JOURNAL OF SOFT COMPUTING AND DATA MINING
e-ISSN: 2716-621X

JSCDM

Vol. 6 No. 3 (2025) 154-169
https://publisher.uthm.edu.my/ojs/index.php/jscdm

High-Dimensional Data Stream Classification: Improving Random Patch Online Ensemble Classifier

Hadjer Imene Bensaoula[1]*, Sarah Nait Bahloul[2]

[1] *SIMPA Laboratory, Computer Science Department,*
University of Science and Technology of Oran Mohamed Boudiaf,
El Mnaouer BP1505 Bir El Djir, Oran, 31000, ALGERIA
[2] *LSSD Laboratory, Computer Science Department,*
University of Science and Technology of Oran Mohamed Boudiaf,
El Mnaouer BP1505 Bir El Djir, Oran, 31000, ALGERIA

*Corresponding Author: hadjerimene.bensaoula@univ-usto.dz
DOI: https://doi.org/10.30880/jscdm.2025.06.03.011

Article Info

Received: 6 August 2025
Accepted: 13 November 2025
Available online: 30 December 2025

Keywords

Data stream mining, online machine learning, incremental classification, high-dimensional stream, online ensemble classifiers, streaming random patches, compressed sensing

Abstract

In recent years, the amount of data produced by human activities has increased massively, giving rise to a constant flow of data generated in real time, known as data streams. Data stream classification requires both incremental and adaptive learning approaches, mainly due to the challenges inherent in the data stream's rapidly changing patterns. The Streaming Random Patches (SRP), investigated in this work, offers a robust online ensemble model for evolving data stream classification. The latter uses incremental decision trees, Hoeffding Trees (HT), as base learners for online forecasts. Each tree is incrementally trained on a unique random patch formed via global feature subspacing and online bagging to ensure ensemble variety. The OB brings bagging to streaming. It suggests instance weights for training frequency instead of sampling with replacement. A drift detection strategy replaces outdated base learners in each tree to keep ensemble relevance to recent data and prevent outdated predictions. The ensemble is incrementally built by testing the HTs before training, updating base learner weights based on testing predictions. Weighted majority voting determines ensemble predictions. Therefore, this study aims to retain good SRP performance when identifying high-dimensional streams. The unpredictable nature of data stream instances and their controllable dimensions can degrade the online classifier's prediction quality, availability, and execution time. To increase SRP classifier performance, we refine the compressed sensing (CS) technique before incremental stream processing to ensure efficient subspace selection. Instead of the HT as the base classifier, we employ the Extremely Fast Decision Tree (EFDT) as a more statistically efficient base learner in the final model. The SRP and other techniques improved high-dimensional data stream prediction performance. Average accuracy gains were +0.15% to +5.43%. The suggested modifications reduced execution time by 95.69%, indicating the method's Green AI alignment.

1. Introduction

Modern data sources, including social networks, real-time financial transactions, IoT devices, and sensors, produce dynamic data generated at very high rates [1][2], known as data streams. A Data Stream (DS) can be defined as a real-time unlimited sequence of data instances $DS = \{I\_1, I\_2, I\_3, \ldots, I\_t, \ldots\}$, whereas each stream instance $I_t$arriving at an instant (t) is described by an M-dimensional feature vector [3].

The enormous volume of data produced in real time makes the storage of data streams extremely resource-intensive, and almost impossible to store in their entirety, as the streams are unlimited. Additionally, a central challenge in data streams stems from concept drift, where the underlying patterns in the data at a time (t) may evolve over time $(t + k)$. Consequently, traditional batch-based machine learning approaches, which rely on static settings and full data storage, are unsuitable for processing real-time evolving data streams [4]. To assess the limitations of these batch data mining methods, incremental data stream mining models [5][6], and online learning approaches [4][7] have been introduced to efficiently extract insights from continuous, real-time data streams.

Among online supervised learning [8] methods, several incremental classifiers have emerged in the literature to address online classification. Such models are required to continuously process, and learn from infinite sequences of data with evolving distributions to assign each stream instance a target class label [3], while maintaining high accuracy. These classifiers may operate in single-model mode, or as an online ensemble of several single online classifiers that learn simultaneously from the streams and employ a voting mechanism to yield classification outcomes. The Adaptive Random Forest (ARF) [9], first proposed in 2017, constitutes the first online ensemble implementation that reflects the benefits of conventional random forests in a streaming setting. To operate efficiently in an online learning environment, the model replaces static decision trees with unique online classifiers, namely incremental decision trees. These trees are tested, built, and constantly refined as stream instances arrive. To promote diversity among the ARF base learners, it incorporates the online bagging [13], and random feature subspace selection at each node. When a concept drift is detected, the ARF updates its ensemble by fully retraining new trees, and substituting its outdated ones.

A robust online ensemble classifier (OEC) critically requires diversity among its base learners. If all base learners become highly similar, they can easily all converge to an outdated representation of a concept. Such synchronized obsolescence forces the OEC to initiate frequent training, updates, and deployment of new base learners. Consequently, predictive performances suffer until new base learners take effect. In contrast, diverse ensembles are more resilient to abrupt performance declines, as they require only partial selected updates.

Accordingly, the Streaming Random Patches (SRP) [10][11], registers as one of the implementations of the OEC architecture, suggesting an alternative approach to foster ensemble diversity, by introducing an incremental adaptation of the random patches. Being partially similar to ARF, the model uses a set of adaptive mechanisms, including concept drift detection methods, to track changes in the stream distributions caused by concept drift and thus initiate updates to the online ensemble model. Along with a set of Hoeffding Trees (HT) [12], to progressively grow incremental decision trees as base learners, and ensure the SRP ability to make online predictions.

Different from the ARF per-tree-node subspacing mechanism, known as local subspacing, the SRP online ensemble implements a global subspace selection that can be referred to as per-tree subspacing. Combining the global subspacing and online bagging [13], it introduces incremental random patches used to construct sets of data to be used for testing and training the ensemble's base learners. This architecture enables the framework to combine the strength of online ensembles along with a systematic base learner's diversity method to enhance classifying evolving stream instances. While state-of-the-art ensemble classifiers effectively adapt to concept drift by leveraging online learning, nevertheless this mechanism remains insufficient when processing streams originating from real-world big data applications.

The stream mining community highlights that the drifting and dynamic behavior of these streams is not the only burden that arises in the incremental learning, as many other challenges coexist, namely class imbalance [14], stream high dimensionality [15], in addition to energy and resource constraints [16], multi-label and multi-class stream instances.

The ARF, registered as the first implementation of steam mining random forests, was often established as the benchmark of several works, addressing the aforementioned challenges in both case studies and theoretical investigations [17][18][19] . Nevertheless, a notable evolution in online random forest subspacing was proposed in 2025, which advocated moving from local feature selection at individual tree nodes to a global subspace strategy, with the aim of improving ARF predictive stability. This concept stands at the core of the contribution introduced in the Adaptive Random Trees Ensemble (ARTE) [20] framework, which posits that superior prediction accuracy is achieved through the implementation of variable-sized subspaces for each ensemble element.

Consequently, while addressing previously mentioned stream mining challenges in the OEC, the usage of global subspacing started to become a recurring pattern in many recent works. Notable studies with prominent contributions include : The Online Ensemble of Multi-Label Hoeffding Adaptive Trees (OEMLHAT) [21] is an online classifier for real-time predictive maintenance, addressing multi-label data streams from industrial environments. OEMLHAT is built on a global subspace strategy to promote diversity among base learners and distinguishes itself by incorporating dynamic feature subspaces within its ensemble of Hoeffding trees. This adaptation allows the model to adjust feature usage adaptively as data evolves. Furthermore, recent efforts to address class imbalance within online ensembles have yielded notable approaches, exemplified by the Smart Adaptive Ensemble Model for Multiclass Imbalanced Non-Stationary Data Streams (SAEM) [22]. Established for multiclass imbalanced data streams subject to concept drift, SAEM begins by initializing its ensemble with base learners trained on globally selected feature subspaces and continuously monitors feature distributions for shifts that signal a warning level. Upon detecting concept drift, SAEM deploys a background set to train new classifiers exclusively on the affected attributes. Concurrently, SAEM adjusts instance weights to prioritize minority classes, effectively tackling both class imbalance and distribution shifts in a cohesive manner.

