IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 24 January 2023, accepted 8 February 2023, date of publication 14 February 2023, date of current version 22 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3245055

RESEARCH ARTICLE

A Comparative Study of Situation Awareness-Based Decision-Making Model Reinforcement Learning Adaptive Automation in Evolving Conditions

RENATO D. COSTA[1], (Member, IEEE), CELSO M. HIRATA[1], AND VICTOR U. PUGLIESE[2]
[1]Department of Computer Science, Instituto Tecnológico de Aeronáutica, São José dos Campos, São Paulo 12228-900, Brazil
[2]Institute of Science and Technology, Universidade Federal de Sao Paulo, São José dos Campos, São Paulo 12247-014, Brazil

Corresponding author: Renato D. Costa (renato.d.costa@gmail.com)

This work was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) foundation from the Ministério da Educação (MEC) of the Brazilian Government.

ABSTRACT Situation-awareness-based decision-making (SABDM) models constructed using cognitive maps and goal-direct task analysis techniques have been successfully used in decision support systems in safety-critical and mission-critical environments such as air traffic control and electrical energy distribution. Reinforcement learning (RL) and other machine learning techniques are used to automate situational awareness mental model parameter adjustments, reducing expert work on the initial configuration and long-term maintenance without affecting the mental model's structure and maintaining the situation-awareness-based decision-making model's cognitive and explainability characteristics. Real-world models should evolve to cope with changes in the environmental conditions. This study evaluates the application of reinforcement learning as an online adaptive technique to adjust the situation-awareness mental model parameters under evolving conditions, a technique we named SABDM/RL. We conducted evaluation experiments using real-world public datasets to compare the performance of the SABDM/RL technique with that of other adaptive machine learning methods under distinct concept drift-evolving conditions. We measured the overall and dynamic performances of these techniques to understand how well they adapt to evolving environmental conditions. The experiments show that the SABDM/RL performs similarly to modern online adaptive machine learning classification methods with the support of concept drift detection techniques while maintaining the mental model strength of the situation-awareness-based decision-making systems.

INDEX TERMS Adaptive systems, artificial intelligence, decision support systems, evolving behavior, explainable artificial intelligence, reinforcement learning, situation awareness.

I. INTRODUCTION

An essential aspect for an agent to act intelligently and make effective decision making in an environment is assessing the situation or situation awareness [1]. The Situation-awareness-based decision-making (SABDM) model has been successfully used to support pilots and air traffic controllers in their operational decisions because of the cognitive, explainable, and situation-awareness-based information offered to them, significantly reducing human-based errors in aeronautic accidents [2]. Endsley [3], [4] developed a situation-awareness-based decision-making model based on mental models and cognitive task-analysis techniques. Model usage extends to decision-support systems in various critical applications, including autonomous vehicles [5], [6] and on-orbit spacecraft decision-making software [7].

Machine learning (ML) techniques such as hill climbing, evolutionary genetics, and reinforcement learning (RL) are used as automation methods to reduce expert work in parameter configuration and adjustments of situation-awareness (SA) mental model belief network implementations [8], [9]. They automate the treatment of experimental and simulated

The associate editor coordinating the review of this manuscript and approving it for publication was Wanqing Zhao.

data captured by techniques such as the Situation Awareness Global Assessment Technique (SAGAT) [10] and Simulation Awareness Rating Technique (SART) [11] on the SA model's parameter adjustments.

However, these analyses are both lengthy and complex [12]. Automation helps reduce specialists' manual work effort with a significant reduction in completion time. This study aimed to overcome this limitation by demonstrating the possibility of reducing this time. By reducing the configuration time and effort, we expect the SABDM model to extend its usage and benefits to more cost-sensitive areas such as traffic management and personal care.

The literature shows that the automation of the SABDM model using machine learning techniques presents good results in model parameter configuration [8], [13] and goal selection [9]. This study extends this knowledge by automating the initial model parameter adjustments and evolutionary maintenance under environmentally changing conditions using reinforcement-learning techniques. This study evaluates the performance of reinforcement learning automation under evolving controlled conditions, including incremental, gradual, abrupt, and reoccurring conditions. It uses overall metrics to evaluate the expected nominal classification performance, and dynamic metrics to evaluate the dynamic response to evolving conditions.

The remainder of this paper is organized as follows. Section II presents background techniques, including situation-awareness-based decision-making models, online incremental adaptive learning, concept drift detection, and one-step online reinforcement learning. Section III presents a one-step online reinforcement learning technique for adjusting the belief network parameters of the SA mental model. Section IV presents the results and evaluates the performance of the adaptive machine and reinforcement learning classification methods under concept drift conditions. Section V discusses the results, conclusions, and proposals for future work.

II. BACKGROUND TECHNIQUES

A. SITUATION AWARENESS DECISION-MAKING MODEL

Zhou et al. [14] advocate situation awareness as a key element in decision-making processes for autonomous ship navigation. Shen et al. [7] reported that the spacecraft's on-orbit mission success relies on knowledge of the situation for optimal decision making. The situation-awareness-based decision-making model directly deals with the cognitive process of human beings in determining their selections in complex everyday situations [15]. Endsley [3] models situation-awareness decision-making at three levels: perception, comprehension, and projection. Perception level concerns the perception of the environment, including the collection of relevant information. Comprehension level refers to understanding the data and information related to achieving the objectives. The projection level represents the projection of the system's future functioning and environment, allowing proactive action to achieve goals. The model uses mental models to maintain knowledge, allowing automatization and information-processing mechanisms to alleviate memory overload. Mental models provide a systematic understanding of how something works, representing an individual's knowledge or beliefs [16], [17]. Cognitive maps are powerful graphical mental models that represent symbolic information processes of the human mind [18]. Cognitive task analysis techniques are used to comprehend the cognitive processes [19]. Goal-Directed Task Analysis (GDTA) is a cognitive task analysis technique that focuses on the goals and decisions that the operator must accomplish to perform a job successfully, clarifying situation awareness in particular domains [2] such as maritime navigation [20] and fire emergency response [21].

