the proposed approach.

3. EB-SPC

3.1. Problem definition and real-world example

Consider a monitored system or process whose in-control data can be used in learning processes. Let p be the number of attributes or features (we will use these two terms interchangeably) in the data, each of which is represented by a random variable X_i, $i = 1, 2, \ldots, p$. Note that X_i can be a categorical, discrete, or continuous random variable; some of them might be fixed as inputs to the process while others might belong to the process outputs. Denote the random vector of p features by $\boldsymbol{X} = \{X_1, X_2, \ldots, X_p\}$, where a sample point (realization) of all features is denoted by lowercase letters $\boldsymbol{x} = \{x_1, x_2, \ldots, x_p\}$

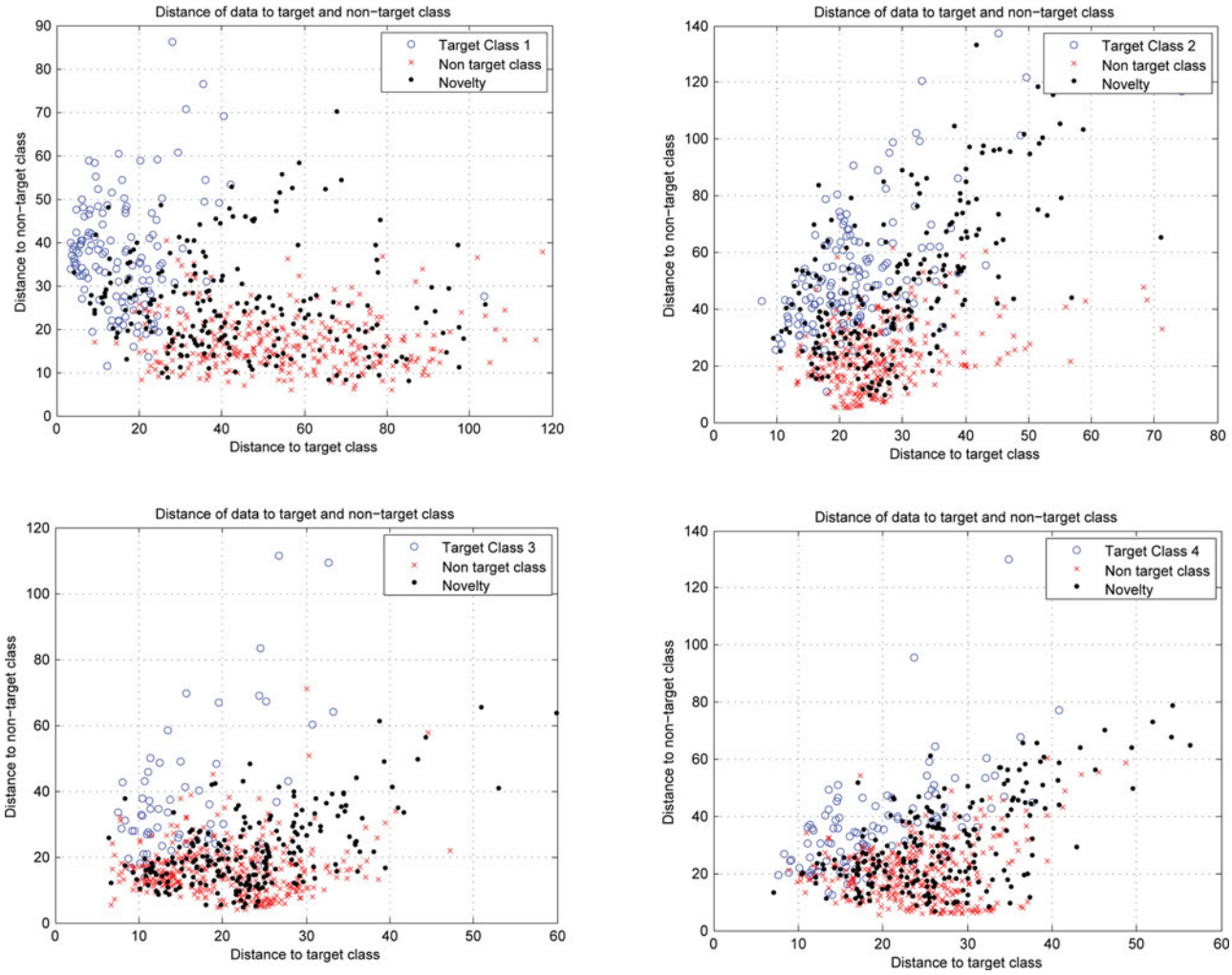

Figure 1. Mahalanobis distances between the target "in-control" class, the rest of the classes, and the novelties, using real-world data from the semiconductor industry. The *x*-axis measures the distance between target data samples to itself. The *y*-axis shows the distance between the target data samples to the other "non-target" classes. "o"-signs represent target class samples, "x"-signs represent non-target class data samples, and "."-signs represent novelty observations.

and for the continuous case $\boldsymbol{x} \in \mathbb{R}^p$. Let the class variable be represented by a discrete random variable Y with values taken from an unordered set of class labels $\mathcal{Y} = \{y_1, y_2, \ldots, y_C\}$, where C is the number of different data classes available for learning. The probability that Y obtains the value $y_c \in \mathcal{Y}$ is denoted by $\Pr(Y = y_c) \equiv p(y_c)$, and $\sum_{c=1}^{C} p(y_c) = 1$ is assumed during the learning stage. Let $D = \{\langle \boldsymbol{x}^{(i)}, y^{(i)} \rangle\}_{i=1}^{N}$ represent the training dataset records, where each set of feature values $\boldsymbol{x}^{(i)}$ is tagged by one of the data classes $y^{(i)} \in \mathcal{Y}$, for $i = 1, 2, \ldots, N$, while N denotes the number of data records. As the gathered dataset represents a multi-mode system, overlapping clusters might exist over the features space tagged with a specific class distribution (see, for example, Fig. 1). Assume that each class (associated potentially with one or more modes of operation) represents a stationary process, in which $p(\boldsymbol{x}|y_c)$ denotes the conditional probability distribution of the data $\boldsymbol{x}$ given the class label y_c. Thus, by applying the Bayes rule and dropping the instance index for simplicity of exposition, one can compute the posterior class probability $p(y_c|\boldsymbol{x}) = p(\boldsymbol{x}|y_c)p(y_c)/p(\boldsymbol{x})$. Furthermore, given the prior class probability, $p(y_c)$, one can compute a class-dependent threshold φ_c^* such that if $p(y_c|\boldsymbol{x}) \leq \varphi_c^*$, with $c \in \{1, 2, \ldots, C\}$ denoting the class index, the data sample $\boldsymbol{x}$ can be labeled as abnormal with respect to that class y_c, as suggested by Fumera *et al.* (2000). In classic approaches, the prior class probability can be estimated, for example, by multivariate Kernel Density Estimation (Scott and Sain, 2005). If the following conditions are met—(i) the data dimensionality is relative low; (ii) there is a considerable number of data samples; (iii) the classes are well separated; and (iv) there are no outliers within the dataset—one can estimate a single threshold value and use it as a rejection threshold (Upadhyaya and Singh, 2012). These conditions, however, are not guaranteed to occur in real-life settings.

As a real-world example, consider a real multi-mode process from the semiconductor industry. Semiconductor lines contain dynamic multi-mode processes and thus represent a relevant monitoring environment. The technology producing integrated circuits is continually changing in order to respond to the high demand for new and faster products with improved quality requirements. The high integration level of functions in dies requires sophisticated monitoring approaches and novel metrology tools (Diebold, 2001). In general, metrology tools measure hundreds of physical characteristics of the dies, generating a huge amount of multivariate data (Diebold, 2001) used for the monitoring of the production quality in semiconductor fabs. Moreover, because wafers are processed over different layers with different tools along the production line, new errors that remain unknown to metrology devices can emerge, due

to degradation of production materials or errors caused by metrology instruments that are uncalibrated. The collected data consist of real-world processed images of dies of different wafer surfaces along the production line. The processed images are labeled with a code identifying the error type on the image. From each image, multivariate data points with mixed data types (as well as missing values) are generated; thus, the collected data represent a multi-class.

Figure 1 shows the metrology dataset from a semiconductor fab that addresses a scenario of novelty detection that might emerge as a result of new combinations of affecting factors, such as new designs, new materials, and changing environmental conditions. To account for novelties, we select four major classes to represent the "in-control" datasets, whereas other smaller classes are considered "novelties" that are unavailable for training during the learning stage. Each of the plots in Figure 1 refers to one of the learned "in-control" classes (we call each of them the "target class" when analyzing it) and shows the Mahalanobis distance between both this target class versus itself and versus the rest of the learned classes (named "non-target classes" at this stage). Such plots map the data into a lower manifold using a "one-against-all" approach superposed with novelties; i.e., the rest of the classes that are not one of the (learned) target classes. The x-axis measures the distance between target data samples to the target class. The y-axis shows the distance between the target data samples to the other "non-target" classes. The "o" signs represent the data points from the target class; "×" signs represent data points from the other learned classes. However, the "·" signs represent novelties that should be detected at later monitoring stages. It is not surprising that circles have smaller values on the x-axis than on the y-axis, as they represent the distance between the target class to itself. Figure 1 also shows that in most cases, the target classes are relatively separable from the other learned classes.

Note that the novelties are spread out among the target and non-target classes but are not well separated from the "in-control" classes in this case, creating overlapping clusters that are hard to separate. From a monitoring perspective, it is interesting to see whether some multi-variate state-of-the-art SPC methods can distinguish clearly between the different classes in the data, including novelties that were not available for training.

As shown in this simple example, in real-world monitoring applications, the training datasets might include outliers, missing values, and mixed data types that can significantly affect the SPC results if the monitoring method is not sufficiently robust. Overlap of data clusters cannot be discarded *a priori*, and a significant amount of high-dimensional data points might be unavailable for training. This last point is critical to reduce the "curse-of-dimensionality" effects when density estimation is involved (Duda *et al.*, 2001). Dimensionality reduction techniques, such as PCA or Independent Component Analysis (ICA), can help reduce these dimensionality effects; however, the identification of anomalies and novelties can be a very challenging task, as new classes might emerge over subsets of attributes, and overlapping of data clusters (that might increase in low dimensions) is expected in such processes (Aggarwal, 2017). Finally, an additional difficulty arises when using thresholding over the posterior class probabilities $p(y_c|\boldsymbol{x})$ to identify novelties if the prior class probabilities $p(y_c)$ are dependent upon each other. Exact values of thresholds can be very hard to estimate when insufficient data have been collected or new classes of data are expected. Tax and Duin (2008) addressed some of these challenges by proposing a *manual intervention* for setting the correct thresholds to discard new data samples or to label them as anomalies. This article addresses such challenges and aims at finding a practical method for novelty detection in which thresholds can be defined via a systematic and automated procedure without relying on external expert intervention. The proposed approach is described in the next subsection.