Before the emergence of ARTE, OEMLHAT, SAEM, and other recent ensemble methods that leverage global subspace selection, its conceptual origins can be traced back to earlier research. As previously noted, the Streaming Random Patches (SRP) [10][11] effectively applied a global feature sampling strategy across online bagging weighted stream instances to obtain incremental random patches to promote diverse base learners. Despite its foundational basis in global subspacing, the SRP contribution to online ensembles' diversity through random patches was rarely explored in depth. In previous studies, SRP's role has often been restricted to acting as a reference point in accuracy comparisons during the evaluation of new ensemble approaches, as illustrated in [23], [24], and [25] .

Consequently, the current study addresses the high dimensionality challenge that arises in a stream classification scenario when building the SRP model. In incremental learning, the dimensional parameter M, representing the feature space of incoming instances cannot be predicted. Given this critical uncertainty, in this work, we suggest that the SRP model should implement a dimensionality reduction mechanism. Particularly as this model utilizes both global and random feature subspace selection, we hypothesize that global subspacing would lead base learners to be stuck with irrelevant features, notably as high-dimensional data inevitably includes such attributes. In addition, random feature selection potentially neglects informative features. This leads to the acquisition of insignificant concepts, and frequent resets of base learners, thus elevating resource consumption.

Consequently, our research aims to ensure the relevancy of subspaces chosen for each SRP base learner construction, when addressing high-dimensional stream instances, by integrating the Compressed Sensing (CS) mechanism [26]. Moreover, our contribution is extended by modifying the existing core matrix implemented with state-of-the-art CS. Refining the sensing matrix, would guarantee that each patch will contain the maximum amount of crucial information. Furthermore, this study investigates the usage of the Extremely Fast Decision Tree (EFDT) [27] as a base learner in the SRP classifier. This statistically efficient incremental tree implementation operates via fast splits, and demonstrates superior performance over HT.

The present paper is organized as follows: The first section provides an overview of data stream classification, focusing on online ensembles implementing incremental decision trees, namely the Streaming Random Patches (SRP), which will constitute the baseline of this study. The next section outlines the challenges experienced within the SRP when addressing the curse of high dimensionality in data streams, which motivated this study. Afterwards, we detail our suggested contributions. Subsequently, a description of the experimental setup, datasets, and obtained results are highlighted. Lastly, the manuscript is accompanied by a conclusion summarizing the main results.

2. Background

Data stream classification, also known as incremental classification [28], emerged as a key research topic due to its applicability in many real-world applications [29][30]. Considering that data streams require real-time acquisition and online processing [31], literature states that batch (static data) classifiers are inefficient to accomplish the online classification task, particularly since they assume that the entire data set would be accessible before they initiate the model construction [32].

Consequently, incremental (data stream) classifiers have been utilized because of their ability to continuously learn from each incoming stream instance. These models constantly refine the classifier accuracy and ensure it is trained on unseen data, as each instance is first used for testing and then training the model. In real-world stream classification contexts, once the online classifier is established, the latter is suggested for drops in prediction performance due to changes in the underlying stream distribution, namely when concept drift starts to occur. As outlined in the literature, this challenge arises under 04 distinct scenarios: sudden, gradual, recurrent, and incremental [33]. Sudden drift is a phenomenon where the data distribution changes suddenly from a concept C_1, to a C_2 concept in a short period of time. Incremental drift consists of a sequence of small changes that evolve slowly over time. Gradual drift occurs when the transition from one data distribution to another is slow and gradual.

During this transition, the two concepts coexist simultaneously until the new concept dominates. Recurrent drift occurs when an old concept that has disappeared resurfaces either regularly or randomly. Therefore, proactive monitoring of data stream status and classifier progressive adaptations are essential to overcome concept drift in its four types and prevent the model from becoming obsolete [34]. Incremental classifier adaptivity implements either active or passive update mechanisms. The passive approach consists of triggering updates when data stream instances arrive. Active updates, on the other hand, can only be triggered when concept drift is detected by the specific drift detection methods to provide more systematic model updates. A more detailed description of drift detection methods may be found in [35].

2.1 Online Single Classifiers

The Hoeffding Trees (HT) [12] are the pioneers among single passive stream classifiers, offering a systematic implementation of incremental decision trees. Literature states that HT can create a tree classifier in an online context that is equivalent to a tree generated under batch learning conditions [11]. The incremental growth of the HT tree leaves is regulated by the Hoeffding bound, guaranteeing model stability. As referenced by authors in [10], this stability is characterized by the consistent selection of the optimal splitting attribute at time t, remaining invariant to new data instances or stream chunks arriving at subsequent times $t+k$.

Despite the HT being a fundamental approach to incremental classification, several studies [2] [36] have shown that the passive update mechanism inherent in its implementation leads to inefficient adaptation to concept drift. Consequently, the literature has been expanded by various studies targeted at ensuring systematic updates to the Hoeffding tree to maintain the model's performance in the presence of highly drifting stream instances.

The main contributions to Hoeffding tree-based stream classifiers can be divided into two main categories: online ensemble classifiers and single HT models that are further broken down into two categories (passive and active) according to how they detect and react to concept drift. Drift detection algorithms are explicitly used to implement responsive model updates for single active variants, such as Hoeffding Adaptive Trees (HAT) [37] and Extremely Fast Hoeffding Adaptive Trees (EFHAT) [38]. In contrast, passive single classifiers, including Hoeffding Trees and Hoeffding Any Time Trees [27], implement regular model updates in order to address any changes in data distribution.

Literature emphasizes that maintaining model robustness and prediction availability are essential for addressing all kinds of data stream challenges. This includes high dimensionality, concept drift, and data velocity. According to the authors of [39], single models (passive or active) may fail to achieve the necessary efficiency levels or accurately capture the complexity associated with data streams on occasions when such flows are highly variable. Consequently, our paper focuses on improving the implementation of an online ensemble classifier.

2.2 Online Ensemble Classifiers

Online ensembles [29] employ a set of single classifiers, referred to as base learners, trained from various segments of a data stream to generate a collective decision [3]. In addition to drift detection approaches that are used in two stages: at the warning and drift levels, thus guaranteeing that base learners are trained and updated on the most recent data distribution [11]. These online ensemble learning algorithms have been shown to offer several advantages over single classifiers when dealing with data streams undergoing severe conceptual drift in any of its 4 scenarios [2][14]. This efficacy is attributed to their natural approach in responding to new concepts by creating a new base classifier, updating existing classifiers, and deleting obsolete ones [3][36].

As such, several contributions in online ensembles were observed, implementing either online bagging or online boosting [13], which are incremental adaptations of the batch machine learning techniques, bagging and boosting [30].

In this study, we detail two of the leading online ensembles that use online bagging [40], namely the Adaptive Random Forest (ARF) [9] and Streaming Random Patches (SRP) [10][11]. Online bagging was chosen as the basis for this study because of its superiority over online boosting in data stream mining. As stated in [40], incorporating online boosting lowers the model's efficacy when dealing with drifting streams. The difficulty stems from identifying which booster elements must be reset. This complication arises because each step of incremental amplification relies on previous predictions [40].

2.2.1 Adaptive Random Forests

J. of Soft Computing and Data Mining Vol. 6 No. 3 (2025) p. 154-169

When the Adaptive Random Forest (ARF) [9] classifier was launched, it consistently outperformed other single HT solutions for classifying evolving data streams. ARF builds the model using online bagging and several Hoeffding trees as base learners. Drift detection methods are used to initiate model updates. The detection approaches are applied in two modes: warning and detection. They enable model upgrades and the systematic replacement of outdated base learners within the ARF.