B. ONLINE INCREMENTAL AND ADAPTIVE LEARNING TECHNIQUES

Incremental learning represents a family of machine learning methods in which a learner tackles a predictive (or any decision-making) task by continuously learning from a sequence of data instances while maintaining the most previously learned knowledge [22]. Incremental learning is called online learning when learning is performed at each new event or decision point. Incremental learning aims to maximize the accuracy of the sequence of predictions/ decisions made by the learner, given the knowledge of correct answers to previous prediction/decision tasks and possibly additional information. In contrast, traditional batch or offline machine learning methods learn a model from the entire training data set at once, with new learning representing, most of the time, a catastrophic forgetting of the previous learning experience [23]. Incremental learning reduces the memory footprint as past learning data can be dismissed, which is very useful in applications with a large amount of data, limiting memory size, processing requirements, and training time. It is the choice for learning from continuous data streams with large amounts of data and concept drift conditions in many real-world applications, such as the Internet of Things (IoT) and 5G cellular network traffic stream analysis [24]. Batch machine-learning methods typically use a hold-out procedure for training and testing [25]. This procedure separates part of the dataset for training and part for testing. Batch solutions often use k-fold cross-validation to obtain an average result, which typically requires a large amount of memory and a high-performance processor. Incremental methods often use predictive sequential or prequential procedures as evaluation strategies [26]. The prequential procedure first uses a new instance to predict and test the predictive model and then considers it for training. The predictive model is incrementally updated while the instances are processed.

An adaptive algorithm can learn a predictive model from an evolving data stream and detect distributional changes or concept drifts [27]. The connection of incremental online learning with concept drift detection techniques and forgetting functions provides adaptive properties for machine learning methods. Adaptation is also provided by directly applying concept drift techniques to batch machine learning methods by full retraining at each drift detection point.

The types of learning tasks and feedback information classify existing online learning works into three major categories: (i) supervised online learning, where complete feedback information is always available, (ii) online learning with limited feedback, and (iii) unsupervised online learning, where no feedback is available [28]. The multi-armed bandit reinforcement learning algorithm is an example of an online learning algorithm that operates with full or limited feedback.

C. CONCEPT DRIFT DETECTION TECHNIQUES

When dealing with evolution or streaming data, we can expect to observe changes in the behavior of environmental conditions and their relationships with the target variables predicted by the applications.

Concept drift occurs when the statistical properties change over time [29], reflecting changes in the joint distribution of the target variable X and the environment or class y such that $P(X, y) = P(X)P(y|X)$. Consequently, the predictions of models trained in the past may become less accurate. The change can have three concept drift sources: (i) change in the prior probability of the target variable $P_{t+1}(X) \neq P_t(X)$, (ii) change in the class-conditional probability of the target variable $P_{t+1}(y|X) \neq P_t(y|X)$, and (iii) changes in the target variable's prior and class-conditional probability [30].

The drifts may manifest in distinct forms: (i) incremental: the concepts change incrementally from the original conditions to the final ones, (ii) sudden or abrupt drift: the changes occur in a short time, (iii) gradual: the new concepts are introduced gradually until a complete change occurs, and (iv) reoccurring: the changes occur on a reoccurring basis. Noise and single-spike changes are considered anomalies and are not treated as concept drifts. To handle concept drifts and manage their effects, the learning models and classification methods require continuous performance diagnostic mechanisms to detect, understand, and adapt to changes in data over time. These detection mechanisms are referred to as concept drift detection methods.

The Scikit-Multiflow Python library [31] provides multiple state-of-the-art concept-drift detection methods, including DDM: Drift Detection Method [32], EDDM: Early Drift Detection Method [33], HDDMa and HDDMw: Drift detection methods based on Hoeffding's bound [34], ADWIN: Adaptive Windowing [35], KSWIN: Kolmogorov-Smirnov Windowing based on the Kolmogorov-Smirnov test [36], and Page Hinkley, which uses the Page-Hinkley test to indicate when the observed values differ considerably from their previous values [37].

D. ONE-STEP, ONLINE REINFORCEMENT LEARNING

Reinforcement Learning (RL) methods deal with sequential decision-making problems by learning from the environment's interactions after a sequence of events with a controller agent making decisions to optimize a given notion of cumulative rewards [38]. Reinforcement-learning controller agents do not require complete knowledge or control of the environment. They use their interaction with the environment to collect information through trial and error or practical experience to learn the behavior incrementally with maximum returns. Reinforcement learning comes from the dynamic system theory, specifically optimal control, and the Markov Decision Process (MDP), which describes a fully observable environment in which the current state completely characterizes the process [38]. A state has the Markov property if and only if it captures and retains all the relevant historical information. The future depends only on the present state and not on past information [39].

To work with situation awareness applications, we require immediate decisions as provided by Multi-Armed Bandits, a one-step multiple-decision reinforcement learning solution [40], [41]. The Contextual Multi-Armed Bandit (C-MAB) configuration considers the environmental situation at each step as a context description [42], as used in online personalized recommendation systems [43], [44]. The belief network parameters gather the underlying probabilities behind the context's event-to-action in the situation awareness model. The probability model constitutes the information state of a Bayes-Adaptive MDP, summarizing all the information accumulated thus far from the event statistics history. The information state-augmented Bayes-adaptive MDP is the basis for the Bayes-Adaptive Contextual Multi-Armed Bandit (BA-C-MAB) algorithm [45], [46] used in this study.

E. RELATED WORKS

The literature presents automation and enhancements to situation-awareness-based decision-making applications by using distinct techniques. Gini et al. [13] employed genetic algorithms to automatically learn the mental or cognitive map (belief network) parameters of a situation awareness model. Koopmanschap et al. [8] evaluated hill-climbing and evolutionary genetic algorithms. Both studies show promising results in tailoring belief network parameters in simulated air force combat operations. D'Aniello et al. [9] used reinforcement learning in a situation awareness framework for adaptive goal selection in a fleet management case study. They proposed an adaptive goal-selection (AGS) approach using goal-driven and data-driven information processing. The system learns how to suggest goals by exploring how users and operators react to suggestions in specific states, using the goal vector as a system state. Their work compared the standard situation awareness solution with the enhanced AGS-SA proposal using an ε-greedy Q-learning reinforcement learning algorithm with environmental stimuli in distinct simulated scenarios. Using reinforcement learning techniques to adapt to evolving scenarios, D'Aniello et al. [9] studied the cognitive map configuration issue using an external desirability function to train and adapt the active goal orientation of the

AGS algorithm. Loia et al. [47] and Gaeta et al. [48] proposed techniques to support Situation Awareness solutions using granular computing. Information granulation is a common activity that humans carry out with the intent of better understanding a problem, acting as an abstraction mechanism that reduces the conceptual load to comprehend the problem and offers better insight into situation awareness models. However, these techniques do not provide automation, as used in this study.