3.2. Proposed framework

Let $\mathrm{H}(D, T, \mathbf{S}_k)$ denote an ensemble of T classifiers, not necessarily identical, trained over the multi-class dataset D, in which each classifier is trained over a set of k randomly selected features, $\mathbf{S}_{k,j}$, which defines the training subspace for the jth classifier. Thus, if $\boldsymbol{X} = \{X_1, X_2, \ldots, X_p\}$ is the set of all features, then $\mathbf{S}_k = \{\mathbf{S}_{k,j}\}_{j=1}^{T}$, $|\mathbf{S}_k| = kT$ features, where each $\mathbf{S}_{k,j} \subseteq \mathbf{X}$, $|\mathbf{S}_{k,j}| = k \leq p$ and $|\mathbf{S}_{k,j} \cap^{\mathbf{S}_{k,u}}| \geq 0, \ \forall j \neq u$, and k is the subspace-size parameter. Based on the multi-class characteristics of the dataset, an ensemble for each data class y_c, $c = 1, 2, \ldots, C$, is constructed using a "one-against-all" training approach (see, e.g., Duda *et al.* (2001)). That is, for each data class y_c, an ensemble of classifiers is trained using all data samples that are labeled with $y^{(i)} = y_c$ (i.e., the target class), where the instances from the remaining classes, $y^{(i)} = y_j$, $j \in \{1, 2, \ldots, C\}$, $j \neq c$, for $i = 1, 2, \ldots, N$, are joined together as a single data class and denoted by $\bar{y}_c$. The resulting ensemble that is trained with respect to a specific target class y_c is denoted, accordingly, by $\mathrm{H}(D, T, \mathbf{S}_k; y_c)$. The justification to follow this approach relies on the objective that for each target class y_c one aims at learning the boundaries of its data distribution with respect to all the other known classes (see, e.g., Duda *et al.* (2001)). Thus, instead of using a resampling procedure based on a uniform distribution, as commonly used in "one-class anomaly detection" approaches (see, e.g., Davenport *et al.* (2006)), we exploit the available information given by all known classes at the training stage. Later, we show by experimentation that such an approach results in an improved performance for detecting abnormal data samples and particularly for detecting novelties that are different from all known classes used for the training. Once the ensembles have been trained with respect to each target class y_c, tested (new) data samples can be classified by applying the well-known Bayesian minimal loss function (see, e.g., Duda *et al.* (2001)). Specifically, given a new observation $\boldsymbol{x}^{(t)}$, the class label is obtained as $\hat{y}^{(t)} = \mathrm{argmax}_{\{y_c \in \mathcal{Y}\}} \hat{p}_{en}(y_c|\boldsymbol{x}^{(t)})$, where $\hat{p}_{en}(y_c|\boldsymbol{x}^{(t)})$ denotes the ensemble-based posterior class conditional probability retrieved by the ensemble $\mathrm{H}(D, T, \mathbf{S}_k; y_c)$, with prior class probabilities that are either estimated from previous observations or rely on a uniform distribution over the classes. However, the classification of $\boldsymbol{x}^{(t)}$ might be reasonable in the presence of balanced data classes and the absence of abnormal observations (Tax and Duin, 2008). To overcome these limitations, we propose a heuristic that is derived from concepts of "mixture-of-experts" (Jacobs *et al.*, 1991) to combine the ensemble outputs. For the class index $c \in \{1, 2, \ldots, C\}$:

$$\tilde{p}\left(\hat{y}_c^{(t)}|\boldsymbol{x}^{(t)}\right) = \frac{\sigma\left(\hat{p}_{en}\left(y_c|\boldsymbol{x}^{(t)}\right); \delta_c\right) \hat{p}_{en}\left(y_c|\boldsymbol{x}^{(t)}\right)}{\sum_{j=1}^{C} \sigma\left(\hat{p}_{en}\left(y_j|\boldsymbol{x}^{(t)}\right); \delta_j\right) \hat{p}_{en}\left(y_j|\boldsymbol{x}^{(t)}\right)}, \quad (1)$$

where $\sigma(\hat{p}_{en}(y_c|\boldsymbol{x}^{(t)}); \delta_c)$ is the sigmoid function governed by

the class-dependent parameter δ_c in a form of logistic regression, as suggested in Jacobs *et al.* (1991), and $\tilde{p}(\hat{y}_c^{(t)}|\boldsymbol{x}^{(t)})$ is thus the posterior conditional probability of class $\hat{y}_c^{(t)}$. The class label of $\boldsymbol{x}^{(t)}$ is obtained as $\hat{y}_c^{(t)} = \text{argmax}_{\hat{y}_c^{(t)} \in \mathcal{Y}}\{\tilde{p}(\hat{y}_c^{(t)}|\boldsymbol{x}^{(t)})\}$. The sigmoid function is defined as $\sigma(\hat{p}_{en}(y_c|\boldsymbol{x}^{(t)}); \delta_c) := 1/1 + \exp\{-\beta(\hat{p}_{en}(y_c|\boldsymbol{x}^{(t)}) - \delta_c)\}$, where β and δ_c are convenient tuning parameters (Jacobs *et al.*, 1991). Adding the offset δ_c provides a better control on estimating the threshold value to detect novelties. The sigmoid function performs the gating with the posterior class conditional probability differently from the classic approach of mixture-of-experts, in which the data sample $\boldsymbol{x}^{(t)}$ is instead used to compute the gating probability (Jacobs *et al.*, 1991). By following this procedure, we avoid parameterizing the sigmoid function for potential high-dimensional datasets, since $\hat{p}_{en}(y_c|\boldsymbol{x}^{(t)})$ is a scalar value.

To compute the thresholds φ_c for each class, we use the conventional F1-Score measure defined as $2PR/(P+R)$, where P and R are the Precision and the Recall performance measures, respectively. The precision is computed by the ratio $P = TP/(TP+FP)$, where TP is the number of true positive outcomes, and FP is the number of false positives. The Recall is computed by the ratio $R = TP/(TP+FN)$, where FN is the number of false negatives. For a given target class y_c, the threshold φ_c^* is defined such that the F1-Score is as close as possible to one. We reconfigure the standard contingency matrix for binary classifiers, as shown in Table 1. In the table, y_c represents the real target class, whereas $\hat{y}_c$ represents the estimated class, as obtained by the ensemble when aiming at maximizing the posterior class probability. The posterior class probability $\tilde{p}(\hat{y}_c^{(t)}|\boldsymbol{x}^{(t)})$ is then obtained by using Equation (1). The threshold φ_c^* is computed by searching for the values of the parameters T and k for the ensembles $H(D, T, \mathbf{S}_k; y_c)$, as well as δ_c and β for the sigmoid function in Equation (1) that maximize the F1-Score derived from Table 1. Recall that in order to estimate the class-dependent thresholds, φ_c^*, by means of Table 1, "in-control" data observations are used. One drawback that occurs when maximizing the F1-Score for computing the threshold φ_c^* is that it requires all data samples of the target class y_c to be correctly classified as "normal" in-control samples. A possible consequence of this maximization procedure is an over-fitting process, leading to a small value of the false-positive rate at the expense of a much higher value of the false-negative rate. To cope with such a limitation, each posterior class probability $\tilde{p}(\hat{y}_c^{(t)}|\boldsymbol{x}^{(t)})$ for each ensemble $\text{H}(D, T, \mathbf{S}_k; y_c)$ is represented by a random variable. A probability mass distribution and a confidence interval for each posterior class probability, derived over the available dataset D, is then computed implementing a bootstrap t-intervals approach (Efron and Tibshirani, 1997). The lower value of the confidence interval is selected as a threshold

Table 1. Proposed configuration of the contingency matrix.

		True state	
		y_c	$\bar{y}_c$
Test outcome	$\hat{y}_c$	$\tilde{p}(\hat{y}_c\|\boldsymbol{x}) > \varphi_c$ (TP)	$\tilde{p}(\hat{y}_c\|\boldsymbol{x}) > \varphi_c$ (FP)
	Non $\hat{y}_c$	$\tilde{p}(\hat{y}_c\|\boldsymbol{x}) \leq \varphi_c$ (FN)	$\tilde{p}(\hat{y}_c\|\boldsymbol{x}) \leq \varphi_c$ (TN)

Framework of EB-SPC Algorithm

For $T = T_1, T_2 \ldots T_{M_T}$
 For $k = k_1, k_2 \ldots k_{M_K}$
  Train $\text{H}(D_T, T, \mathbf{S}_k; y_c)$ with D_T
  Evaluate $\text{H}(D_V, T, \mathbf{S}_k; y_c)$ on D_V
  Compute F1-Scores: $F_{\mu}^{Class}(T,k)$ and $F_{\text{M}}^{Class}(T,k)$
  Compute
$F^{Class}(T,k) = \sqrt{F_{\mu}^{Class}(T,k) F_{\text{M}}^{Class}(T,k)}$
  For $c = 1,2 \ldots C$
   For $\delta_c \in [\delta_{min}; \delta_{max}]$
    Compute Eq. (1)
    Compute
$\varphi_c = E\{\tilde{p}(\hat{y}_c|\boldsymbol{x})\} - t_{\alpha_c} Std(\tilde{p}(\hat{y}_c|\boldsymbol{x}))$
    (α_c can be set to 95$^{\text{th}}$ percentile or searched greedily)
    Compute Novelty F1-Score: $F^{Novelty}(T, k, \{\alpha_c, \delta_c\}, \beta)$
    Compute $F_c(\cdot) = \sqrt{F^{Novelty}(\cdot) F^{Class}(T,k)}$
    where $\boldsymbol{x} \in D_V$
   End
  End
 End
End
Select $\text{H}(D, T^*, \mathbf{S}_k^*; y_c)$ and $\varphi_c^* = argmax_{\varphi_c}\{F_c(\cdot)\}$

Figure 2. Pseudo-code of the parameterization algorithm.

by which $\varphi_c = \text{E}\{\tilde{p}(\hat{y}_c|\boldsymbol{x})\} - t_{\alpha_c}\text{Std}(\tilde{p}(\hat{y}_c|\boldsymbol{x}))$, where t_{α_c} denotes the $1-\alpha_c$ critical value of the bootstrapped t-distribution for class y_c, and $\text{E}(\cdot)$ and $\text{Std}(\cdot)$ denote the expected value and the standard deviation of the posterior class probability distribution, respectively. That is, the interval term is obtained by the bootstrapped t-distribution taking the $1-\alpha_c$ percentile according to industrial and academic standards (which is often based on the 95th or the 99th percentiles). To obtain the t-distribution, we use a Kernel Density Estimation via a diffusion algorithm, as described in Botev *et al.* (2010). This algorithm automatically estimates the best bandwidth using Gaussian kernels.

Figure 2 shows the proposed pseudo-code for training the ensemble of classifiers and for combining their outputs and computing the class-dependent threshold φ_c. First, the dataset D is randomly split into training and validation sets, D_T and D_V, respectively. Then, the ensembles $\text{H}(\text{D}_T, T, \mathbf{S}_k; y_c)$ are trained for each class using only the training split, D_T, and evaluating the performance of the ensemble using the validation split, D_V. Recall that T and $\mathbf{S}_k$, respectively, represent the ensemble size and the set of subspaces, each of which are comprised of k randomly selected attributes. The ensemble, which contains T classifiers trained over k randomly selected attributes is denoted by $\text{H}(D_T, T, \mathbf{S}_k; y_c)$. Random selection of subspaces allows ensemble classification to reduce overfitting and local minima effects and is the main reason for implementing random selection ensembles in the proposed

method (Kuncheva, 2004). The pseudo-code evaluates M_T different ensemble sizes and M_K different subspace sizes; i.e., $k_j j = 1, 2, \ldots, M_K$ and $T_i i = 1, 2, \ldots, M_T$ combinations are evaluated. Since the dataset D represents a multi-class system, we compute the micro and macro F1-Scores as proposed in Özgür *et al.* (2005) which are denoted in Figure 2 as $F_{\mu}^{Class}(T, k)$ and $F_{\mathrm{M}}^{Class}(T, k)$, respectively. Specifically, in Özgür *et al.* (2005), the micro F1-Score is defined as $F_{\mu} = 2\pi\varrho/\pi + \varrho$, where $\pi = \sum_{c=1}^{C} TP_c/(TP_c + FP_c)$ and $\varrho = \sum_{c=1}^{C} TP_c/(TP_c + FN_c)$. TP_c denotes the number of true positive outcomes, FP_c denotes the number of false positive outcomes, FN_c denotes the corresponding number of false positive outcomes for data class $y_c \in \mathcal{Y}$, and C denotes the total number of data classes. Similarly, the macro F1-Score is defined as $F_M = \sum_{c=1}^{C} F_c/C$, where F_c is the F1-Score for class y_c previously defined. The upper index "*Class*" implies that the performance measure is relative to the classification result of the ensembles. We combine both performance values with the geometric mean, represented as $F^{Class}(T, k)$. Once the ensembles are trained for each class, Equation (1) is used to compute the posterior class probability using the validation dataset. Table 1 is then used to compute the F1-Score and is thereafter geometrically combined with the $F^{Class}(T, k)$ values. The final ensemble of classifiers $H(D,\ T^*,\ \mathbf{S}_k^*; y_c)$ and the threshold values φ_c^* are then calculated so that the obtained F1-Score, denoted in Figure 2 by $F_c(\cdot)$, is maximized.