Online bagging is implemented in the ARF to introduce diversity into the model's base learners. This is accomplished by simulating instance replacement by having each instance allocated a weight, a procedure controlled by the λ parameter. Such a parameter controls how frequently an instance is shown to a base learner in online sampling while also assisting in denoting levels of randomness. In ARF, the λ default value is set to 6, as this value has been shown to increase the model's prediction accuracy. Furthermore, rather than using the complete feature set throughout the learning process and at each leaf level, each instance is represented by a randomly selected subset of features. The construction of the ARF ensemble begins with the initialization of a set of HT base learners. Subsequently, each base learner grows incrementally, powered by subspaces of instance samples that are randomly selected and repeated based on the weights established by the online bagging technique. Possible concept drift occurrences are continuously monitored using the warning-level drift detection technique. When drift alerts are received, new base learners are initiated, and these models are built alongside current ones. The replacement of current base learners with these new models occurs after a drift is confirmed by drift detection methods. This architectural approach ensures that models adapt to emerging concepts and recover from potential drifts.

2.2.2 Streaming Random Patches [9]

Authors in [10][11] illustrate that in incremental classification environments involving online sets, selecting feature subsets once per base learner, known as global subsampling, performs better than selecting local subsets for each base learner's internal nodes. The Streaming Random Patches (SRP) [10][11] suggest an enhancement to ARF to improve the sub-spacing mechanisms applied to stream instances. Unlike ARF, which selects feature subspaces separately for each leaf of the base trees and uses online bagging independently, SRP suggests combining global subspacing and online instance bagging simultaneously. This integration leads to the creation of "random patches", where a unique random subset of features is selected globally once before each base model is built.

In addition, SRP incorporates an active drift detection strategy, addressing the challenges posed by non-stationary data streams in which relevance can change over time. When a drift is detected, SRP regenerates new random subspaces for each of the background learners. SRP draws an analogy to the dropout technique in deep learning [11]. This technique randomly omits some neural networks to enhance accuracy. In SRP, randomly subsampling attributes for each base model is identical to randomly dropping attributes [11]. Meanwhile, during testing, all features are taken into account by the ensemble voting mechanism, in the same way that neural networks use all neurons after being trained with the exclusion technique [11]. These combined strategies, including global selection of subsets, active drift detection, and analogy with dropout, make SRP a more systematic incremental ensemble classifier. Consequently, this research centers on improving the SRP model, an advancement upon the foundational ARF implementing the global features subspacing and random patches.

2.3 High Dimensional Data Stream Classification

Real-world data streams, which are often characterized by high dimensionality [41], impose significant challenges for data analysis, decision-making processes [42] and online learning approaches. A high-dimensional stream instance is described by having an increasing number of features. Additionally, the dynamic nature of these data streams, characterized by a continuous influx of new features and a change in the perceived relevance of existing features due to concept drift, makes the identification of an optimal subset of features extremely difficult.

This is further complicated by the presence of both relevant, and redundant features in high-dimensional data [43] [44]. Dimensionality reduction is the process of projecting high-dimensional data into a lower-dimensional space, thereby reducing the number of dimensions while preserving the important information for the learning task [45] [46]. Usage of such techniques mitigates the complexities of real-world datasets, thereby improving online models' performances [6]. Consequently, this method becomes crucial for ensuring the scalability and real-time analysis of dynamic data streams [46]. Projecting this issue into online ensemble classifiers, namely ARF, the following improvements have been introduced. Starting with the Principal Component Analysis ARF (PCA-ARF) [47], where PCA is applied to reduce the dimensionality of the dataset, thereby facilitating the extraction of the most informative features. Subsequently, the Adaptive Random Forest algorithm is utilized to build the incremental stream classifier. The second contribution to this topic was denoted by Compressed Sensing Adaptive Random Forests (CS-ARF) [48]. The CS technique is implemented as an internal online preprocessing mechanism to transform high-dimensional data stream instances into a lower feature vector, while preserving most of the original information, whereas the obtained low-dimensional representation of data is used to incrementally build the model.

In researching the basis for this study, we found that CS-ARF demonstrated good levels of accuracy when processing concrete high-dimensional data, unlike PCA-ARF, where no empirical experiments have been conducted on real high-dimensional data. The advent of CS has provided a new direction for dimensionality reduction techniques, where it has gained importance when applied to batch data mining and processing, as shown by recent studies in different fields of medical imaging [49]. Accordingly, our study investigates combining compressed sensing (CS) with the SRP classifier.

2.3.1 Compressed Sensing [26]

Compressed sensing (CS) [26] is a technique that allows signals or images to be captured and efficiently reconstructed using fewer measurements compared to other approaches, without losing essential information. Many different fields frequently use compressed sensing [50], examples of such implementations can be found at [51][52][53][54]. The originators of the CS technique argued as follows: Why *waste so much energy collecting all the data when most of it will be rejected? Is there no method to directly assess the proportion that won't be abandoned?*. Therefore, they introduced CS to propose a dimensionality reduction mechanism that would work as if it were possible to efficiently acquire the important information about the signals/images while ignoring the part of the data that would ultimately be discarded by compression. The fundamental aspect of CS is its ability to recover sparse or compressible signals from a smaller number of measurements than is usually necessary, suggesting that CS can capture the few most informative features of a signal. The mathematical expression of the CS process is as follows:

$$y = \Phi\, v \tag{1}$$

Where y is the measurement vector, Φ the sensing matrix and v is the original signal. As highlighted in equation (1), high-dimensional data is projected into a lower-dimensional subspace by CS using a sensing matrix. A mathematical operator that is used in dimensionality reduction, where the important information and features of the data are preserved. Maximum coherence and the restricted isometry property (RIP) are two interdependent qualities that are used to choose the best CS measurement matrix.

Authors in [26][55], emphasize that the restricted isometry property (RIP) is concurrently a necessary and sufficient condition for data reconstruction, with randomness playing an important role in the formation of the sampling matrix. RIP aims to ensure that a matrix approximates the distances between sparse signals during dimensionality reduction.

Accordingly, state-of-the-art CS uses the Gaussian random matrix. The Gaussian random matrix is a well-known class of random sampling matrices that satisfy the restricted isometry property with high probability. It is popular because of its simplicity, where the elements of the matrix are drawn independently from a Gaussian distribution. Since its emergence, the CS technique has been widely used for signals and images [26][55]. In addition, a successful application of this technique to the classification of online data streams has recently been introduced in the CS-ARF online classifier variant [48].

3. Motivation: Streaming Random Patches when Processing High-Dimensional Data Streams

Real-world data streams, with their unique properties compared to static data, frequently present unpredictable behaviors and challenges, among which high dimensionality becomes an unavoidable constraint. Conveniently, the online learning approach must demonstrate robust performance. As mentioned in previous sections, the SRP patches result from the application of online bagging using the weight (w) assigned to the stream instance.
Together with the global feature subspacing, that initiates the base learners with the subspaces of the features (chosen randomly). Consequently, upon arrival at the SRP, each instance is fully transmitted to every base learner. Each base classifier then utilizes its designated subspace and the corresponding instance weights to generate patches. These generated patches are subsequently employed to train the HT model associated with that base classifier.

However, when SRP processes and learns from instances of medium- to high-dimensional streams, inevitably, some of these base learners become ineffective. This is because the subspaces used for patch selection are selected one time only, which may imply that the base learners are blocked by irrelevant features, a situation that may be made inevitable by the fact that the subspaces are randomly selected without any basis as to their importance to the flows, resulting in unnecessary consumption of resources of the irrelevant base learner. Additionally, in the prediction process, the SRP implementation suggests the use of the full instance to make a prediction via a voting mechanism, whereas base learners implementing irrelevant noisy features could

J. of Soft Computing and Data Mining Vol. 6 No. 3 (2025) p. 154-169

easily lower the overall accuracy of the model. The observed characteristics of the SRP model in handling high-dimensional data streams constituted the principal motivation for this study.

4. Methodology

Current applications of compressed sensing require that input data exhibit the sparsity property. Considering that such criteria are exhibited in high-dimensional streams, as indicated in [56], where the authors state that the curse of dimensionality causes high-dimensional spaces to be sparse. The present study addresses the SRP high-dimensional stream classification challenges. We suggest implementing CS, as an additional layer prior to random patches that feature selection and tree growth in the ensemble base classifiers (see Figure 1).