Adaptive learning based on online or incremental learning and concept drift techniques is used to deal with streaming and evolving conditions and where there is the influence of human actuation as expected for situation-awareness application environments. Agrahari and Singh [49] showed that the classifier accuracy deteriorates because of concept drift under evolving conditions. Traditional batch classifiers are not expected to learn patterns in nonstationary data distributions, and concept drift detectors are necessary to address the evolving conditions. Shahraki et al. [24] investigated and compared online techniques in the networking domain, highlighting the advantages of online learning and the challenges associated with online-based network traffic stream analysis. Vakili and Rezaei [50] analyzed the performance of online incremental algorithms applied to the real-time prediction of human physical movements using a human activity recognition (HAR) approach. They show that as the style of activities made by a person typically changes, the patterns of performing activities vary from person to person, with incremental methods attaining consistently better performance than offline batch methods. Nallaperuma et al. [51] proposed an intelligent traffic management platform to capture the dynamic patterns in traffic data streams. This solution supports traffic signal control decisions based on real-time data streaming from IoT devices using an online machine learning approach to overcome the dynamic nature of continuously changing patterns and concept drifts in the traffic environment.

III. SITUATION AWARENESS-BASED DECISION-MAKING MODEL AUTOMATION

Decision-making is a human cognitive process that determines rational, heuristic, and intuitive selections in complex situations and standard procedures [15]. The situation-awareness-based decision-making model directly deals with the cognitive perspective. Endsley [3] models the situation-awareness requirements levels of perception, comprehension, and projection using mental models and cognitive task analysis techniques in its construction. Mental models are graphically represented by cognitive maps with belief network connections to address reasoning under uncertainty [52] inherent in the decision-making domain.

Machine learning techniques have been used to automate the configuration of situation-awareness-based decision-making model belief parameters. Typically, algorithms use batches of past actual or simulated data for the initial configuration automation to learn the best model configuration. Evolutionary maintenance configuration automation can be achieved in two ways: online and adaptive machine learning methods with a forgetting mechanism of past information. Alternatively, concept drift detection methods are used to detect changes in environmental conditions to trigger new training in batch machine-learning methods.

The SABDM/RL technique uses the Bayes-Adaptive Contextual Multi-Armed Bandit (BA-C-MAB) reinforcement learning algorithm as an online and adaptive machine learning method for adjusting the situation-awareness decision-making model belief parameters. Situation-awareness applications identify situations or contexts for decision making. This method builds a probabilistic model that summarizes all the information accumulated thus far, from the historical statistics of the event to the action. The probabilistic model is used to automate the belief parameter learning of cognitive maps that support situation-awareness-based decision-making problems with incremental or temporal difference learning.

A. BA-C-MAB REINFORCEMENT LEARNING

The Multi-Armed Bandit (MAB) problem can be described as a sequential decision model of one of N independent decisions, a *Bandit arm*, assigned to one of N possible treatments or actions [40]. The MAB is modeled as a tuple $\langle A, R, \pi \rangle$ where A is the set of decisions or actions a, R is the set of rewards r, and π is an action policy. At trial t, based on policy π, the agent selects action $a \in A$ bringing out a reward $r \in R$. The main goal of policy π is to maximize the accumulated reward returned.

Regret is defined as the difference between the maximum or optimal reward r^* and actual reward r from the environment after the applied action. Maximizing the accumulated reward minimizes the accumulated regret:

$$max \sum_{t=1}^{T} r_t \equiv min \sum_{t=1}^{T} \left(r_t^* - r_t\right) \tag{1}$$

where r_t is the reward and $r_t^* - r_t$ the regret at trial t

The Contextual Multi-Armed Bandit (C-MAB) is constructed from the MAB by applying context information to the action decision [42]. The C-MAB introduces the situation context concept s^e corresponding to the observed environmental state. The C-MAB is modeled as a tuple $\langle s^e, A, R, \pi \rangle$ where $S^e = \mathbb{P}[s^e]$ is an unknown probability distribution over context s^e, $R_s^a = \mathbb{P}[r|s^e, a]$ is an unknown probability distribution over rewards, and π is an action policy. At each trial t, the environment generates the context state $s_t^e \in S^e$, the user agent selects an action $a_t \in A$ based on the action policy π, and the environment generates a reward $r_{t+1} \in R_s^a$.

We augment the one-step decision-making C-MAB model as a sequential decision-making problem, including at each step t the information state $\tilde{s}$ summarizing the historical statistics of all information accumulated so far. The information state has the Markov property because it does not depend on the past information. The augmented state is $s \overset{\text{def}}{=} \langle s^e \tilde{S} \rangle$ where $s^e \in S^e$ is an environmental situation and $\tilde{s}$ is the historical statistics of s^e. The augmented state constructs the Bayes-Adaptive Markov Decision Process (BAMDP) [53] $\tilde{M}$ defined by the tuple $\langle S, A, \tilde{P}, R, \gamma \rangle$ as the basis for the information state augmented Bayes Adaptive C-MAB (BA-C-MAB) algorithm where $S \stackrel{\text{def}}{=} \left\langle S^e, \tilde{S} \right\rangle$ is the augmented state, S^e is the set of possible environmental situations, $\tilde{S}$ is the information state, and A is the set of possible actions a. Each action a causes a transition from the agent information state $\tilde{s}$ to a new information state $\tilde{s}'$ by adding new information with probability $\tilde{P}\left(\tilde{s}' \mid \tilde{s}, s^e, a\right)$, the conditional distribution for the next information state $\tilde{s}'$ given the current information state $\tilde{s}$, the environmental state s^e, and action a. R is the set of possible environment's rewards to actions or reward set. $U(R)$ is a utility function over rewards and γ is the discount factor for future rewards. The Markov property is warranted because the new context and action do not depend on historical information, with the current information state encapsulating all necessary historical statistics.