Note that in Figure 2, $F_c(\cdot)$ is a function of the parameters T, k, $\{\alpha_c, \delta_c\}$, and β. Additionally, note that $F^{Novelty}(\cdot)$ is used to measure the performance among "in-control" data samples. Finally, recall that only normal "in-control" data observations are used to train the proposed framework.

During the monitoring phase, for a given new data sample $\boldsymbol{x}^{(t)}$, we compute the posterior probability $\tilde{p}(\hat{y}_c^{(t)}|\boldsymbol{x}^{(t)})$ for each target class y_c using Equation (1) where $\hat{p}_{en}(y_c|\boldsymbol{x}^{(t)})$ is derived by the ensemble $H(\boldsymbol{x}^{(t)}, T^*, \mathbf{S}_k^*; y_c)$. In particular, a class label y_c is assigned to the data sample $\boldsymbol{x}^{(t)}$ if and only if

$$\tilde{p}\left(\hat{y}_c^{(t)}|\boldsymbol{x}^{(t)}\right) > \tilde{p}\left(\hat{y}_j^{(t)}|\boldsymbol{x}^{(t)}\right) \ \forall j \in \{1, 2, \ldots, C\},$$
$$j \neq c, \text{ and } \tilde{p}\left(\hat{y}_c^{(t)}|\boldsymbol{x}^{(t)}\right) > \varphi_c^*.$$

Otherwise, if $\tilde{p}(\hat{y}_c^{(t)}|\boldsymbol{x}^{(t)}) \leq \varphi_c^*$, then the data sample $\boldsymbol{x}^{(t)}$ is labeled as being "abnormal" with respect to that target class. Let us define D_Q as the set containing all data samples classified as abnormal. Furthermore, the term $Sim(D_Q, \mathbf{S}_k^*)$ represents the algorithmic procedure that measures the *similarity* among the abnormal samples to each other in D_Q using the set of subspaces $\mathbf{S}_k^*$. Thus, if more than Q abnormal samples exist in the set D_Q, such that $\tilde{p}(\hat{y}_c^{(t)}|\boldsymbol{x}^{(t)}) \leq \varphi_c^*\ \forall \hat{y}_c^{(t)} \in \mathcal{Y}$, $Sim(D_Q, \mathbf{S}_k^*) < \lambda_c$, where λ_c is a class-dependent threshold, we define a "novelty" class in the system. The threshold value λ_c is computed by the following heuristic. Specifically, we use the selected subspaces during the training of the ensembles—i.e., $\mathbf{S}_k^* = \{\mathbf{S}_{k*,i}\}_{i=1}^{T^*}$—for each class y_c, each of which contains k^* randomly selected features. Assuming that in class y_c there are N_c data samples, we compute the average Euclidean distance for each data sample $\boldsymbol{x}^{(j)}$, $j = 1, 2, \ldots, N_c$, using its G nearest observations in each subspace S_i; i.e., $dist(\boldsymbol{x}^{(j)}, \mathbf{S}_{k^*,i}) = (1/G)\sum_{u=1}^{G} \boldsymbol{x}^{(j)}(\mathbf{S}_{k^*,i}) - \boldsymbol{x}^{(u)}(\mathbf{S}_{k^*,i})_2$, $i = 1, 2, \ldots, T^*$, where $\boldsymbol{x}^{(u)}(\mathbf{S}_{k^*,i})$ representes one of the G closest data samples to $\boldsymbol{x}^{(j)}(\mathbf{S}_{k^*,i})$ in the subspace $\mathbf{S}_{k^*,i}$. We then arrange these distances in a matrix $\boldsymbol{\Omega} \in \mathbb{R}^{T^* \times N_c}$, in which each column corresponds to the $dist(\boldsymbol{x}^{(j)}, \mathbf{S}_{k^*,i})$ for $i = 1, 2, \ldots, T^*$, and each row corresponds to $dist(\boldsymbol{x}^{(j)}, \mathbf{S}_{k^*,i})$, $j = 1, 2, \ldots, N_c$. We then compute the average distance of each row to obtain a set of T^* values—i.e., $(1/N_c)\sum_{j=1}^{N_c} dist(\boldsymbol{x}^{(j)}, \mathbf{S}_{k^*,i})$—that are arranged in the vector $\bar{\boldsymbol{\Omega}} \in \mathbb{R}^{T^*}$. In the following we illustrate the matrix $\boldsymbol{\Omega}$ and the resulting array $\bar{\boldsymbol{\Omega}}$:

$$\boldsymbol{\Omega} = \begin{bmatrix} dist(\boldsymbol{x}^{(1)}, \mathbf{S}_{k^*,1}) & \cdots & dist(\boldsymbol{x}^{(N_c)}, \mathbf{S}_{k^*,1}) \\ \vdots & \ddots & \vdots \\ dist(\boldsymbol{x}^{(1)}, \mathbf{S}_{k^*,T^*}) & \cdots & dist(\boldsymbol{x}^{(N_c)}, \mathbf{S}_{k^*,T^*}) \end{bmatrix} \text{ and}$$

$$\bar{\boldsymbol{\Omega}} = \begin{bmatrix} 1/N_c \sum_{j=1}^{N_c} dist(\boldsymbol{x}^{(j)}, \mathbf{S}_{k^*,1}) \\ \vdots \\ 1/N_c \sum_{j=1}^{N_c} dist(\boldsymbol{x}^{(j)}, \mathbf{S}_{k^*,T^*}) \end{bmatrix}.$$

Finally, we select $\lambda_c = max\{\bar{\boldsymbol{\Omega}}\}$. We can also select a specific percentile within the distribution of averaged distances in $\bar{\boldsymbol{\Omega}}$ to ensure a specific level of compactness in the novel class, which nevertheless is defined specifically to the used application. To summarize the above, novelties are identified by the following rule: Given Q abnormal observations in the set D_Q, the distance measures $dist(\boldsymbol{x}^{(q)}, \mathbf{S}_{k^*,i})\forall \boldsymbol{x}^{(q)} \in D_Q$ by using the G nearest data samples to $\boldsymbol{x}^{(q)}$ in each subspace $\mathbf{S}_{k^*,i}$ are computed, while arranging these values in the matrix $\boldsymbol{\Omega}_Q \in \mathbb{R}^{T^* \times Q}$. Following, the row-average values of $\boldsymbol{\Omega}_Q$ are computed and denoted by $\bar{\boldsymbol{\Omega}}_Q$. A novel class is then defined if at least Q entries in $\bar{\boldsymbol{\Omega}}_Q$ are less than λ_c.

Figure 3(a) (training) and Figure 3(b) (monitoring) show the pseudo-code for the proposed novelty detection; note that in Figure 3(b) we denote $\mathbb{I}(\cdot)$ as the indicator function. Recall that before an observation is identified as an abnormal data sample, it is classified with a trained class label and therefore we used λ_c for each abnormal data sample $\boldsymbol{x}^{(q)}$, assuming that the class label is given.

4. Experimental results

Several experiments were executed to benchmark the proposed EB-SPC framework against both state-of-the-art Multivariate SPC methods as well as against ensemble-of-classifiers approaches. The dual comparison has been used to illustrate that the contribution of the proposed approach lies on the integration between an ensemble approach and the new SPC paradigm.

Novelty Detection Algorithms

For each class y_c in D_T:
  For $j = 1,2 \ldots N_c$
    For $i = 1,2 \ldots T^*$
      $\boldsymbol{\Omega}(i,j) = dist(\boldsymbol{x}^{(j)}, \mathbf{S}_{k^*,i})$
    End
  End
  Compute $\bar{\boldsymbol{\Omega}}(i) = {}^{1}/_{N_c} \sum_{j=1}^{N_c} \boldsymbol{\Omega}(i,j)$ for $i = 1,2 \ldots T^*$
  $\lambda_c = \max\{\bar{\boldsymbol{\Omega}}\}$
End

(a) Training

For each abnormal observation $\boldsymbol{x}_q \in D_Q$:
  For $i = 1,2 \ldots T^*$
    $\boldsymbol{\Omega}_Q(i,q) = dist(\boldsymbol{x}^{(q)}, \mathbf{S}_{k^*,i})$
  End
  Compute $\bar{\boldsymbol{\Omega}}_Q(i) = {}^{1}/_{Q} \sum_{j=1}^{Q} \boldsymbol{\Omega}_Q(i,q)$, for $i = 1,2 \ldots T^*$
  If $\sum_{i=1}^{T^*} \mathbb{I}(\bar{\boldsymbol{\Omega}}_Q(i) < \lambda_c) \geq Q$ then declare novel class

(b) Monitoring

Figure 3. Pseudo-code of the novelty detection algorithm for (a) the training and (b) the monitoring stages.

Table 2. Characteristics of 20 UCI public datasets used in this study.

Dataset	*Classes*	*Instances*	*Dimensionality*
Arrhythmia	16	452	279
Audiology[(1)]	9	226	69
Dermatology	6	366	33
Glass	6	210	10
Pen Digits	10	5620	64
Segmentation	6	9900	50
Features - Fourier[(2)]	7	860	74
Features - Karhunen-Loeve[(2)]	7	860	64
Features - Pixels[(2)]	7	860	240
Letters[(3)]	13	19 255	16
Splice[(4)]	3	3190	60
Satimage	6	4204	36
Zoo[(1)]	7	101	7
Wine[(1)]	3	178	14
Waveform[(4)]	3	5000	22
Waveform2[(4)]	3	5000	40
Faults[(2)]	7	1941	27
Isolet[(2)]	26	6238	617
Thyroid	5	3770	29
Covertype[(2)]	7	581 012	54

Notes.
[(1)] Four classes were selected as normal modes, and nine classes were defined as novelties.
[(2)] Four classes were selected as normal modes, and three classes were defined as novelties.
[(3)] Seven classes were selected as normal modes and six classes as novelties.
[(4)] Two classes were selected as normal modes and one class as novelties.

In these benchmark studies, we used both public datasets from the UCI repository (Bache and Lichman, 2013) as well as real-world datasets gathered from a multi-mode semiconductor process.

4.1. Used data

Representative datasets were selected from the UCI repository to compare the benchmarked algorithms. Table 2 shows the characteristics of 20 multi-class datasets used in the experimentation. In addition, we tested the EB-SPC using real-world data gathered from an industrial line of a world-leading semiconductor firm, as shown in Figure 1. Table 3 shows the characteristics of the real-world dataset that was used in this study.