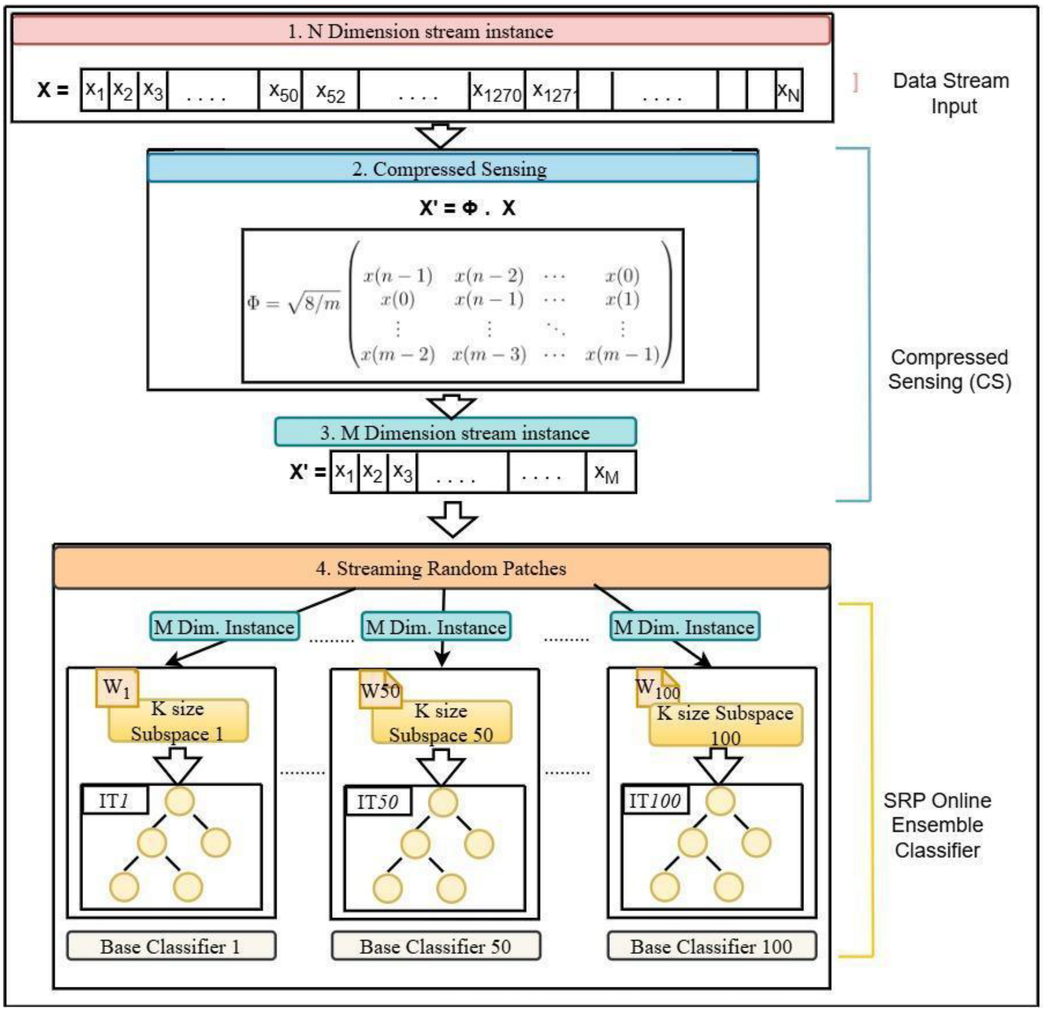

Fig. 1 *Description of the suggested compressed sensing SRP (CS-SRP) architecture*

The CS will project each N-sized stream instance into an M-dimensional feature space (with $M < N$). The SRP ensemble then assigns a random K-dimensional subspace (with $K < M$) for each of its base classifiers. Accordingly, each classifier combines the assigned subspaces and instance weights (W) to construct the random patches required to grow and update the incremental tree.

This layer (denoted as CS in Figure 1) aims to ensure that the model grows utilizing only a relevant representation of the high-dimensional feature space, preventing resource waste on irrelevant ones. On that account, we intend to proceed with a two-stage approach: Firstly, we investigate the existence of an improved measurement matrix for use in the compressed sensing layer, as the optimal choice of measurement matrix has always been a problem in compressed sensing itself, as denoted by authors in [26] [55]. Next, to find a model that outperforms the existing HT base learner, we examine the behavior of the CS-SRP obtained in conjunction with a recent implementation of the HT, namely the Extremely Fast Decision Tree (EFDT) [27]. Regarding drift detection, the proposed CS-SRP framework retains the core SRP drift handling mechanism utilizing the ADWIN (Adaptive Windowing) detection method. This involves ADWIN continuous monitoring of individual CS-SRP base classifiers' performance. Whereas if the active tree performance deterioration triggers the warning threshold, CS-SRP initiates a new base learner with a unique feature subset for constructing its incremental tree.

This newly initiated model undergoes training in parallel with the existing classifier. Once concept drift is confirmed, this model replaces the outdated active classifier, thereby maintaining ensemble effectiveness in evolving data environments. Figure 1 highlights the final state of our contribution, called CS-SRP, which, in summary, will result from the above improvements, to be carried out in two phases:

1. Investigating an improved implementation of compressed sensing addressing online learning.
2. Investigating a variant for the SRP base learner, denoted as incremental tree (IT) in Figure 1, to refer to both the default HT and investigated EFDT in the suggested CS-SRP model.

4.1 Improving the Compressed Sensing: Changing the Gaussian Random Matrix

Accurate reconstruction of the original signal is achievable through the compressed sensing mechanism. This capability persists despite noise or incomplete sampling during acquisition. However, the authors indicate a recurring problem. Determining a stable measurement matrix remains an issue. Such a matrix is necessary to protect the compressed signal from damage during dimensionality reduction (compression phase) [26] [55].

Accordingly, the appropriate requirement for a measurement matrix is that it preserves the length of the sparse dimensions [55]. The restricted isometry property (RIP) is a fundamental matrix characteristic. It allows for the recovery of a sparse high-dimensional signal from its compressed measurement. Matrices satisfying the RIP are crucial in compressed sensing [57]. Using random designs for these matrices is the simplest method to meet such criteria [26] [55] [57]. Therefore, the random Gaussian matrix is used in the default CS implementation to suit this demand. However, while searching for other matrices meeting the RIP characteristic, non-random matrices were investigated. The authors of [50] state that the deterministic matrices were found not to meet the RIP characteristic, making them unsuitable for use in the CS method. Consequently, the authors of [58] suggested as a contribution to design a sensing matrix using a chaotic sequence, and exploit its usage within a semi-deterministic matrix called the Toeplitz matrix. Resulting in a Toeplitz-structured Chaotic Sensing Matrix (TsCSM). Their study was motivated by the fact that the chaotic sequence itself succeeds in conforming to the RIP characteristic. The authors utilized a widely recognized discrete chaotic system, the logistic map, whose difference equation is presented below.

$$z_+ = r\, z\, (1 - z) \tag{2}$$

With $r \in]0,4]$ and $z \in]0,1[\subset \mathbb{R}$ representing the discrete state. They defined $z_i(t)$ as the output sequence produced by the logistic map (equation 2) with an initial condition of $z_i(0)$. Subsequently, $x_i(t)$ denotes the regularization of $z_i(t)$, expressed as:

$$x_i(t) = z_i(t) - 0.5 \tag{3}$$

To construct the TsCSM matrix, authors in [50] highlight that an initial condition $z(0) \in \mathbb{R}$ is first set for the chaotic system (see equation 2). From this, a sequence $x \in \mathbb{R}^n$ is generated according to equation 3. These steps enable the construction of a Toeplitz-structured matrix $\Phi \in \mathbb{R}^{m \times n}$, which appears in the following configuration, with $\sqrt{\frac{8}{m}}$ serving as a normalization factor [50].

$$\Phi = \sqrt{8/m} \begin{pmatrix} x(n-1) & x(n-2) & \cdots & x(0) \\ x(0) & x(n-1) & \cdots & x(1) \\ \vdots & \vdots & \ddots & \vdots \\ x(m-2) & x(m-3) & \cdots & x(m-1) \end{pmatrix} \tag{4}$$

Our study examines the application of the TsCSM matrix within compressed sensing to reduce the dimensionality of high-dimensional data streams. This reduction aims to ensure that SRP utilizes only pertinent features when generating training patches for its base learners.