SABDM/RL is based on the information state-augmented Bayes Adaptive C-MAB (BA-C-MAB) reinforcement learning algorithm. It is a one-step, adaptive *Policy-based*, *Model-free* agent type, with the SA-GDTA decision tree working as a policy for mapping the environmental event context to action, SA requirements forming the environmental state s^e, belief parameters of the SA-GDTA decision tree, and information state $\tilde{s}$.

For the Bayesian evolution of the information state, we use the *Temporal-Difference SARSA* (TD-SARSA) learning technique, updating the $Q(S, A)$ state-action value function [38]. Accordingly, the Bellman equation [54] describes the incremental update of Q as:

$$Q_{t+1} = Q_t + \delta_t \tag{2}$$

where:

$$\delta_t = U(R_{t+1}) + \gamma Q\left(S_{t+1}, A_{t+1}\right) - Q\left(S_t, A_t\right) \tag{3}$$

is the error or temporal difference.

To address nonstationary problems or evolutionary conditions, we assign more weight to recent rewards than to long-past ones, using the step-size parameter α as the learning rate. The Bellman equation, including the step-size parameter α becomes:

$$Q_{t+1} = Q_t + \alpha\delta_t \tag{4}$$

where α is the learning rate (step-size parameter).

Note that the discount rate γ in Equation (3) applies to future rewards, and the learning rate α in Equation (4) applies to the temporal difference, creating an exponential vanishing effect for past accumulated errors on the information state of $1 - \alpha$.

This process constructs an endless sequential one-step episodic MDP as the basis of the BA-C-MAB algorithm. Considering the one-state situation with no future reward, zeroing the discount rate γ in the general Bellman equation, and using a learning rate $\alpha < 1$ for convergence, we describe the temporal difference equation to update the state-action value function $Q\left(S_t, A_t\right)$ for nonstationary problems as:

$$Q_{t+1} = Q_t + \alpha\delta_t \tag{5}$$

$$\delta_t = U(R_{t+1}) - Q\left(S_t, A_t\right) \tag{6}$$

$$\begin{aligned} Q_{t+1} &= U(R_{t+1}) + (1-\alpha)\, Q\left(S_t, A_t\right) \\ Q_0 &= Pr\left(U\left(R\right)\right) = \text{any prior knowledge to } U(R) \text{ or } 0 \end{aligned} \tag{7}$$

The goal of the TD-SARSA reinforcement learning task is to learn through the environmental reward R, the SA-GDTA policy that maximizes $Q\left(S_t, A_t\right)$. The probabilistic model constructed by the information state constitutes the belief network parameters of the SA-GDTA goal-decision model, strengthening the influence of each situation awareness requirement on the decision making of the SABDM/RL agent.

B. THE SABDM/RL ALGORITHM

The SABDM/RL algorithm automates the initial configuration of cognitive maps and long-term evolving maintenance in applications with concept-drift conditions.

In the initial configuration, the method uses the existing past feedback sequentially, giving more importance to newer returns than older ones, considering the possibility of concept drifts in the training set.

In evolutionary maintenance, the method updates the belief parameters of cognitive maps in an incremental learning manner each time the applications provide new online feedback. The process captures any variation or concept drift that occurs, keeping the cognitive map parameters updated and the accuracy of the applications.

Algorithm 1 presents the SABDM/RL algorithm:

Algorithm 1 SABDM/RL Algorithm

1: For each event t observed from the environment:
2: Get the event information s_t^e and determine the situation context s_t^{eg} for agent selected goal g.
3: Run the SA-GDTA model to determine the recommended action a_t. Action a_t is the action $argmax_a Q_{s_t}^{a_t}$ that provides the greatest return Q.
4: After receiving reward r_{t+1} from the actual action, update the cognitive map belief parameters of the SA-GTDA model using the BA-C-MAB algorithm.
5: End For each

IV. SITUATION AWARENESS DECISION-MAKING CONFIGURATION AUTOMATION EVALUATION

In this section, we evaluate the application of the BA-C-MAB reinforcement learning method over the parameter configuration automation of situation-awareness-based decision-control systems using insect public datasets [55] available at the USP data stream repository (available online at https://sites.google.com/view/uspdsrepository). The insect