In all tested datasets, the experiment was defined as follows. First, the dataset instances were labeled, indicating those that belonged to "normal" ("in control") classes versus those that belonged to the "novelty" classes. The normal classes were defined as those that contained the majority of the instances (i.e., the novelty classes contained the minority of instances). In balanced datasets, in which the amount of data in each class was nearly equal, we randomly tagged some classes as the "normal" ones, whereas the rest of the classes were tagged as "novelty." The data points in the normal classes were split into three subsets: one for training (approximately 70% of the data), one for validation (approximately 20% of the data), and one for testing (approximately 10% of the data). Missing data were imputed by estimating the data distribution in each class using Gaussian Mixture Models (see, e.g., Bishop (2006)).

Table 3. Characteristics of the four real-world semiconductor datasets used in this study.

Dataset	*Classes*	*Instances*	*Dimensionality*
Dataset #1[(1)]	25	1013	128
Dataset #2[(2)]	16	2876	104
Dataset #3[(3)]	17	2091	47
Dataset #4[(4)]	17	5011	40

Notes.
[(1)] Six classes were selected as normal modes (80% of the data).
[(2)] Five classes were selected as normal modes (92% of the data).
[(3)] Three classes were selected as normal modes (92% of the data).
[(4)] Four classes were selected as normal modes (90% of the data.)

4.2. Parameters and experimental setting

As mentioned above, Rotation Forest and Random Forest were used as the ensemble of classifiers in the proposed EB-SPC. A clear motivation for using a Random Forest model is that it has been shown to obtain a higher performance than most of the other popular classifiers (Fernandez-Delgado *et al.*, 2014). Rotation Forest applies both random selections of subspaces and feature extraction by rotating the subspaces into their principal components, thus improving diversity of the final ensemble. In our implementation of Rotation Forest, we did not eliminate subsets of classes as we followed the one-against-all learning strategy.

The CART approach was selected as the classifier of the forest in both ensembles (Breiman *et al.*, 1984). CARTs are known as efficient models when simultaneously handling different data categories and are relatively easy models to implement. Recall from Section 3 that the ensemble requires two parameters: the number of classifiers, denoted by T and the subspace size (the number of randomly selected attributes), denoted by k. Often the values of both parameters can be determined by means of cross-validation techniques, as proposed in the framework shown in Figure 2. In practice, we initially selected the ensemble size T and then proceeded with the experimentation and searched for a proper subspace size. Once the parameters' values were found, the ensemble of classifiers remained the same for a given dataset D through all of the experiments and the comparisons against other ensemble-based benchmark methods to ensure a fair performance comparison among the results.

The EB-SPC ensemble size T was searched using a greedy approach within a linear-spaced grid of 50 values in the range of $[10,\ 500]$ CART classifiers. Throughout the experiments, following an exhaustive empirical experimentation, it was found that an EB-SPC ensemble size of 110 CART trees resulted in a relatively high and robust classification accuracy. Following Aggarwal and Sathe (2017), the subspace sizes were selected from a pool of five candidate values, each of which was a multiple of $\sqrt{p}$, where p represents the data dimensionality. The percentile (significance) parameter α was selected to follow the 95th percentile, based on industrial (and academic) standards. Note, however, that the value of α can be tuned by means of a grid search over the interval—e.g., $[0.01,\ 0.99]$—to obtain the highest $F_c(\cdot)$ score over the validation dataset during estimation of thresholds (as indicated in Fig. 2). The δ-parameter was empirically set using a grid search within the range $[0.3,\ 0.99]$ so that the overall F1-Score was maximized, as indicated in Figure 2. The β-parameter in the sigmoid function was empirically selected after several grid search trials and was fixed to 50 throughout the experiments. We followed the proposed procedure in Efron and Tibshirani (1997) and selected a sample

size of 2000 samples in the bootstrap step to compute the class-dependent thresholds.

For performance measurement, we used the F1-Score in all experiments, as it measures the rate between correct detection and false detection or misdetection of novelties. The F1-Score can also be defined as 2TP/(2TP + FP + FN), where the true positive value (i.e., TP) was defined by the number of novelties being correctly detected and the false negative (i.e., FN) and false positive (i.e., FP) values were defined as the number of missed novelties and the number of normal data samples wrongly labeled as novelties, respectively. The evaluation of each method was repeated 20 times for each dataset, such that in each replication different samples for training–validation–test were used after shuffling the datasets. The novelty parameters Q, ε, and G were defined such that minimal unexpected observations could form groups with significant similarities and, hence, detected as novelties. Specifically, we used $Q = 5$, $\varepsilon = 1.0$, and $G = 2$. Obviously, these parameters can be tuned based on computational and practical considerations or by using a learning procedure. All experiments were executed on an i7-core, 16 GB PC running Microsoft Windows 7 Version 6.1 (Build 7601: Service Pack 1) and MATLAB® Version: 8.0.0.783 (R2012b).

4.3. Benchmark to MSPC methods

In the first study, we compared EB-SPC with non-ensemble SPC approaches to evaluate the improvement of the proposed EB-SPC over conventional SPC frameworks. In particular, the following methods were used for the comparison:

1. PCA/SPE with Hotelling T^2 statistical limits (Chiang *et al.*, 2001).
2. SVDD for novelty detection (Tax and Duin, 2004).
3. Novelty detection based on Gaussian Mixture Models (GMM; Bishop (2006)).
4. Minimum Volume Set (MV-Set) based on the Plug-In estimator approach as proposed in Park *et al.* (2010).
5. The Bayesian Monitoring Statistic (BMS) method, which was proposed specifically for the monitoring of a multi-mode operating system (Ge and Song, 2013).

In all cases, we learned a model for each class and then used these models for the monitoring phase.

For the PCA/SPE with Hotelling's T^2 method, a PCA was applied in which the monitoring of variables is performed using the Hotelling's T^2 and Q statistics, known also as the SPE (see, e.g., Chiang *et al.* (2001)). A control limit was computed for each class and was subsequently used in the monitoring stage to detect abnormal data samples. To apply the SVDD model for novelty detection, we used the implementation of LibSVM Version 3.20 for MATLAB® (Chang and Lin, 2014). As we considered multi-class datasets, one hyper-sphere was learned for each class using the training dataset and then fine-tuned using the validation dataset. A sample point found outside the learned boundary limits was defined as abnormal if supported by sufficient data points, as discussed above. For the SVDD approach, we pre-processed the training dataset so that all features were scaled to a range [0, 1], as this usually helps the convergence in the quadratic optimization (Tax and Duin, 2004). To scale the training dataset, the maximum and minimum values of each feature were extracted and used for value scaling; i.e., $\tilde{x}_{i,k} = (x_{i,k} - max\{\mathbf{x}_i\})/(max\{\mathbf{x}_i\} - min\{\mathbf{x}_i\})$, where $\mathbf{x}_i$ denote the vector values of feature i, $x_{i,k}$ is the kth element of the vector values of feature i, and $\tilde{x}_{i,k}$ is the resulting scaled element. The test dataset was also scaled so that each element of the features $\tilde{x}_{i,k}^{Test} \in [-3\sigma_i; 1 + 3\sigma_i]$, where σ_i represent the standard deviation of the scaled feature i in the training dataset. Saturation was also used to avoid numerical inconsistencies of the SVDD classification algorithm (see Tax (2001)). Thus, because each feature of the training data was scaled to the [0, 1] range, some variation was expected in the test data but not beyond $\pm 3\sigma_i$ limits.

For the GMMs, the algorithm described in Bishop (2006) was implemented. The training dataset was used to estimate the GMM parameters, and the validation set was used to compute a log-likelihood threshold lower limit for each class. In the monitoring stage, the log-likelihood scores of new data points were benchmarked against the log-likelihood lower limits, tagging them as anomalies if their values were found lower than all of these limits. To obtain a log-likelihood lower limit for each class, we estimated the log-likelihood score distribution using the validation dataset, applying a Kernel Density Estimation (KDE) with a Gaussian Kernel, as implemented in Botev *et al.* (2010). As the log-likelihood distribution is a one-dimensional distribution, the KDE obtained an efficient bias-variance trade-off (Bishop, 2006). For classes with a lower number of instances than their dimensionality, a single GMM was used to model all classes. The number of Gaussian components was set such that it obtains a minimal Akaike Information Criterion (Bishop, 2006). For the MV-Set we followed the implementation described in Park *et al.* (2010), as it provides (in the asymptotical sense) the smallest possible type-II error (false negative error) for any given fixed type-I error (false positive error). In Park *et al.* (2010), PCA is used first to reduce the data dimensionality and then a KDE is used to compute the empirical probability in order to find the Minimum Volume set for one data class. In this experimentation, we performed this analysis for the majority classes separately and applied the following simple rule: A sample observation is declared novel if it is classified as novel observations for all data classes. The BMS method applies a decomposition of PCA and ICA, as suggested by Ge and Song (2013), to account for both Gaussian and non-Gaussian measures. The data samples from each target class were independently modeled by means of ICA and its associated residuals and then by means of PCA. The number of components in each model was selected, as specified by Ge and Song (2013), without applying the fuzzy-c clustering step that requires the identification of major classes. The BMS approach aims at detecting nonconforming data points, assuming that new data samples can belong to several modes. Accordingly, one can expect to use the BMS to identify novelties, as the latter do not belong to previously learned classes.

To illustrate one of the tests that were performed in this stage, we slightly adjusted the "Features–Fourier" dataset from the UCI public repository to simulate a scenario of novelty detection in a multi-mode system with overlapping clusters. Similar to Figure 1, each of the plots in Figure 4 refers to a member of the "target class" and shows the Mahalanobis distance between

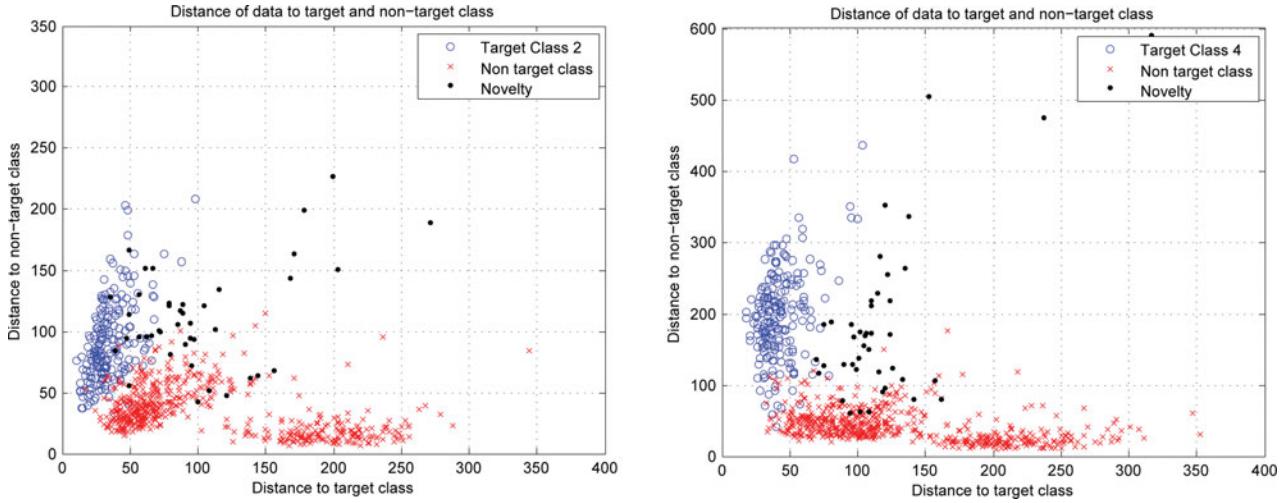

Figure 4. Mahalanobis distances between the target "normal" class, the rest of the classes, and the novelties (Features-Fourier dataset). The *x*-axis measures the distance between target data samples to itself. The *y*-axis shows the distance between the target data samples to the other "non-target" classes. "o"-signs represent target class samples, "x"-signs represent non-target class data samples, and " . "-signs represent novelties observations.

both the target class versus itself (the *x*-axis) and the target class versus the other "non-target" learned classes (the *y*-axis). The "o" signs represent the data points from the target class; "×" signs represent data points from the other learned classes; and the "·" signs represent the novelties that should be detected. Figure 4 highlights that in both cases, the target classes are well separated from the rest of the learned classes. However, novelties, in general, are not well separated from the "in-control" classes. Figure 5 shows the corresponding SPC chart and the F1-Scores for the implementation of the Hotelling T^2-PCA SPC to the "Features–Fourier" dataset. The first 240 samples in the control chart are based on data points from the learned (target) classes, whereas from sample 241 on, data points are from the novelty classes. The Hotelling T^2 control limit was computed based on a 5% type-I error upper limit. As seen, the Hotelling T^2–PCA method cannot clearly distinguish the novelties from the target class. Note that even when the user was exposed to the novelty data during the learning phase and could theoretically rely on a 37% significance-level limit (depicted by the dashed line in the left figure), which lies below the majority of the novelty data points, a very high false alarm rate would result.