4.2 Investigating the Usage of the EFDT as an SRP Base Learner

Upon setting the CS layer, in the second phase of our contribution, we sought to use the Extremely Fast Decision Tree (EFDT) presented in [27] as a base learner in the CS-SRP classifier. Our choice is motivated by the fact that the EFDT proved to be statistically more efficient and more tolerant to concept drift than the HT used in the default SRP implementation. While Hoeffding trees (HT) build a tree by delaying the selection of a split until an attribute suggesting high confidence is found (this confidence is controlled using the Hoeffding bound), and do not revisit this attribute choice. The EFDT, in contrast, selects an attribute and implements a split as soon as it seems beneficial, offering a non-negligible information gain with sufficient certainty, leading to faster learning. Then it modifies the split if a better one emerges later in the construction process, a process also known as revising the tree's sub-branches.

Conveniently, the EFDT uses the Hoeffding bound to assess if the potential benefit of splitting on the best attribute outweighs the benefit of not splitting, or of splitting on the attribute currently in use. resulting in the EFDT choosing an attribute to begin with and switching when necessary. Given the statistical superiority of EFDT over the HT implementation, as demonstrated in prior research by [27], this study seeks to explore EFDT's potential advantages on high-dimensional data streams by using it as a base learner within the suggested CS-SRP, implementing the refined version of compressed sensing. Table 1 outlines our novel contributions, which will be experimentally evaluated with respect to the baseline Streaming Random Patches. In addition, while the authors of [48] introduced the CS-ARF method, their work did not include a comparative assessment between the full-feature ARF and the proposed CS-ARF variant. Addressing this gap, and to validate the established superiority of SRP over ARF as documented in the literature, the state-of-the-art SRP and our suggested variants' performances will be compared to the ARF predictive accuracy levels.

Table 1 *Description of online ensemble classifiers employed for evaluating predictive performances*

Approach	Description
ARF	Default Adaptive Random Forests
SRP	Default Streaming Random Patches (with HT as base learners)
$CS_{Gauss}SRP_{HT}$	SRP (with HT trees as base learners), combined with default compressed sensing, implementing the Gaussian random matrix
$CS_{TsCSM}SRP_{HT}$	SRP (with HT trees as base learners), combined with compressed sensing, implementing the TsCSM matrix
$CS_{TsCSM}SRP_{EFDT}$	SRP (with EFDT trees as base learners), combined with compressed sensing, implementing the TsCSM matrix

5. Experiments and Results

As part of our study, a suitable online learning environment is needed to run comparative experiments on the ARF, the existing SRP and the SRP combined with both CS (using both Gaussian and TSCSM matrices) and the EFDT. This setup should enable model implementation, testing, and analysis in an online learning context. Therefore, we utilized the River Python library. It is an online learning platform introduced in [59], that implements a test-then-train learning paradigm to construct their models, where stream instances are first used for testing and subsequently for incrementally training the model, as opposed to establishing distinct training and testing sets.

The open-source River package provides multiple state-of-the-art learning methods, among which the ARF, SRP, HT, and EFDT classifiers were found. The ARF and SRP online ensembles were launched using their default implementations and configurations introduced in River. To guarantee a fair comparison, both benchmark models were configured with the same base learner hyperparameters. This configuration, inherited from the ARF implementation, included a tie threshold of 0.05, a split decision error tolerance (delta) parameter of 0.01, and an information gain split criterion. The extensible nature of the River platform facilitated the sequential integration of our proposed approaches, implementing the same previously detailed base learner's configuration. The hyperparameters for the 5 ensemble components were held constant: the online bagging lambda parameter was set to 6, and the sub-spacing size was defined as 60% of the original feature dimensionality. Furthermore, while the standard ARF and SRP implementation employs an ensemble of *100* base learners, this research adapts the ensemble to a condensed configuration of *10* base classifiers, constrained by computational resource limitations.

The sensing matrices size will be $m \times n$, with m set to *50;* the standard deviation is minimal for this value, as emphasized in [48], and n denotes the initial stream instance size. Each base learner will be given a grace period value of *50* (*g=50*) denoting when the latter starts learning. While the EFDT tree's split reevaluation period was configured based on the processed dataset size: 2000 for large datasets, 300 for medium, and 100 for small. We suggest that in a deployment environment with infinite data streams, the default value of 2000 provides a stable and effective configuration. The chaotic sequence parameter, r, was set to 4, as this value sits at the upper bound of its theoretically valid interval, $r \in]0,4]$. Considering the stochastic initialization of the z parameter and to ensure methodological consistency, each experiment was repeated 3 times. Reported results reflect the average (avg) and standard deviation (std) of performance metrics across conducted runs. Although the river package provides so-called stream generators. The present study will use real-world high-dimensional data sets. These data sets simulating both binary and multiclass online classification scenarios (see Table 2) were obtained from the UCI repository [60]. According to the authors of [61], when assessing the performance of online stream classifiers, the most crucial factor is their accuracy, which they consider to be a sufficient evaluation measure. For the purpose of evaluating the methods proposed in this study, two parameters will be examined: CPU time (measured in seconds) and the accuracy that quantifies the rate of correctly classified stream instances. Powered by an 11th Generation Intel® Core™ i5-1145G7 processor, our local system was used for conducting our research experiments. The latter is equipped with 4 physical cores and 8 logical processors, operating at a base frequency of 2.60 GHz, and supported by 16 GB of physical memory (RAM).

Table 2 Description of used data streams

Datasets	Features	Instances	Classes
Toxicity	1203	171	2
Enron	1000	1702	2
Ads	1558	3279	2
Parkinson	754	756	2
Cnae	856	1080	2
Gas Sensor	128	13910	6
Har	561	10299	6
Nomao	118	34465	2

5.1 Results

The empirical results presented in Table 3 validate the established superiority of SRP over ARF in terms of predictive performance, as posited by [10] [11]. Nevertheless, the obtained accuracy gains come at a substantial computational expense, which aligns with conclusions made by the authors of [24][62] , and more importantly, lends strong support to our hypothesis that SRP is a non-optimal learner for high-dimensional data streams. Accordingly, we now shift our focus to use the obtained results when comparing the performances of the proposed CS-SRP variants against the default SRP. Starting with a simple comparative analysis targeting model runtime, the empirical results highlight that by using compressed sensing, the runtime of any of the CS-SRP variants (implementing either HT or EFDT as base learners or using the default Gaussian random detection or TsCSM matrices) is significantly optimized. Optimizations ranged from 27% to 95.69% of the initial runtime.

Across all three CS-SRP variants, this runtime efficiency results from utilizing CS as internal preprocessing. The latter ensured that the subspaces for each base learner are selected from a smaller, relevant representation containing the same information as the original. Although theoretically the usage of EFDT involves several split revisions, when used with CS-SRP, CS reduced the need for reevaluating splits, as the likelihood of initiating node splits with irrelevant or noisy features is decreased. Similarly, the results underline that the EFDT model, when used as a CS-SRP base learner, causes the classifier to perform in a more optimal, energy-efficient mode, having an execution time always bounded by that of the default SRP, affirming the EFDT-suggested fast splits mechanism. Furthermore, while the HT base learner does not revise splits, using it in the default SRP yields high execution time [24] [62]. The root cause is that the HT base learner would wait until it finds a feature that ensures high confidence before splitting, which leads to iterative splitting attempts and causes model growth delays. Consequently, our findings validate the proposed hypotheses concerning execution time and inadvertent resource consumption caused by high-dimensional stream classifications.