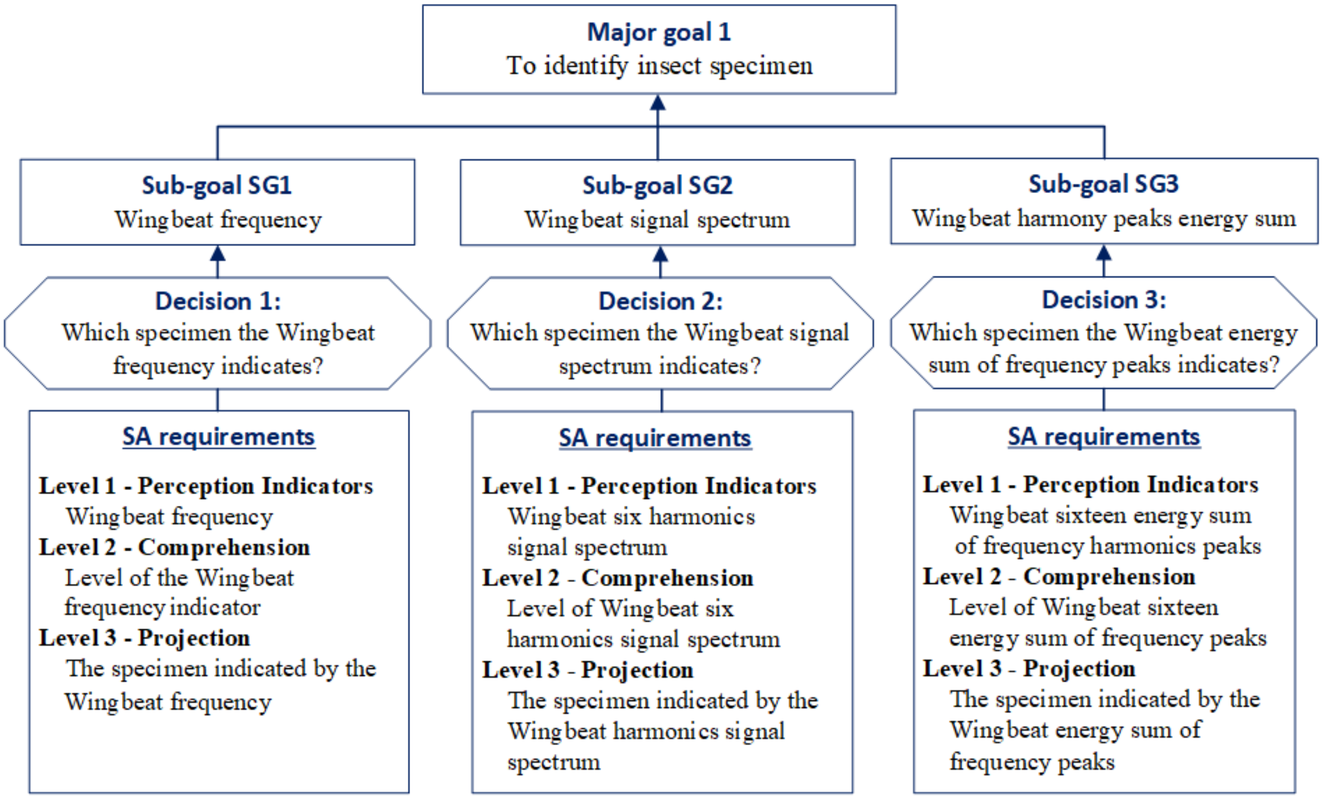


FIGURE 1. Insect database situation awareness - goal-direct task analysis.

datasets are real-world streaming datasets with known and distinct concept drift conditions designed for the benchmark evaluation of stream classifiers and concept drift detectors. The datasets were constructed using a streaming application to identify six flying insect species (male and female *Aedes aegypti, Aedes albopictus*, and *Culex quinque-fasciatus*) traveling into an intelligent trap under temperature-controlled conditions using optical sensors. The datasets have 33 features related to the wingbeat frequency, complexity measures of the signal spectrum, and the energetic sum of the observed signal peaks [55]. We used balanced datasets with the following concept drift types: (i) incremental, (ii) abrupt, (iii) incremental-gradual, (iv) incremental-abrupt-reoccurring, and (v) incremental-reoccurring.

We compared the situation-awareness-based decision-making model adjusted by the BA-C-MAB reinforcement learning algorithm (SABDM/RL) with six adaptive classification methods obtained from the Scikit-Multiflow Python library [31]: Adaptive Naïve Bayes (ANB), Adaptive Random Forest (ARF) [56], Extremely Fast Decision Tree Classifier (EFDT) [57], Hoeffding's Adaptive Tree Classifier (HAT), also referred to as Very Fast Decision Tree Classifier (VFDT) [58], Online Perception (OLP) [59], and Very Fast Decision Rules (VFDR) [60]. They all have a *partial_fit* method for online learning available from the library. We also compared SABDM/RL with XGBoost [61], a batch machine-learning method applying distinct concept drift detection techniques. XGBoost has shown the best results in various applications being a machine-learning reference method [62]. We used concept drift detection methods obtained from the Scikit-Multiflow Python library: Adaptive Windowing method for concept drift detection (ADWIN), Drift Detection Method (DDM), Early Drift Detection Method (EDDM), Drift Detection Method based on Hoeffding's bounds with moving average test (HDDM_A), Drift Detection Method based on Hoeffding's bounds with moving weighted average-test (HDDM_W), Kolmogorov-Smirnov Windowing method for concept drift detection (KSWIN), and Page-Hinkley method for concept drift detection (PageHinkley).

A. GOAL-DIRECT TASK ANALYSIS EVALUATION OF INSECT SPECIES

We constructed a situation awareness goal-direct task analysis decision-making for the SABDM model of the experiments. We established the recommendation objective, the decisions to be taken, and the goal-direct tree using three sets of features: wingbeat frequency, six signal spectra, and twenty-six harmonic signals energic sum, where each set evaluates the insect specimen as a sub-goal of the goal-direct tree.

FIGURE 1 presents the situation awareness goal-direct task analysis (SA-GDTA) tree constructed for the experiments showing the hierarchical goal structure, the sub-goals with support decisions, and the SA requirements. We normalized each requirement's comprehension

TABLE 1. Metrics of Experiment 1 with incremental drift.

Experiment 1: Incremental drift	Adaptive Naive Bayes (ANB)	Adaptive Random Forest (ARF)	Extremely Fast Decision Tree (EFDT)	Hoeffding Adaptive Tree (HAT)	Online Perception (OLP)	Very Fast Decision Rules (VFDR)	XGBoost + ADWIN	XGBoost + HDDM_W	XGBoost + KSWIN	Mean	Std Deviation	SABDM/RL
Weighted Accuracy (%)	47.53	**64.86**	57.06	53.93	53.20	52.10	60.88	64.64	63.15	57.48	5.85	60.80
Weighted F1-score (%)	46.36	64.31	55.95	52.60	53.12	50.77	61.07	**64.70**	63.26	56.90	6.29	58.97
Matthews MCC (%)	38.07	**58.01**	48.75	45.55	43.84	43.50	53.08	57.58	55.79	49.35	6.71	53.88
Minimum Accuracy (%)	35.40	36.10	38.90	38.80	38.30	33.70	45.80	48.40	**50.10**	40.61	5.62	44.80