Using the same analysis procedure, we compared the proposed EB-SPC against non-ensemble SPC methods for the rest of the datasets. The applied EB-SPC relies on Rotation Forest as the ensemble-of-classifiers, which was found to be statistically insignificant with respect to the EB-SPC version that relies on Random Forest (below, the statistical significance method used in the experiments as well as a comparative study between the two EB-SPC versions are given). Table 4 summarizes the results obtained from 20 UCI repository datasets, and Table 5 summarizes the results obtained from four real-world datasets from metrology measurements in a semiconductor production line. In both tables (and in later comparisons), the maximum average F1-Scores are marked by bold letters, showing that in most of the datasets, the EB-SPC yields better results with respect to the benchmarked methods. Few datasets contain two bold scores, in

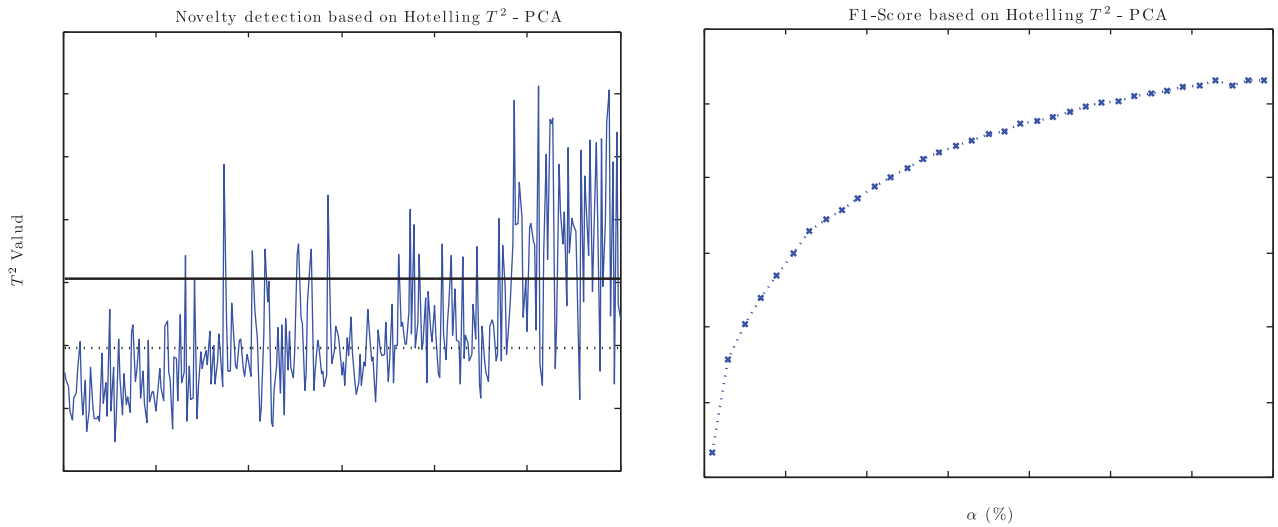

Figure 5. SPC chart (left) and F1-Scores (right) of the Hotelling T^2-PCA method applied to the "Feature - Fourier" dataset from the UCI repository. The first 240 samples belong to the learned ("in-control") classes, whereas the rest of the samples represent novelties (classes that were not available for training). The horizontal lines show the Hotelling limits (a 37% significance level for the upper solid line and a 5% significance level for the lower dashed line). Right: F1-Score for Hotelling T^2-PCA based on all available data (from learned classes and the novelty data samples). The F1-Score obtains a maximum value for a confidence interval of 37%.

Table 4. Comparing EB-SPC versus non-ensemble SPC-based methods on UCI data (F1-Score). Average and standard deviation values of the F1-Score measure are computed over 20 iterations. In each dataset, the highest average F1-Score values are marked in bold (including the next-highest values if they fall within a 10% tolerance from the highest value).

	EB-SPC		*Hotelling PCA/T^2*		*SVDD*		*GMM*		*MV-Set*		*BMS*	
Dataset	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*
Arrhythmia	**0.889**	0.027	0.526	0.029	0.274	0.056	0.309	0.016	0.431	0.106	0.623	<0.001
Audiology	**0.900**	0.027	0.545	0.100	0.545	0.067	0.299	0.072	0.076	0.050	0.584	<0.001
Dermatology	**0.921**	0.028	0.564	0.030	0.632	0.040	0.714	0.068	0.235	0.095	0.648	<0.001
Glass	**0.938**	0.028	0.267	0.070	0.562	0.061	0.299	0.072	**0.960**	0.016	0.636	0.004
Pen Digits	**0.989**	0.030	0.686	0.008	0.613	0.015	0.560	0.001	0.352	0.040	0.573	0.002
Segmentation	**0.943**	0.028	0.832	0.093	0.607	0.076	**0.906**	0.027	**0.859**	0.024	0.542	0.008
Fourier	**0.825**	0.025	0.438	0.013	0.501	0.041	0.531	0.055	0.080	0.031	0.611	0.002
Karhunen-Loeve	**0.845**	0.025	0.525	0.028	0.637	0.035	0.717	0.036	0.630	0.107	0.637	0.011
Pixels	**0.884**	0.027	0.598	0.015	0.636	0.035	0.351	0.016	0.701	0.010	0.667	<0.001
Letters	**0.622**	0.019	0.033	0.004	0.208	0.017	0.018	0.001	0.020	0.002	**0.664**	<0.001
Splice	**0.921**	0.016	0.183	0.008	0.487	0.026	0.132	0.010	0.483	0.041	0.644	0.004
Satimage	**0.790**	0.011	0.138	0.010	**0.716**	0.018	0.616	0.048	0.326	0.016	0.665	0.002
Zoo	**0.895**	0.008	0.484	0.158	0.556	0.102	**0.809**	0.051	0.198	0.024	0.658	<0.001
Wine	**0.807**	0.004	0.490	0.039	0.622	0.067	0.519	0.059	0.556	0.048	0.661	0.007
Waveform	**0.725**	0.007	0.248	0.019	0.631	0.020	0.340	0.004	0.314	0.062	0.638	0.002
Waveform2	**0.707**	0.006	0.241	0.009	0.578	0.026	0.329	0.005	**0.637**	0.024	0.640	0.010
Faults	**0.820**	0.015	0.341	0.039	0.604	0.030	0.311	0.039	0.668	0.000	0.622	0.002
Isolet	**0.855**	0.019	0.650	0.014	0.563	0.026	0.386	0.049	**0.950**	0.012	0.652	0.001
Thyroid	**0.695**	0.067	0.516	0.032	0.621	0.096	**0.708**	0.095	0.623	0.121	**0.653**	0.006
Covertype	**0.680**	0.006	0.107	<2e-3	0.322	0.025	0.151	0.026	0.593	0.005	**0.659**	<0.001

which the second-best method (in a few cases also the third-best method) obtained a performance value within a 10% tolerance of the maximum score, following the comparison method proposed by Sathe and Aggarwal (2016).

Table 4 shows that in all the 20 datasets, the EB-SPC obtains better results than does the Hotelling T^2–PCA method. A possible explanation for the weakness of the Hotelling T^2–PCA in detecting novelties lies in the distribution of its statistics. The underlying assumption for the distribution relies on mean vectors having linear interactions among coupled features within the variance–covariance matrices. In datasets where the classes are well separated, the Hotelling T^2–PCA method results in statistical thresholds that are useful for identifying novelties; for example, as seen in the "Segmentation" dataset, in which the Hotelling T^2–PCA obtains relatively good results. However, as a general note, the Hotelling T^2–PCA distribution assumption was found less practical in real-world novelty processes, as seen in Table 5, that rely on real-world semiconductor data, for which the advantage of the proposed EB-SPC was even more evident. The GMM and SVDD SPC methods obtain a lower performance than the EB-SPC, specifically in datasets of high dimensionality. In 19 out of 20 datasets, the proposed EB-SPC obtains a better performance than that obtained by the SVDD, whereas in one dataset ("Satimage") the SVDD SPC obtains a result within 10% of the tolerance. Similarly, in 17 of 20 cases, the EB-SPC outperforms the GMM technique, whereas in three datasets ("Segmentation," "Zoo," and "Thyroid"), the GMM obtains an equivalent performance to the EB-SPC. With respect to MV-Set, the results in Table 4 show that in 16 out of 20 datasets, the proposed EB-SPC achieves a better performance than the MV-Set, whereas in four out of 20 datasets ("Glass," "Segmentation," "Waveform2," and "Isolet"), both methods obtained equivalent results. As in the case of Hotelling T^2, in datasets where classes might be ill separated, the MV-Set approach has a challenge to identify novelties. Thus, although it has been proven that the MV-Set provides (*in the asymptotical sense*) the smallest possible type-II error for a given fixed type-I error in anomaly detection tasks (Park *et al.*, 2010), it seems that the use of PCA at the learning stage impacts negatively on the overall performance of the MV-Set for novelty detection tasks in high-dimensional space. A possible explanation is that unlike conventional anomalies that are well spread over the sample space (e.g., well represented by white noise), novelties are often visible and correlated only in subsets of attributes that might have been eliminated by the linear combination of the attributes in the PCA step.

A comparison between EB-SPC and the BMS, which was specifically proposed by Ge and Song (2013) to monitor multimode systems, shows that in 17 of 20 datasets, the EB-SPC is found more effective, whereas in three of 20 datasets ("Letters,"

Table 5. Comparing EB-SPC versus non-ensemble SPC-based methods on real-world semiconductor data (F1-Score). Average and standard deviation values of the F1-Score measure are computed over 20 iterations. In each dataset, the highest average F1-Score values are marked in bold (including the next-highest values if they fall within a 10% tolerance from the highest value).

	EB-SPC		*Hotelling PCA/T^2*		*SVDD*		*GMM*		*MV-Set*		*BMS*	
Dataset	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*
Dataset #1	**0.752**	0.019	0.589	0.013	0.348	0.037	0.519	0.028	0.569	0.038	0.523	0.003
Dataset #2	**0.773**	0.023	0.423	0.008	0.473	0.024	0.343	0.018	0.443	0.012	0.614	0.001
Dataset #3	**0.763**	0.015	0.549	0.007	0.473	0.031	**0.698**	0.054	**0.744**	0.050	0.644	0.002
Dataset #4	**0.793**	0.020	0.270	0.008	0.463	0.049	0.251	0.008	0.651	0.028	0.592	<0.001

"Thyroid," and "Covertype"), the two approaches are found to be equivalent. Note that in these datasets, the number of instances is much higher with respect to the data dimensionality.