Table 3 Online classification performances of the SRP and CS-SRP variants on high-dimensional streams

Datasets	Algorithm	Mean Acc. (%)	Std Acc. (%)	Mean Time(s)	Std Time(s)
Gas Sensor	ARF	92.27	±0.11	355	±106
	SRP	92.87	±0.1	1013	±289
	$CS_{GAUSS}SRP_{HT}$	96.33	±0.22	585	±184
	$CS_{TsCSM}SRP_{HT}$	95.81	±0.38	483	±147
	$CS_{TsCSM}SRP_{EFDT}$	97.56	±0.15	697	±111
Har	ARF	87.46	±0.07	1587	±223
	SRP	88.41	±0.46	3755	±572
	$CS_{GAUSS}SRP_{HT}$	87.06	±1.06	473	±24
	$CS_{TsCSM}SRP_{HT}$	86.06	±0.84	476	±55
	$CS_{TsCSM}SRP_{EFDT}$	82.03	±0.03	821	±68
Toxicity	ARF	72.35	±0.48	10	±1
	SRP	71.96	±0.73	30	±2
	$CS_{GAUSS}SRP_{HT}$	80.39	±1.54	2	±0
	$CS_{TsCSM}SRP_{HT}$	79.61	±1.69	2	±0
	$CS_{TsCSM}SRP_{EFDT}$	83.92	±0.55	2	±0
Enron	ARF	95.85	±0.35	97	±7
	SRP	95.75	±0.14	266	±26
	$CS_{GAUSS}SRP_{HT}$	95.83	±0.13	22	±1
	$CS_{TsCSM}SRP_{HT}$	95.67	±0.15	24	±1
	$CS_{TsCSM}SRP_{EFDT}$	93.83	±1.45	25	±3
Ads	ARF	99.58	±0.01	103	±19
	SRP	99.61	±0.03	598	±13
	$CS_{GAUSS}SRP_{HT}$	99.61	±0.14	41	±5
	$CS_{TsCSM}SRP_{HT}$	99.61	±0.14	49	±2
	$CS_{TsCSM}SRP_{EFDT}$	99.35	±0.01	56	±8
Parkinson	ARF	81.98	±1.15	48	±1
	SRP	82.82	±0.72	115	±3
	$CS_{GAUSS}SRP_{HT}$	75.43	±0.45	13	±2
	$CS_{TsCSM}SRP_{HT}$	75.63	±0.96	11	±1
	$CS_{TsCSM}SRP_{EFDT}$	81.59	±0.71	24	±1
Cnae	ARF	99.72	±0	34	±2
	SRP	99.72	±0	93	±7
	$CS_{GAUSS}SRP_{HT}$	99.72	±0	12	±1
	$CS_{TsCSM}SRP_{HT}$	99.72	±0	13	±1
	$CS_{TsCSM}SRP_{EFDT}$	99.72	±0	4	±0
Nomao	ARF	97.15	±0.06	506	±78
	SRP	97.27	±0.04	1559	±293
	$CS_{GAUSS}SRP_{HT}$	96.72	±0.07	1171	±78
	$CS_{TsCSM}SRP_{HT}$	96.42	±0.1	515	±50
	$CS_{TsCSM}SRP_{EFDT}$	96.20	±0.1	539	±46

The second stage of this discussion addresses the models performances on high-dimensional data stream classification. The approaches $CS_{GAUSS}SRP_{HT}$, $CS_{TsCSM}SRP_{HT}$ and $CS_{TsCSM}SRP_{EFDT}$ exhibited competitive accuracy gains compared to the basic SRP. Their average accuracy gains over the default *SRP* performance were +2.78%, +0.15%, and +5.43%, respectively. This can be attributed to Gaussian random and TsCSM matrices preserving the RIP property. The RIP property emphasizes maintaining signal distance approximations during dimensionality reduction. Incorporating these methods into the CS-SRP variants ensures the preservation of concepts outlined by stream instances in high-dimensional spaces.

The semi-deterministic nature of TsCSM makes it a preferable choice for severe high-dimensional binary classification tasks. This is demonstrated by $CS_{TsCSM}SRP_{EFDT}$ performance on the Toxicity dataset as features dimensionality surpasses stream length indicating improvements of +11.96% over SRP, with a standard deviation highlighting stability compared to the bases SRP. This is further supported by TsCSM performances on Ads (1558 features) binary stream sets where models show both stability and predictive efficiency without additional computational runtime as seen in SRP.

The slight randomness implemented in the chaotic sequence of the TsCSM further suggests a better performance rate for multiclass streams of non-severe dimensionality, as highlighted in the GasSensor dataset with an accuracy gain estimated at +3.46% , +2.94% and +14.69% by the$CS_{GAUSS}SRP_{HT}$, $CS_{TsCSM}SRP_{HT}$ and $CS_{TsCSM}SRP_{EFDT}$, respectively. Among the 3 variants, the CS-SRP implementing the Gaussian random matrix was a better proposition for multiclass classification. CS_{Gauss} offered excellent performance for the severely high-dimensional multiclass HAR dataset, reaching an accuracy level of 87.06%.

Further comparisons with default SRP performance on the same multi-class HAR dataset (500 feature), revealed that the SRP achieves an accuracy level of 88.41%, but at the cost of an exponential rise in execution time (9 times slower than the $CS_{GAUSS}SRP_{HT}$). This inefficiency is caused by base learners struggling with irrelevant features, triggering a corrective, yet computationally expensive, retraining process. At severe high-dimensionality, this leads to a decline in the overall performance of SRP, as observed in the GasSensor and Toxicity datasets.

6. Conclusion

In today's world, many applications generate unlimited amounts of dynamic data in real time, known as data streams. Online learning approaches can learn progressively and efficiently from these data streams. This study aimed at improving the performance of an online ensemble classifier, namely Streaming Random Patches (SRP), when processing high-dimensional streams. We suggested incorporating a dimensionality reduction mechanism, namely compressed sensing (CS), as a preprocessing layer to overcome the curse of data streams' high dimensionality. Furthermore, instead of using only the default CS implementation, the current study examined the behavior of the CS towards another sensing matrix, namely the Toeplitz-structured Chaotic Sensing Matrix (TsCSM), to maintain the consistency of compressed data in an online learning stream classifier. Additionally, we investigated combining the resulting CS-SRP with a statistically more optimal base learner, known as Extremely Fast Decision Tree (EFDT). The CS-SRP variants introduced in this study outperformed SRP, suggesting robust stream classification performances. They mitigated accuracy degradation caused by high-dimensional streams in binary and multiclass scenarios. The average accuracy gains were +2.78%, +0.15%, and +5.43% for the respective CS-SRP variants, namely the $CS_{GAUSS}SRP_{HT}$, $CS_{TsCSM}SRP_{HT}$ and $CS_{TsCSM}SRP_{EFDT}$. Moreover, they achieved better execution times, with improvements attaining 95.69%. As the current experiments were necessarily limited to 10 base learners due to resource constraints, future efforts will examine the model's performance and any exposure to overfitting with a larger ensemble size. To mitigate any potential information loss, the reliability of the sensing matrices will be tested against more complex concepts in high-dimensional data streams. Furthermore, acknowledging that concept drift remains a significant challenge for online learning algorithms, a promising work venue lies in refining the CS-SRP base learners' drift detection and adaptation sensitivity by exploring recent literature [38] [63] [64].

Conflict of Interest

The authors declare that they have no conflict of interest regarding the publication of this paper.

Author Contribution

The authors confirm their contribution to the paper as follows: Hadjer Imene Bensaoula: ***conceptualization****,* ***methodology****,* ***implementation****,* ***analysis,*** *and* ***draft manuscript preparation****. Sarah Nait Bahloul:* ***Supervise****,* ***review****, and* ***edit****. All authors reviewed the results and approved the final version of the manuscript.*

Declaration of AI Use in Manuscript Preparation

Regarding the use of the AI tools, we, the authors of the manuscript, affirm that this study is entirely our own original research and that **no results** were generated by an artificial intelligence (AI) technology. However, in order to improve the clarity of our own ideas, expressions and to make our original research and findings simpler to read, the authors used QuillBot and DeepL Translate to assist in grammar checking and language editing when expressing and describing the details of the study. All content generated was reviewed and verified by the authors, who take full responsibility for the final submission.