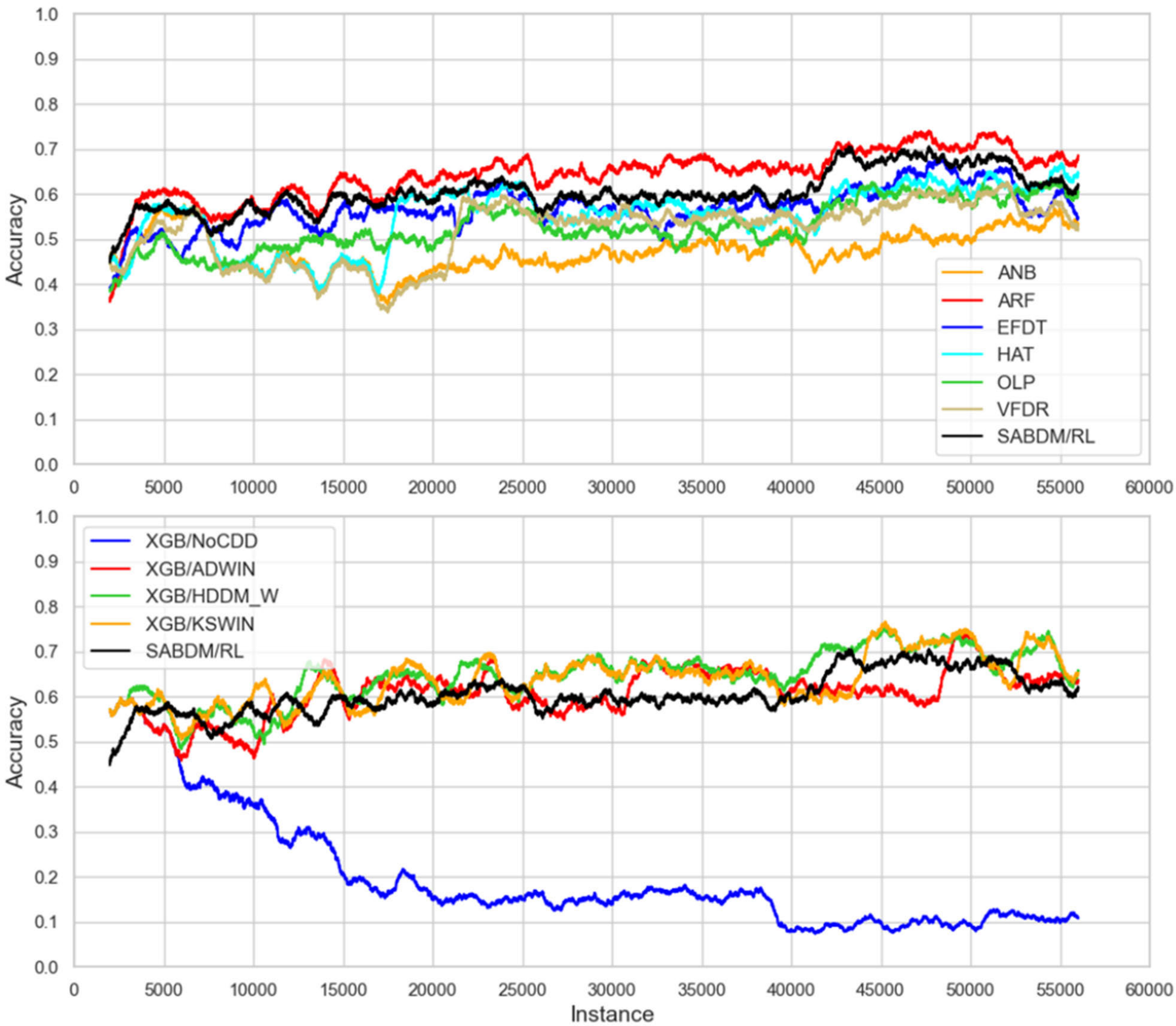


FIGURE 2. Dynamic accuracy of Experiment 1 with incremental drift.

evaluation using the requirement's mean and categorized them into nine categories using the distance from the mean based on the standard deviation.

B. THE EXPERIMENTS

Five insect-balanced datasets were used for comparison. Considering the multiclass classification objective of the