Table 5 shows the results computed over a real-world dataset obtained from semiconductor lines. In all cases, the obtained F1-Score is inferior to the ones obtained from the UCI repository datasets. A possible explanation might be related to the level of noise in real-world datasets or that the feature selection procedure for building the ensembles was not optimally tuned, leading to poor classification results and, hence, a less robust detection of novelties. Nevertheless, in three out of four datasets, the proposed EB-SPC outperformed the classic approaches for novelty detection, including the multimode statistic method BMS, while in one dataset (#3) the GMM and MV-Set obtain lower yet equivalent results.

We also tested the statistical significance of the results, merging Tables 4 and 5 and using the evaluation methodology recommended in Demšar (2006). First, we applied the non-parametric test with the null hypothesis that EB-SPC is equivalent to the benchmark methods. As recommended by Demšar (2006), we used the Iman-Davenport correction to generate a statistic value that follows an F-distribution. We obtained an F-Statistic value of 75.355, resulting in a rejection of the null hypothesis, based on the critical value of 3.245 at a significance level of 0.05. We also include the post hoc Bonferroni-Dunn statistical test by evaluating the z-statistics between EB-SPC with ensemble Rotation Forest versus EB-SPC with Random Forest. We found that the average rank difference was 0.375, which determined a z-statistic of 0.709 and corresponding p-value of 0.479. Therefore, we conclude that using Rotation Forest or Random Forest for the ensemble base for EB-SPC does not result in a significant difference (thus, in the comparison studies we used the Rotation Forest EB-SPC). Table 6 shows the z-statistic computations, the corresponding p-values, and the Bonferroni-Dunn significance test, based on results in Table 4 and Table 5, for the non-ensemble SPC methods. As seen, based on the post hoc statistic values, one can reject the null hypothesis and conclude that the EB-SPC outperforms the non-ensemble SPC approaches in the considered cases.

4.4. Benchmark to ensemble-of-classifiers methods

To isolate the contribution of the proposed SPC framework from the contribution of the ensemble-based modeling, we also benchmarked the proposed EB-SPC with ensemble-of-classifiers based on Rotation Forests but without applying the proposed SPC framework. To compare the monitoring part, we selected three monitoring methods that combine the output of the ensemble-of-classifiers in a fashion similar to the EB-SPC.

Table 6. Post hoc Bonferroni-Dunn significance test for F1-Score values between EB-SPC and the non-ensemble SPC benchmarked methods.

Methods	*z-Statistic*	*p-Value*
EB-SPC vs. T^2	7.796	<0.0001
EB-SPC vs. SVDD	6.378	<0.0001
EB-SPC vs. GMM	4.903	<0.0001
EB-SPC vs. MV-Set	4.692	<0.0001
EB-SPC vs. BMS	4.055	<0.0001

Specifically, we trained the Rotation Forest by using the training classes' datasets and then combined the ensemble outputs using the T-Norm approach, as proposed by Tax and Duin (2008); the AWE approach, as proposed by Wang *et al.* (2003); and the MineClass method as presented in Masud *et al.* (2009). Finally, we also selected MINAS (De Faria *et al.*, 2016) as another alternative state-of-the-art method for novelty detection that, different from the other benchmarked frameworks, builds an ensemble-of-classifiers based on the known k-means clustering algorithm.

In addition to the evaluation of the F1-Score measures in all experiments, we computed the Area Under the Curve (AUC) measure to quantify the performance of the benchmarked ensemble-based methods by adjusting the estimated class-dependent threshold in the EB-SPC, Rotation Forest, Random Forest, T-Norm, and the AWE approaches. For MineClass we followed the parameterization proposed in Masud *et al.* (2009) and set $K = 50$ as well as defined a novel observation if all classifiers in the ensemble classify the data observation as novel. The number of clusters for each data class for the MINAS method was selected following De Faria *et al.* (2016). Note that the MineClass and the MINAS methods do not incorporate a probability threshold to detect novelties. Instead, both methods use the distance to the closest centroid of clusters within each data class to classify a new data sample and make use of the cluster's hyper-spherical radius to declare a new observation as a novelty. Specifically, if the closest Euclidean distance to the cluster centroid is less than the cluster radius, the observation is classified as being part of the centroid's class. Otherwise, it is classified as a novelty sample. Since the radius of the centroids is critical to detect novelties in both these approaches, we handled the radius values as a threshold of the closest centroid when classifying new observations. By multiplying the radius by a factor in the range $[0,\ 1]$, we obtained different points on the receiver operating characteristic curve that allowed us to compute the AUC values. Tables 7 and 8 compare the EB-SPC against ensemble-of-classifiers approaches, based on the F1-Scores for the UCI repository and the real-world semiconductor datasets, respectively. Table 9 shows the post hoc Bonferroni-Dunn significance test for the ensemble-based approaches, using the F1-Score as the performance criterion. Table 10 (UCI repository datasets) and Table 11 (real-world semiconductor datasets) have similar comparisons, based on the AUC-scores. Finally, Table 12 shows the post hoc Bonferroni-Dunn significance test for the ensemble-based approaches, using the AUC as performance criterion. Similar to the previous experiments, the results obtained for the EB-SPC with Random Forest and Rotation Forest were statistically insignificant and, hence, only the results for the EB-SPC based on Rotation Forest will be further discussed. In Tables 7 and 8, one can clearly see that the proposed EB-SPC outperforms the ensemble-of-classifiers for Rotation Forest showing that the proposed novel detection framework boosts Rotation Forest when applied in novelty detection tasks. Table 7 shows that the EB-SPC achieves a better performance than does the T-Norm in 18 of 20 studied cases, whereas in three of 20 cases ("Fourier," "Karhunen-Loeve," and "Pixels") the EB-SPC achieves equivalent results to the AWE. In general, the AWE achieves a relatively good performance in detecting novelties, because it

Table 7. Comparing EB-SPC versus ensemble-of-classifiers benchmark methods on UCI data (F1-Score). Average and standard deviation values of the F1-Score measure are computed over 20 iterations. In each dataset, the highest average F1-Score values are marked in bold (including the second-highest value if it falls within a 10% tolerance from the highest value).

	EB-SPC		*Rotation forest*		*T-Norm*		*AWE*		*MineClass*		*MINAS*	
Dataset	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*
Arrhythmia	**0.889**	0.027	0.126	0.046	0.669	0.011	0.584	0.046	0.269	0.063	0.229	0.057
Audiology	**0.900**	0.027	0.023	0.008	0.651	0.098	0.627	0.110	0.339	0.064	0.222	0.002
Dermatology	**0.921**	0.028	0.543	0.101	0.406	0.042	0.527	0.086	0.264	0.033	0.280	0.121
Glass	**0.938**	0.028	0.050	0.001	0.666	0.001	0.666	0.001	0.636	0.150	0.210	0.034
Pen Digits	**0.989**	0.030	0.235	0.077	0.612	0.306	0.844	0.057	0.888	0.024	**0.916**	0.011
Segmentation	**0.943**	0.028	0.227	0.078	0.378	0.227	0.209	0.120	0.648	0.075	0.110	0.009
Fourier	**0.825**	0.025	0.161	0.028	0.678	0.010	**0.809**	0.039	**0.785**	0.042	**0.792**	0.226
Karhunen-Loeve	**0.845**	0.025	0.515	0.024	**0.762**	0.009	**0.845**	0.023	**0.849**	0.026	0.689	0.149
Pixels	**0.884**	0.027	0.479	0.047	0.749	0.010	**0.865**	0.018	**0.869**	0.025	0.697	0.169
Letters	**0.622**	0.019	0.362	0.013	**0.676**	0.005	0.515	0.007	0.375	0.015	0.232	0.003
Splice	**0.921**	0.016	0.043	0.025	0.660	0.005	0.654	0.011	0.180	0.001	0.082	0.038
Satimage	**0.790**	0.011	0.100	0.008	0.690	0.019	0.668	0.028	0.473	0.029	0.490	0.019
Zoo	**0.895**	0.008	0.702	0.262	0.677	0.034	0.681	0.038	0.280	0.052	0.197	0.025
Wine	**0.807**	0.004	0.080	0.000	0.553	0.236	0.590	0.209	0.251	0.072	0.356	0.171
Waveform	**0.725**	0.007	0.024	0.013	0.552	0.008	0.395	0.010	0.393	0.040	0.180	0.041
Waveform2	**0.707**	0.006	0.022	0.009	0.593	0.005	0.437	0.014	0.346	0.022	0.126	0.054
Faults	**0.820**	0.015	0.132	0.041	0.712	0.006	0.602	0.050	0.179	0.000	0.122	0.059
Isolet	**0.855**	0.019	0.286	0.037	0.681	0.010	0.653	0.028	**0.849**	0.005	**0.846**	0.015
Thyroid	**0.695**	0.067	0.166	0.000	0.447	0.077	0.591	0.247	0.164	0.016	0.057	0.010
Covertype	**0.680**	0.006	0.025	0.006	0.534	0.007	0.317	0.008	0.003	0.001	0.082	0.038

Table 8. Comparing EB-SPC versus ensemble-of-classifiers methods on real-world semiconductor dataset (F1-Score). Average and standard deviation values of the F1-Score measure are computed over 20 iterations. In each dataset, the highest average F1-Score values are marked in bold (including the second-highest value if it falls within a 10% tolerance from the highest value).

	EB-SPC		*Rotation forest*		*T-Norm*		*AWE*		*MineClass*		*MINAS*	
Dataset	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*
Dataset #1	**0.752**	0.019	0.267	0.043	0.696	0.014	0.621	0.022	0.354	0.003	0.202	0.048
Dataset #2	**0.773**	0.023	0.273	0.076	**0.778**	0.018	**0.845**	0.009	0.294	0.046	0.162	0.082
Dataset #3	**0.763**	0.015	0.103	0.051	**0.715**	0.009	**0.829**	0.022	0.442	0.091	0.128	0.118
Dataset #4	**0.793**	0.020	0.074	0.028	0.664	0.008	0.547	0.048	0.324	0.060	0.132	0.075

merges the ensemble outputs using the classification errors as weights. Nevertheless, weighting the classification error is not sufficient to generate thresholds that identify novelties for the investigated UCI repository datasets. In Table 8, one can see that the AWE, T-Norm, and EB-SPC obtain similar performances in detecting novelties in real-world semiconductor datasets in two out of four datasets, while the EB-SPC outperforms the other two datasets. However, let us note that a more extensive dataset collection from metrology processes will be required to further compare EB-SPC to AWE and T-Norm and draw comprehensive conclusions.

Table 7 shows that EB-SPC outperforms MineClass in 16 out of 20 studies cases, whereas MineClass achieves comparable performance results to the proposed EB-SPC in four out of 20 cases. Table 8 shows that MiniClass achieves a poorer performance on average when applied to the semiconductor real-world datasets. Regarding the MINAS method, Table 7 shows that it achieves a comparable performance in three out of 20 analyzed cases ("Pen Digits," "Fourier," and "Isolet"), while the EB-SPC has the best performance in the remaining 17 datasets. Table 8 shows that when applied to the real datasets from the semiconductor industry, MINAS obtains a much poorer performance with respect to all the other methods.

Table 9. Post hoc Bonferroni-Dunn significance test for F1-Score values between EB-SPC and ensemble-of-classifiers benchmarked methods.