References

[1] Rutkowski, L., Jaworski, M., & Duda, P. (2020). *Stream Data Mining: Algorithms and Their Probabilistic Properties* (Vol. 56). Springer International Publishing. https://doi.org/10.1007/978-3-030-13962-9

[2] Bakhshi, S., Ghahramanian, P., Bonab, H., & Can, F. (2023). A Broad Ensemble Learning System for Drifting Stream Classification. *IEEE Access*, *11*, 89315–89330. https://doi.org/10.1109/ACCESS.2023.3306957

[3] Khezri, S., Tanha, J., & Samadi, N. (2024). An experimental review of the ensemble-based data stream classification algorithms in non-stationary environments. *Computers and Electrical Engineering*, *118*, 109420. https://doi.org/10.1016/j.compeleceng.2024.109420

[4] Mulimani, D., Patil, P., Totad, S., & Benni, R. (2024). Online Detection and Adaptation of Concept Drift in Streaming Data Classification. *Procedia Computer Science*, *235*, 2803–2811. https://doi.org/10.1016/j.procs.2024.04.265

[5] Eisa, A., El-Rashidy, N., Alshehri, M., El-Bakry, H., & Abdelrazek, S. (2021). Incremental Learning Framework for Mining Big Data Stream. *Computers, Materials & Continua*, *71*, 2901–2921. https://doi.org/10.32604/cmc.2022.021342

[6] Zaman, E. A. K., Mohamed, A., & Ahmad, A. (2022). Feature selection for online streaming high-dimensional data: A state-of-the-art review. *Applied Soft Computing*, *127*, 109355. https://doi.org/10.1016/j.asoc.2022.109355

[7] Wu, Z., Wang, H., Guo, J., Yang, Q., & Shao, J. (2024). Learning evolving prototypes for imbalanced data stream classification with limited labels. *Information Sciences*, *679*, 120979. https://doi.org/10.1016/j.ins.2024.120979

[8] Singh, C., & Sharma, A. (2023). A review of online supervised learning. *Evolving Systems*, *14*(2), 343–364. https://doi.org/10.1007/s12530-022-09448-y

[9] Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfharinger, B., Holmes, G., & Abdessalem, T. (2017). Adaptive random forests for evolving data stream classification. *Machine Learning*, *106*(9), 1469–1495. https://doi.org/10.1007/s10994-017-5642-8

[10] Gomes, H. M., Read, J., & Bifet, A. (2019). *Streaming Random Patches for Evolving Data Stream Classification* (p. 249). https://doi.org/10.1109/ICDM.2019.00034

[11] Gomes, H. M., Read, J., Bifet, A., & Durrant, R. J. (2021). Learning from evolving data streams through ensembles of random patches. *Knowledge and Information Systems*, *63*(7), 1597–1625. https://doi.org/10.1007/s10115-021-01579-z

[12] Domingos, P., & Hulten, G. (2000). Mining high-speed data streams. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 71–80. https://doi.org/10.1145/347090.347107

[13] Oza, N. C., & Russell, S. J. (2001). Online Bagging and Boosting. *International Workshop on Artificial Intelligence and Statistics*, 229–236. https://proceedings.mlr.press/r3/oza01a.html

[14] Aguiar, G., Krawczyk, B., & Cano, A. (2023). A survey on learning from imbalanced data streams: Taxonomy, challenges, empirical study, and reproducible experimental framework. *Mach. Learn.*, *113*(7), 4165–4243. https://doi.org/10.1007/s10994-023-06353-6

[15] Din, S., Kumar, J., Shao, J., Cobbinah, B., & Ndiaye, W. (2021). Learning High-Dimensional Evolving Data Streams With Limited Labels. *IEEE Transactions on Cybernetics*, *PP*, 1–12. https://doi.org/10.1109/TCYB.2021.3070420

[16] Onuki, E. K. T., Malucelli, A., & Barddal, J. P. (2023). A Tool for Measuring Energy Consumption in Data Stream Mining. In M. C. Naldi & R. A. C. Bianchi (Eds.), *Intelligent Systems* (pp. 429–440). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-45392-2_28

[17] Boiko Ferreira, L. E., Murilo Gomes, H., Bifet, A., & Oliveira, L. S. (2019). Adaptive Random Forests with Resampling for Imbalanced data Streams. *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–6. https://doi.org/10.1109/IJCNN.2019.8852027

[18] Loezer, L., Enembreck, F., Barddal, J. P., & De Souza Britto, A. (2020). Cost-sensitive learning for imbalanced data streams. *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 498–504. https://doi.org/10.1145/3341105.3373949

[19] Wu, Y., Liu, L., Yu, Y., Chen, G., & Hu, J. (2025). Online ensemble learning-based anomaly detection for IoT systems. *Applied Soft Computing*, *173*, 112931. https://doi.org/10.1016/j.asoc.2025.112931

[20] Paim, A. M., & Enembreck, F. (2025). Adaptive random tree ensemble for evolving data stream classification. *Knowledge-Based Systems*, *309*, 112830. https://doi.org/10.1016/j.knosys.2024.112830

[21] Esteban, A., Cano, A., Ventura, S., & Zafra, A. (2025). Simultaneous fault prediction in evolving industrial environments with ensembles of Hoeffding adaptive trees. *Applied Intelligence*, *55*(13), 930. https://doi.org/10.1007/s10489-025-06786-7

[22] Palli, A. S., Jaafar, J., Md Saad, M. H., Mokhtar, A. A., Gomes, H. M., Soomro, A. A., & Gilal, A. R. (2025). Smart adaptive ensemble model for multiclass imbalanced nonstationary data streams. *Scientific Reports*, *15*(1), 21140. https://doi.org/10.1038/s41598-025-05122-w

[23] Guo, H., Wu, Z., Liu, Z., Zhang, S., & Wang, W. (2025). Adaptive interactive network component ensemble for streaming data with concept drift. *Data Mining and Knowledge Discovery*, *39*. https://doi.org/10.1007/s10618-025-01125-3

[24] Beshah, Y., Abebe, S., & Mulugeta, H. (2024). Drift Adaptive Online DDoS Attack Detection Framework for IoT System. *Electronics*, *13*, 1004. https://doi.org/10.3390/electronics13061004

[25] Cano, A., & Krawczyk, B. (2022). ROSE: Robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams. *Machine Learning*, *111*(7), 2561–2599. https://doi.org/10.1007/s10994-022-06168-x

[26] Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, *52*(4), 1289–1306. https://doi.org/10.1109/TIT.2006.871582

[27] Manapragada, C., Webb, G., & Salehi, M. (2018). *Extremely Fast Decision Tree* (No. arXiv:1802.08780). arXiv. http://arxiv.org/abs/1802.08780

[28] Benllarch, M., El Hadaj, S., & Benhaddi, M. (2023). Improving McDiarmid Tree Performance for Predicting Heart Disease from Data Streams with Missing and Meaningless Values. *International Journal on Artificial Intelligence Tools*, *32*(06), 2350029. https://doi.org/10.1142/S021821302350029X

[29] Li, Z., Huang, W., Xiong, Y., Ren, S., & Zhu, T. (2020). Incremental learning imbalanced data streams with concept drift: The dynamic updated ensemble algorithm. *Knowledge-Based Systems*, *195*, 105694. https://doi.org/10.1016/j.knosys.2020.105694

[30] Zhang, K., Zhang, T., & Liu, S. (2023). A novel ensemble framework driven by diversity and cooperativity for non-stationary data stream classification. *Data Knowl. Eng.*, *148*(C). https://doi.org/10.1016/j.datak.2023.102232

[31] Sun, J., Du, M., Lew, Z., & Dong, Y. (2024). TWStream: Three-Way Stream Clustering. *IEEE Transactions on Fuzzy Systems*, *32*(9), 4927–4939. https://doi.org/10.1109/TFUZZ.2024.3369716

[32] Blockeel, H., Devos, L., Frénay, B., Nanfack, G., & Nijssen, S. (2023). Decision trees: From efficient prediction to responsible AI. *Frontiers in Artificial Intelligence*, *6*, 1124553. https://doi.org/10.3389/frai.2023.1124553

[33] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2019). Learning under Concept Drift: A Review. *IEEE Transactions on Knowledge and Data Engineering*, *31*(12), 2346–2363. https://doi.org/10.1109/TKDE.2018.2876857

[34] Mulimani, D., Totad, S., Patil, P., & Seeri, S. (2021). *Adaptive Ensemble Learning with Concept Drift Detection for Intrusion Detection* (pp. 331–339). https://doi.org/10.1007/978-981-16-0171-2_31