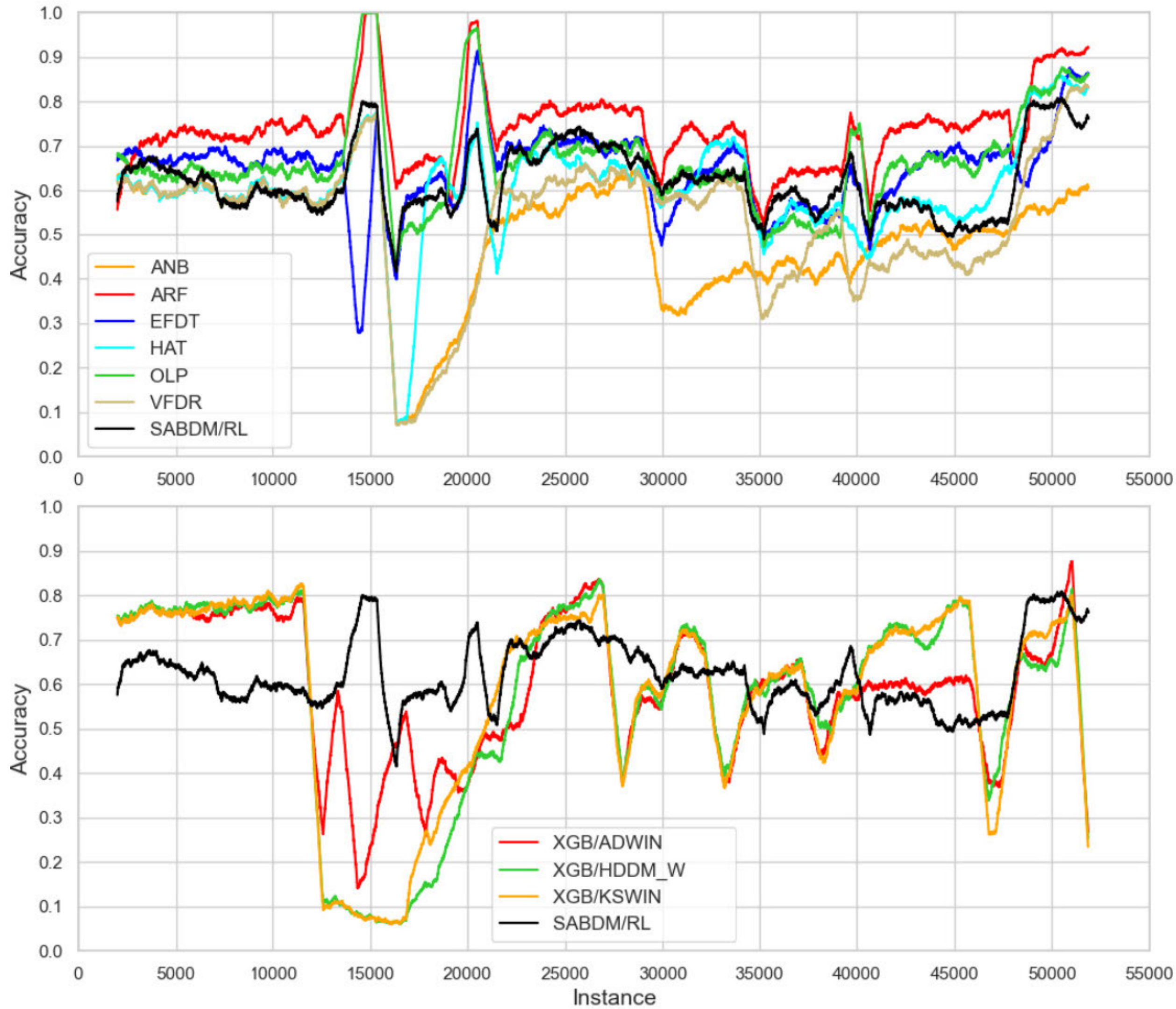


FIGURE 3. Dynamic accuracy of Experiment 2 with abrupt drift.

algorithms and the balanced characteristics of the datasets, the following metrics were selected for the experiments [63]:

(i) *Weighted average accuracy*: The weighted average accuracy is computed by taking the average accuracy of all classes weighted by the total number of instances of each class.

(ii) *Weighted average F1-score*: The F1-score represents the balance between Precision and Recall, weighted by the total number of instances of each class.

(iii) *Matthews Correlation Coefficient (MCC)*: MCC indicates the statistical correlation between the true and predicted values, being the phi-coefficient φ applied to multiclassification [64], [65].

(iv) *Minimum weighted accuracy*: The minimum weighted accuracy is used to measure the dynamics of the classification, evaluated using a moving window of 1000 events.

All metric computations discarded the first 2000 events of the datasets used for the initial training of the experiments.

Experiment 1 used the incremental balanced insect dataset. It provides 57,018 instances and six distinct insect classes, with temperatures varying from 20 °C to 40 °C incrementally. The class instances are balanced over the entire stream. TABLE 1 shows the accuracy metrics of Experiment 1 for the adaptive classifiers, XGBoost with concept drift detectors, and SABDM/RL.

XGBoost achieved the best results in all the experiments combined with the ADWIN, HDDM_W, and KSWIN concept drift detectors. Bold numbers indicate the best figures.

FIGURE 2 (top) shows the dynamic behavior of the accuracy metric for the adaptive classifiers and SABDM/RL in Experiment 1. The bottom of the figure shows the XGBoost with concept drift detectors and SABDM/RL. The blue line at the bottom of the figure represents the dynamic behavior of XGBoost without concept drift adaptation, thereby demonstrating the importance of adaptation for accuracy.

TABLE 2. Metrics of Experiment 2 with abrupt drift.

Experiment 2: Abrupt Drift	Adaptive Naive Bayes (ANB)	Adaptive Random Forest (ARF)	Extremely Fast Decision Tree (EFDT)	Hoeffding Adaptive Tree (HAT)	Online Perception (OLP)	Very Fast Decision Rules (VFDR)	XGBoost + ADWIN	XGBoost + HDDM_W	XGBoost + KSWIN	Mean	Std Deviation	SABDM/RL
Weighted Accuracy (%)	50.31	**74.97**	64.99	61.25	67.83	53.95	58.86	57.13	56.25	60.62	7.17	63.41
Weighted F1-score (%)	48.92	**74.79**	64.8	60.43	67.78	52.83	58.99	56.95	55.94	60.16	7.48	62.14
Matthews MCC (%)	41.18	**70.02**	58.15	53.93	61.40	45.30	50.68	48.65	47.61	52.99	8.43	56.61
Minimum Accuracy (%)	7.10	**52.00**	27.80	7.30	41.90	7.00	14.10	6.00	6.10	18.81	16.57	41.50