Methods	*z-Statistic*	*p-Value*
EB-SPC vs. Rotation Forest (RoF)	6.000	<0.0001
EB-SPC vs. T-Norm	2.050	0.040
EB-SPC vs. AWE	2.200	0.028
EB-SPC vs. MineClass	4.953	<0.0001
EB-SPC vs. MINAS	5.011	<0.0001

Table 9 shows the Bonferroni-Dunn significance test based on Tables 7 and 8 for the ensemble-of-classifiers approaches as shown previously for the non-ensemble SPC approaches. The obtained F-statistic value is 34.041, leading to the rejection of the null hypothesis, as the critical value used is 3.245 with a significance level of 0.05. Table 9 shows the z-statistics and the corresponding p-values when comparing the proposed EB-SPC versus the ensemble-of-classifiers benchmarked approaches. The null hypothesis is rejected at a significance level of 0.05, leading to the conclusion that the EB-SPC outperforms the ensemble-of-classifiers benchmarked approaches.

Tables 10 and 11 show the resulting AUC computations for the ensemble-of-classifiers approaches. Table 10 shows that in 10 of 20 datasets, the EB-SPC achieves a better performance than the benchmarked Rotation Forest approach, whereas in the other datasets both methods achieve comparable results. The EB-SPC achieves a better performance than the T-Norm method in 19 of 20 datasets and an equivalent result in one dataset. The AWE method achieves a higher AUC value only in one of 20 datasets ("Thyroid") and comparable AUC values in

Table 10. Comparing EB-SPC versus ensemble-of-classifiers methods on UCI data (AUC). Average and standard deviation values of the AUC measure are computed over 20 iterations. In each dataset, the highest average AUC values are marked in bold (including the second-highest value if it falls within a 10% tolerance from the highest value).

	EB-SPC		*Rotation forest*		*T-Norm*		*AWE*		*MineClass*		*MINAS*	
Dataset	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*
Arrhythmia	**0.681**	0.023	**0.630**	0.031	0.460	0.052	0.590	0.054	0.594	0.073	0.449	0.112
Audiology	**0.733**	0.049	0.483	0.019	0.440	0.052	0.638	0.106	0.546	0.117	0.302	0.004
Dermatology	**0.759**	0.035	**0.807**	0.019	0.239	0.146	0.578	0.150	0.414	0.120	0.348	0.263
Glass	0.710	0.029	0.502	0.007	0.000	0.000	0.523	0.014	**0.950**	0.029	0.356	0.000
Pen Digits	**0.983**	0.003	0.773	0.013	0.452	0.412	**0.943**	0.015	**0.964**	0.014	**0.977**	0.003
Segmentation	**0.777**	0.067	0.586	0.065	0.145	0.145	0.153	0.155	**0.755**	0.085	0.172	0.037
Fourier	**0.901**	0.016	0.598	0.036	0.686	0.030	**0.814**	0.042	**0.875**	0.036	**0.833**	0.304
Karhunen-Loeve	**0.942**	0.010	**0.873**	0.015	0.596	0.060	**0.961**	0.010	**0.917**	0.016	0.631	0.271
Pixels	**0.972**	0.008	**0.882**	0.014	0.653	0.043	**0.959**	0.009	**0.948**	0.015	0.648	0.242
Letters	**0.683**	0.006	**0.633**	0.006	0.454	0.007	0.493	0.014	**0.730**	0.011	0.523	0.020
Splice	**0.686**	0.009	**0.653**	0.011	0.431	0.019	**0.686**	0.012	0.423	0.025	0.408	0.051
Satimage	**0.827**	0.007	0.595	0.011	0.730	0.040	0.723	0.018	0.723	0.018	**0.751**	0.020
Zoo	**0.977**	0.030	**0.978**	0.012	0.409	0.129	**0.991**	0.009	0.528	0.136	0.578	0.044
Wine	**0.668**	0.116	0.501	0.041	0.353	0.216	0.496	0.195	0.383	0.172	0.305	0.011
Waveform	**0.722**	0.008	0.507	0.008	0.459	0.007	0.506	0.008	**0.739**	0.043	0.211	0.008
Waveform2	**0.727**	0.009	0.501	0.009	0.444	0.010	0.510	0.009	**0.690**	0.022	0.305	0.014
Faults	**0.648**	0.016	0.570	0.023	0.484	0.025	0.639	0.040	0.434	0.026	0.403	0.010
Isolet	**0.799**	0.015	0.664	0.012	0.315	0.030	0.704	0.029	**0.843**	0.012	**0.825**	0.042
Thyroid	0.522	0.032	0.533	0.015	0.523	0.343	**0.693**	0.138	**0.674**	0.014	0.294	0.090
Covertype	**0.469**	0.007	**0.483**	0.004	0.380	0.005	**0.452**	0.005	**0.485**	0.250	0.132	0.023

Table 11. Comparing EB-SPC versus ensemble-of-classifiers methods on real-world data (AUC). Average and standard deviation values of the AUC measure are computed over 20 iterations. In each dataset, the highest average AUC values are marked in bold (including the second-highest value if it falls within a 10% tolerance from the highest value).

	EB-SPC		*Rotation forest*		*T-Norm*		*AWE*		*MineClass*		*MINAS*	
Dataset	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*
Dataset #1	**0.698**	0.026	**0.647**	0.020	0.481	0.039	**0.635**	0.033	0.469	0.051	0.055	0.046
Dataset #2	**0.901**	0.010	0.799	0.048	0.753	0.022	**0.887**	0.012	0.641	0.033	0.085	0.079
Dataset #3	**0.855**	0.038	0.745	0.043	0.642	0.035	**0.851**	0.022	**0.866**	0.056	0.069	0.129
Dataset #4	**0.726**	0.041	0.624	0.018	0.299	0.039	0.554	0.039	0.557	0.038	0.097	0.102

seven out of 20 datasets. The obtained results show that EB-SPC outperforms the MineClass novelty detection approach in eight out of 20 datasets; MineClass outperforms the EB-SPC in two 2 out of 20 cases, while in 10 out of 20 cases both methods obtain comparative performance results. The MineClass approach is found to be the second-best performing novelty detection framework with respect to the AUC score values. For the real semiconductor dataset shown in Table 11, one can see that the AWE performs similarly to EB-SCP in three out of four datasets, outperforming the MineClass and MINAS methods.

The Bonferroni-Dunn significance tests for the AUC values (by merging Table 10 and Table 11) are shown in Table 12, resulting in an F-statistic value of 9.021 and leading to rejection of the null hypothesis, based on a critical value of 3.245 with a significance level of 0.05. Table 12 shows that Rotation Forest, AWE, and MiniClass are slightly worse with respect the proposed EB-SPC in terms of significance analysis, in conformance with the obtained performance results in the experiments. Nevertheless, EB-SPC shows that in all cases, the null hypothesis can be rejected at a significance level of 0.05, proving the efficacy of the proposed EB-SPC over the benchmarked approaches.

Table 12. Post-hoc Bonferroni-Dunn significance test of AUC values between EB-SPC and ensemble-of-classifiers benchmarked approaches.

Methods	*z-Statistic*	*p-Value*
EB-SPC vs. RoF	2.010	0.044
EB-SPC vs. T-Norm	5.950	<0.0001
EB-SPC vs. AWE	2.040	0.040
EB-SPC vs. MineClass	2.011	0.044
EB-SPC vs. MINAS	5.801	<0.0001

The obtained results show that the EB-SPC framework outperforms classic and state-of-the-art approaches for novelty detection. Thus, the high classification performance of ensemble-based classifiers when combined with a new SPC framework can be successfully used for novelty detection tasks. Moreover, note that both Random Forest and Rotation Forest were applied in this study using a basic CART configuration. CART has known advantages, such as being simple, providing direct handling of mixed types of attributes, and robustness to outliers within the training data and with respect to irrelevant attributes. Other Decision Tree–based classifiers (e.g., ID3 (Quinlan, 1986), C.45 (Quinlan, 1993), DID (Ben-Gal *et al.*, 2014), and CHAID (Kass, 1980)) can be easily implemented in the proposed EB-SPC. Define tree-based classifiers, partition the feature space into a set of "hyperplanes" during the training phase, and then fit a simple majority decision in each subspace. Thus, they are easily trained and usually maintain good performance in classification tasks when combined in ensembles (Fernandez-Delgado *et al.*, 2014). However, one should note that their partitions are occasionally rough, leaving low-density regions in the feature space in which data are sparse (Duda *et al.*, 2001). Voting techniques or weighting outputs based on

classification errors rarely identify novelties unless the classes are well separated, both in the training dataset and during the monitoring stage. The proposed EB-SPC overcomes some of these challenges by combining the posterior class conditional probability from each class, as shown in Equation (1). A consequence of this approach is that if low-posterior class probabilities are obtained from all ensembles, their combined scores become a low posterior probability, leading to an improved performance for novelty detection. Benchmark methods such as T-Norm and AWE are based on approaches different from the EB-SPC. T-Norm computes a threshold for each class, approximating the prior class distribution by summing separately the classifier outputs over all training data samples for each class. Then, it uses the minimal value of the prior class probability over the training sample to define a threshold value for each class. In comparison, the EB-SPC uses all classes to define a threshold for each class that is estimated by bootstrapping. The implementation of statistical inference, combined with class-dependent information, leads the EB-SPC to outperform T-Norm in most of the analyzed cases. Conversely, AWE shows a relatively better performance than does the T-Norm in our experiments. AWE uses an estimated class error and an estimated classification error for each ensemble element and merges the ensemble outputs, although differently than the EB-SPC. Accordingly, in our study, it appears that the AWE captures more information on novelty classes by evaluating a class-dependent threshold. These weights are then combined linearly with the ensemble output to obtain a weighted class posterior probability. Nevertheless, the AWE's classification error, as obtained during the training phase, appears to underperform in feature space regions with low data density. This is the reason why the performance of the AWE in the novelty detection is relatively poorer than that of EB-SPC. Notice that MineClass makes use of the k-means clustering algorithm on node leafs to estimate the cluster hyper-radius radius where expected data samples might be classified and, consequently, novel observation be detected. With respect to the F1-Score performance values, in most of the analyzed cases, MineClass seems to generate a far-reaching area where novel observations are hardly identified and, henceforth, it achieves a poorer performance with respect to the proposed EB-SPC. Finally, recall that MINAS also uses k-means in each data class to classify data samples based on the minimal Euclidean distance to the cluster centroids of the different classes. Similar to the MineClass method, the MINAS approach uses the hyper-radius of the resulting selected cluster to detect novelties. This is the reason for similar obtained F1-Score results of the MINAS and MineClass methods. However, the MineClass method outperforms MINAS with respect to the AUC scores, due to the higher classification quality derived from the Rotation Forest.

Another advantage of the EB-SPC approach over the benchmarked ensemble-of-classifiers methods is that the statistical limit of the type-I error can be specified for each data class during the training stage of the ensemble. However, similar to other multi-class novelty detection methods, such as the T-Norm (Tax and Duin, 2008), the EB-SPC does not control the overall type-I error of falsely detected novelties, in particular, when data samples lie on the boundaries of a corresponding data class y_c but are classified as part of another class by a different threshold. Under such a scenario, the performance of the proposed EB-SPC method might be limited in terms of controlling the overall type-I error.