[35] A.Mehdi, O., Ali, N., Pardede, E., Alazab, A., Alquraishi, T., & Das, B. (2024). Roadmap of Concept Drift Adaptation in Data Stream Mining, Years Later. *IEEE Access*, *PP*, 1–1. https://doi.org/10.1109/ACCESS.2024.3358817

[36] Fahy, C., Yang, S., & Gongora, M. (2023). Scarcity of Labels in Non-Stationary Data Streams: A Survey. *ACM Computing Surveys*, *55*(2), 1–39. https://doi.org/10.1145/3494832

[37] Bifet, A., & Gavaldà, R. (2009). Adaptive Learning from Evolving Data Streams. *Advances in Intelligent Data Analysis VIII*, 249–260. https://doi.org/10.1007/978-3-642-03915-7_22

[38] Manapragada, C., Salehi, M., & Webb, G. I. (2022). Extremely Fast Hoeffding Adaptive Tree. *2022 IEEE International Conference on Data Mining (ICDM)*, 319–328. https://doi.org/10.1109/ICDM54844.2022.00042

[39] Cassales, G., Gomes, H. M., Bifet, A., Pfahringer, B., & Senger, H. (2023). Balancing Performance and Energy Consumption of Bagging Ensembles for the Classification of Data Streams in Edge Computing. *IEEE Transactions on Network and Service Management*, *20*(3), 3038–3054. IEEE Transactions on Network and Service Management. https://doi.org/10.1109/TNSM.2022.3226505

[40] Gunasekara, N., Pfahringer, B., Gomes, H., & Bifet, A. (2024). Gradient boosted trees for evolving data streams. *Machine Learning*, *113*(5), 3325–3352. https://doi.org/10.1007/s10994-024-06517-y

[41] Jiang, X., Jiang, Y., Liu, L., Xia, M., & Jiang, Y. (2024). Time dimension feature extraction and classification of high-dimensional large data streams based on unsupervised learning. *Journal of Computational Methods in Sciences and Engineering*, *24*(2), 835–848. https://doi.org/10.3233/JCM-237085

[42] Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79. https://doi.org/10.1016/j.neucom.2017.11.077

[43] Rong, M., Gong, D., & Gao, X. (2019). Feature Selection and Its Use in Big Data: Challenges, Methods, and Trends. *IEEE Access*, *7*, 19709–19725. https://doi.org/10.1109/ACCESS.2019.2894366

[44] You, D., Li, R., Sun, M., Ou, X., Liang, S., & Yuan, F. (2020). Online Markov Blanket Discovery With Streaming Features. *2020 IEEE International Conference on Knowledge Graph (ICKG)*, 92–99. https://doi.org/10.1109/ICBK50248.2020.00023

[45] Gomes, H. M., Read, J., Bifet, A., Barddal, J. P., & Gama, J. (2019). Machine learning for streaming data: State of the art, challenges, and opportunities. *ACM SIGKDD Explorations Newsletter*, *21*(2), 6–22. https://doi.org/10.1145/3373464.3373470

[46] Mutambik, I. (2024). An Entropy-Based Clustering Algorithm for Real-Time High-Dimensional IoT Data Streams. *Sensors*, *24*(22), Article 22. https://doi.org/10.3390/s24227412

[47] Fatlawi, H. K., & Kiss, A. (2023). Efficiency improvement of adaptive random forest using principal component analysis for mining data stream. *Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae. Sectio Computatorica*, *55*, 39–48. https://doi.org/10.71352/ac.55.039

[48] Bahri, M., Gomes, H. M., Bifet, A., & Maniu, S. (2020). CS-ARF: Compressed Adaptive Random Forests for Evolving Data Stream Classification. *IJCNN 2020 - International Joint Conference on Neural Networks*, 1–8. https://doi.org/10.1109/IJCNN48605.2020.9207188

[49] Wang, Y., Chen, Y., Zhao, Y., & Liu, S. (2024). Compressed Sensing for Biomedical Photoacoustic Imaging: A Review. *Sensors*, *24*(9), Article 9. https://doi.org/10.3390/s24092670

[50] Xu, K., Yu, J., & Gao, W. (2025). Towards privacy-preserving compressed sensing reconstruction in cloud. *Computers & Security*, *151*, 104348. https://doi.org/10.1016/j.cose.2025.104348

[51] Avros, R., Kesselman, U., & Volkovich, Z. (2024). Exploring Compressed Sensing for Proficient Cryptocurrency Mining. *Procedia Computer Science*, *246*, 4988–4997. https://doi.org/10.1016/j.procs.2024.09.456

[52] Zhang, B., Li, X., Wang, Q., Zhang, W., & Wang, K. (2024). Cipher-image compression by using compressive sensing and interweaving permutation. *Journal of King Saud University - Computer and Information Sciences*, *36*(5), 102046. https://doi.org/10.1016/j.jksuci.2024.102046

[53] Singh, H., Manikandan, M. S., & Pachori, R. B. (2024). Compressed ECG sensing based heart rate asymmetry analysis for energy-constrained fast health monitoring. *Biomedical Signal Processing and Control*, *95*, 106423. https://doi.org/10.1016/j.bspc.2024.106423

[54] Wang, S., Li, T., Nie, X., Zhao, Y., Meng, Y., Lin, C., & Qing, A. (2025). Efficient millimeter-wave compressed sensing 3D imaging for object representation. *Measurement*, *248*, 116882. https://doi.org/10.1016/j.measurement.2025.116882

[55] Baraniuk, R. (2007). Compressive Sensing [Lecture Notes]. *IEEE Signal Processing Magazine*, *24*(4), 118–121. https://doi.org/10.1109/MSP.2007.428657

[56] Heyden, M., Fouché, E., Arzamasov, V., Fenn, T., Kalinke, F., & Böhm, K. (2024). Adaptive Bernstein change detector for high-dimensional data streams. *Data Mining and Knowledge Discovery*, *38*(3), 1334. https://doi.org/10.1007/s10618-023-00999-5

[57] Kasiviswanathan, S. P., & Rudelson, M. (2019). *Restricted Isometry Property under High Correlations* (No. arXiv:1904.05510). arXiv. https://doi.org/10.48550/arXiv.1904.05510

[58] Lei Yu, Barbot, J. P., Gang Zheng, & Hong Sun. (2010). Toeplitz-structured Chaotic Sensing Matrix for Compressive Sensing. *2010 7th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP 2010)*, 229–233. https://doi.org/10.1109/CSNDSP16145.2010.5580428

[59] Montiel, J., Halford, M., Mastelini, S. M., Bolmier, G., Sourty, R., Vaysse, R., Zouitine, A., Gomes, H. M., Read, J., Abdessalem, T., & Bifet, A. (2020). *River: Machine learning for streaming data in Python* (No. arXiv:2012.04740). arXiv. https://doi.org/10.48550/arXiv.2012.04740

[60] *UCI Machine Learning Repository*. (n.d.). Retrieved May 21, 2025, from https://archive.ics.uci.edu/

[61] Weinberg, A. I., & Last, M. (2023). EnHAT — Synergy of a tree-based Ensemble with Hoeffding Adaptive Tree for dynamic data streams mining. *Information Fusion*, *89*, 397–404. https://doi.org/10.1016/j.inffus.2022.08.026

[62] Anithakumari, T., & Mishra, S. (2025). OASIS: Online adaptive ensembles for drift adaptation on evolving IoT data streams. *Internet of Things*, *31*, 101545. https://doi.org/10.1016/j.iot.2025.101545

[63] Nowak Assis, D., Barddal, J. P., & Enembreck, F. (2025). Behavioral insights of adaptive splitting decision trees in evolving data stream classification LAST. *Knowledge and Information Systems*, *67*(7), 5751–5782. https://doi.org/10.1007/s10115-025-02395-5

[64] Lu, P., Lu, J., Liu, A., & Zhang, G. (2024). *Early Concept Drift Detection via Prediction Uncertainty* (No. arXiv:2412.11158). arXiv. https://doi.org/10.48550/arXiv.2412.11158