5. Conclusions

This article introduces the EB-SPC approach for novelty detection in multi-class systems. The EB-SPC combines ensemble-of-classifiers, Bayesian inference, and an SPC paradigm. The random selection of subspaces over high-dimensional datasets leads to an efficient identification of novelties when using a Bayesian inference mechanism. This efficiency is a result of the novelties often being correlated over smaller subspaces, unlike noise-based anomalies that can be scattered uniformly over all of the space. An advantage of the EB-SPC with respect to other anomaly detection methods is that it does not apply a feature selection procedure at the learning stage that might result in eliminating features that are later found to be informative for the identification of novelties. The proposed EB-SPC provides a robust framework for various classification algorithms in the ensemble without requiring the user to apply a specific classifier. This is an important property of EB-SPC, since different classifiers are found to be dominant in different problem settings. Thus, the ability to provide a general ensemble-based SPC framework that is classifier-agonistic can promote the effectiveness of the proposed approach in various applications and use cases. In this study, we specifically used Rotation Forest and Random Forest based on the popular CART classifiers that provide a simple means of computing class probabilities by statistical inference methods. CART classifiers have the advantage of handling both categorical and numerical (discrete and continuous) features as well as of treating missing data via surrogate features. The proposed scheme is nonparametric and extends the scope of SPC to account for novelties in multi-modal systems. Another aspect of the proposed framework is that depending upon the data being monitored, the training method can be individually chosen, thus allowing it to be extended to online training scenarios. Finally, the obtained results show that EB-SPC has the potential to be implemented in metrology as a specific process monitor in the semiconductor industry. A clear drawback of ensemble-based methods that rely on randomly selected subspaces, including the proposed EB-SPC, is that they often require several repetitions before converging to an efficient solution. Such an iterative process can be computationally intensive and time consuming. One potential research direction to overcome this challenge is to replace the random subspace-selection mechanism with a more "analytic" one; for example, using information-theoretic measures to select promising subspaces, a method that is currently being studied by the authors.

Funding

This research was partially supported by the MAGNET/METRO450 Consortium (http://www.metro450.org.il/).

Notes on contributors

Marcelo Bacher was born in Buenos Aires, Argentina. He completed his engineering studies in electronics at the Technical University in Buenos Aires and obtained honors. In 2004, he completed his M.S. in Information Technology in Mannheim, Germany. He was an R&D engineer at Continental AG, in the Department of Hybrid and Electric Vehicles in Berlin,

Germany. He developed algorithms for lithium-ion battery fault detection, parameter estimation, and battery condition calculation. His contribution to the firm included several patents. Currently he is pursuing a Ph.D. in the Industrial Engineering Department at Tel Aviv University, under the supervision of Professor Irad Ben-Gal. He currently researches novel machine learning algorithms for monitoring complex industrial processes.

Irad Ben-Gal is a full professor and the head of the AI and Business Analytics Lab at the Faculty of Engineering in Tel Aviv University. He is a visiting professor at Stanford University, teaching analytics in action and co-heading the "Digital Life 2030" research project. His research interests include monitoring of stochastic processes, machine learning, and information theory applications to industrial and service systems. Irad is the former chair of the Quality Statistics & Reliability (QSR) society at INFORMS, a member in the Institute of Industrial Engineers (IIE), and a member of the European Network for Business and Industrial Statistics (ENBIS). He has written three books, published more than 100 scientific papers and patents, and received numerous awards for his work. He has supervised dozens of graduate students, served on the editorial boards of several professional journals, led many R&D projects, and worked with companies such as Oracle, Intel, GM, AT&T, Applied Materials, Siemens, Kimberly Clark, and Nokia. He is the co-founder and chairman of CB4 ("See Before"), a startup backed by Sequoia Capital that provides granular predictive analytics solutions to retail organizations. He is an advisory board member in several startup companies that focus on AI.

References

Aggarwal, C.C. (2015) Outlier analysis. *In Data mining* (pp. 237–263). Springer International Publishing.

Aggarwal, C.C. and Sathe, S. (2017) *Outlier Ensembles: An Introduction*, Springer.

Aldrich, C. and Auret, L. (2013) *Unsupervised Process Monitoring and Fault Diagnosis With Machine Learning Methods*, Springer, London, England.

Al-Khateeb, T., Masud, M., Al-Naami, K., Seker, S., Mustafa, A., Khan, L., … Han, J. (2016) Recurring and novel class detection using class-based ensemble for evolving data stream. *IEEE Transactions on Knowledge and Data Engineering*, **28**(10), 2752–2764.

Bache, K. and Lichman, M. (2013) UCI machine learning repository. Avaiable at http://archive.ics.uci.edu/ml.

Ben-Gal, I. (2010) Outlier detection, in *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, Springer.

Ben-Gal, I., Dana, A., Shkolnik, N. and Singer, G. (2014) Efficient construction of decision trees by the dual information distance method. *Quality Technology & Quantitative Management*, **11**(1), 133–147.

Ben-Gal, I. and Singer, G. (2004) Statistical process control via context modeling of finite-state processes: An application to production monitoring. *IIE Transactions*, **36**(5), 401–415.

Bishop, C. (2006) *Pattern Recognition and Machine Learning*, Springer, New York, NY.

Bodesheim, P., Freytag, A., Rodner, E. and Denzler, J. (2015) Local novelty detection in multi-class recognition problems, in *Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision*, IEEE Press, Piscataway, NJ, pp. 813–820.

Bodesheim, P., Freytag, A., Rodner, E., Kemmler, M. and Denzler, J. (2013) Kernel null space methods for novelty detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Press, Piscataway, NJ, pp. 3374–3381.

Botev, Z., Grotowski, J. and Kroese, D. (2010) Kernel density estimation via difussion. *The Annals of Statistics*, **38**(5), 2916–2957.

Breiman, L. (2001) Random forests. *Machine Learning*, **45**, 5–32.

Breiman, L., Friedman, J., Stone, C.J. and Olshen, R. (1984) *Classification and Regression Trees*, Chapman and Hall, New York, NY.

Byon, E., Shrivastava, A.K. and Ding, Y. (2010) A classification procedure for highly imbalanced class sizes. *IIE Transactions*, **42**(4), 288–303.

Chandola, V., Banerjee, A. and Kumar, V. (2007) Anomaly detection: A survey. Report, Department of Computer Science and Engineering, University of Minnesota: Minneapolis, MN.

Chang, C.-C. and Lin, C.-J. (2014) LIBSVM: A library for support vector machines. Available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

Chiang, L., Russell, E. and Braatz, R. (2001) *Fault Detection and Diagnosis in Industrial Systems*, Springer-Verlag, London, UK.

Davenport, M.A., Baraniuk, R.G. and Scott, C.D. (2006) Learning minimum volume sets with support vector machines, in *Proceedings of the 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, IEEE Press, Piscataway, NJ, pp. 301–316.

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

Demšar, J. (2006) Statistical comparisons of classifiers over multiple data sets. *International Journal of Machine Leaning Research*, **7**, 1–30.

Diebold, A. (2001) *Handbook of the Semiconductor Metrology*, Marcel Dekker, Inc., Austin, TX.

Duda, R.O., Hart, P.E. and Stork, D.G. (2001) *Pattern Classification*, John Wiley & Sons, New York, NY.

Efron, B. and Tibshirani, R. (1997) Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, **92**(438), 548–560.

Fernandez-Delgado, F., Cernadas, E. and Barro, S. (2014) Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, **15**, 3133–3181.

Fumera, G., Roli, F. and Giacinto, G. (2000) Reject option with multiple thresholds. *Pattern Recognition*, **33**(12), 2099–2101.

Ge, Z.Q. and Song, Z.H. (2013) *Multivariate Statistical Process Control: Process Monitoring Methods and Applications*, Springer, London, UK.

Jacobs, R.A., Jordan, M.I., Nowlan, S.J. and Hinton, G.E. (1991) Adaptive mixtures of local experts. *Neural Computation*, **13**(1), 79–87.

Jumutc, V. and Suykens, J. (2014) Multi-class supervised novelty detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **36**(12), 2510–2523.

Kass, G.V. (1980) An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, **29**(2), 119–127.

Kenett, R.S. and Zacks, S. (2014) *Modern Industrial Statistics with Applications in R, MINITAB and JMP*, John Wiley & Sons, Ltd, West Sussex, UK.

Kuncheva, L. (2004) *Combining Pattern Classifiers. Methods and Algorithms*, John Wiley & Sons, Hoboken, NJ.

Lazzaretti, A.E., Tax, D.M., Neto, H.V. and Ferreira, V.H. (2016) Novelty detection and multi-class classification in power distribution voltage waveforms. *Expert Systems with Applications*, **45**, 322–330.

MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics And Probability*, pp. 281–297, University of California Press, Berkeley, CA.

Marques, H., Campello, R., Zimek, A. and Sander, J. (2015) On the internal evaluation of unsupervised outlier detection, in *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*, pp. 7, ACM, New York, NY.

Masud, M.M., Al-Khateeb, T.M., Khan, L., Aggarwal, C.C., Gao, J., Han, J., and Thuraisingham, B. (2011) Detecting recurring and novel classes in concept-drifting data streams, in *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, IEEE Press, Piscataway, NJ, pp. 1176–1181.

Masud, M.M., Gao, J., Khan, L., Han, J. and Thuraisingham, B. (2009) Integrating novel class detection with classification for concept-drifting data streams, in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 74–94, Springer, Berlin, Heidelberg.

Masud, M.M., Gao, J., Khan, L., Han, J. and Thuraisingham, B. (2011) Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Transactions on Knowledge and Data Engineering*, **23**(6), 859–874.

Montgomery, D.C. (2008) *Introduction to Statistical Quality Control*, John Wiley & Sons, New York, NY.

Özgür, A., Özgür, L. and Güngör, T. (2005) Text categorization with class-based and corpus-based keyword selection, in *Proceedings of the International Symposium on Computer and Information Sciences*, pp. 606–615, Springer, Berlin, Heidelberg.

Park, C., Huang, J.Z. and Ding, Y. (2010) A computable plug-in estimator of minimum volume sets for novelty detection. *Operations Research*, **58**(5), 1469–1480.

Pimentel, M., Clifton, D., Clifton, L. and Tarassenko, L. (2014) A review of novelty detection. *Signal Processing*, **99**, 215–249.

Quinlan, J.R. (1986) Induction of decision trees. *Machine Learning*, *1*, 81–106.

Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Francisco, CA.

Rodriguez, J.J. and Kuncheva, L.I. (2006) Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Learning*, **28**(10), 1619–1630.

Sathe, S. and Aggarwal, C. (2016, December) Subspace outlier detection in linear time with randomized hashing, in *Data Mining (ICDM)*, 2016 IEEE 16th International Conference on (pp. 459–468), IEEE.

Schölkopf, B., Williamson, R. Smola, A. J., Shawe-Taylor, J., and Platt, J. C. (2000). Support vector method for novelty detection. In *Advances in neural information processing systems* (pp. 582–588).

Scott, D.W. and Sain, S.R. (2005) Multi-dimensional density estimation, in *Handbook of Statistics*, pp. 229–261, Elsevier, Delft, The Netherlands.

Tax, D.M. (2001) One-class classification. Ph.D. thesis, Technische Universiteit Delft, The Netherlands.

Tax, D.M. and Duin, R.P. (2004) Support vector data description. *Machine Learning*, **54**(1), 45–66.

Tax, D.M. and Duin, R. (2008) Growing a multi-class classifier with a reject option. *Pattern Recognition Letters*, **29**(10), 1565–1570.

Upadhyaya, D. and Singh, K. (2012) Classification based outlier detection techniques. *International Journal of Computer Trends and Technology*, **3**(2), 294–298.

Vapnik, V. N. (1998) *Statistical Learning Theory*, Wiley-Interscience, New York, NY.

Wang, H., Fan, W., Yu, P.S. and Han, J. (2003) Mining concept-drifting data streams using ensemble classifiers, in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 226–235.