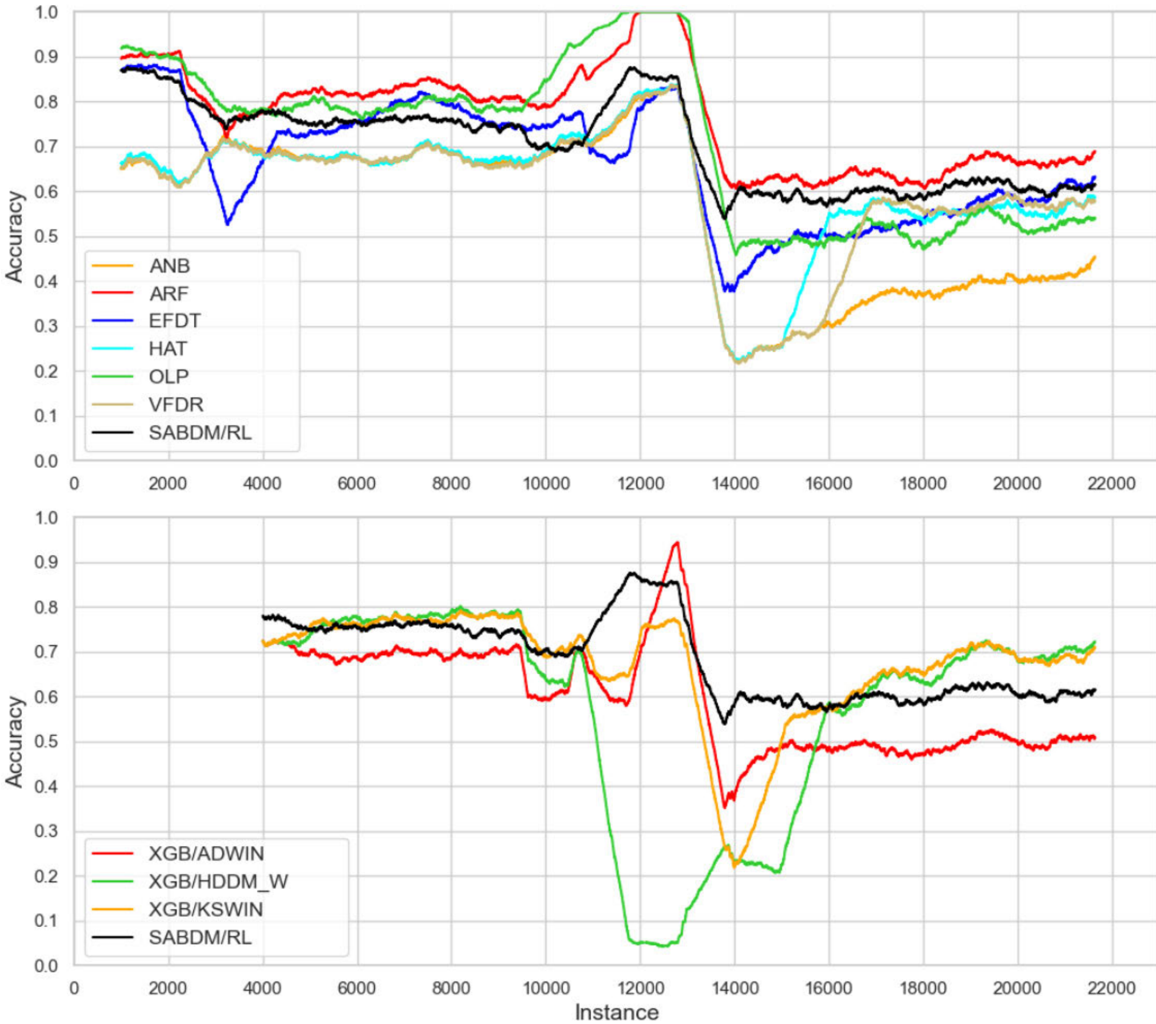


FIGURE 4. Dynamic accuracy of Experiment 3 with incremental-gradual drift.

Experiment 2 used the abrupt balanced insect dataset. It provides 52,848 instances and six distinct insect classes, with five sudden change points. The first-stream instances were collected at 30 °C. The temperature abruptly changes to 20 °C and returns to approximately 35 °C. Three additional abrupt changes occur until the end of the stream. TABLE 2

TABLE 3. Metrics of Experiment 3 with incremental-gradual drift.

Experiment 3: Incremental Gradual Drift	Adaptive Naive Bayes (ANB)	Adaptive Random Forest (ARF)	Extremely Fast Decision Tree (EFDT)	Hoeffding Adaptive Tree (HAT)	Online Perception (OLP)	Very Fast Decision Rules (VFDR)	XGBoost + ADWIN	XGBoost + HDDM_W	XGBoost + KSWIN	Mean	Std Deviation	SABDM/RL
Weighted Accuracy (%)	53.01	**77.01**	65.40	61.62	72.82	60.29	58.59	57.69	66.32	63.64	7.17	69.59
Weighted F1-score (%)	51.52	**76.37**	65.23	60.59	72.77	59.64	59.06	57.39	66.62	63.24	7.36	68.89
Matthews MCC (%)	44.03	**72.53**	58.47	54.11	67.36	52.45	50.69	49.71	59.69	56.56	8.47	63.62
Minimum Accuracy (%)	21.80	**60.60**	37.70	22.80	45.80	21.70	35.20	4.20	21.80	30.18	15.61	53.9

TABLE 4. Metrics of Experiment 4 with incremental-abrupt-reoccurring drift.

Experiment 4: Incremental Abrupt Reoccurring Drift	Adaptive Naive Bayes (ANB)	Adaptive Random Forest (ARF)	Extremely Fast Decision Tree (EFDT)	Hoeffding Adaptive Tree (HAT)	Online Perception (OLP)	Very Fast Decision Rules (VFDR)	XGBoost + ADWIN	XGBoost + HDDM_W	XGBoost + KSWIN	Mean	Std Deviation	SABDM/RL
Weighted Accuracy (%)	58.57	**75.36**	69.02	65.11	69.70	57.40	55.29	56.74	59.10	62.92	6.69	63.62
Weighted F1-score (%)	55.76	**75.08**	68.32	62.94	69.66	55.88	56.05	57.56	59.89	62.35	6.71	59.91
Matthews MCC (%)	51.15	**70.58**	63.07	58.69	63.64	49.37	46.67	48.30	51.09	55.84	7.93	57.35
Minimum Accuracy (%)	17.60	**52.90**	34.2	25.00	46.80	22.70	13.60	8.70	11.30	25.87	14.81	25.60

shows the accuracy metrics of Experiment 2 for the adaptive classifiers, XGBoost with concept drift detectors, and SABDM/RL. FIGURE 3 (top) shows the dynamic behavior of the accuracy metric for the adaptive classifiers and SABDM/RL. The bottom of the figure shows XGBoost with concept drift detectors and SABDM/RL.

Experiment 3 used the incremental-gradual balanced insect dataset. It provides 24,150 instances and six distinct insect classes, with the temperature in the first instances around 37 °C. The temperature incrementally decreases until 35 °C, followed by a period with a gradual change that intercalates the temperature values of 35 °C and 23 °C, with the time for the higher temperature diminishing until it definitively changes to 23 °C. At the end of the stream, the temperature increases incrementally to 27 °C. TABLE 3 shows the accuracy metrics of Experiment 3 for the adaptive classifiers, XGBoost with concept drift detectors, and SABDM/RL. FIGURE 4 (top) shows the dynamic behavior of the accuracy metric for the adaptive classifiers and SABDM/RL. The bottom of the figure shows XGBoost with the three best-concept drift detectors and SABDM/RL.

Experiment 4 used the incremental-abrupt-reoccurring balanced insect dataset. It provides 79,986 instances and six distinct insect classes, with three cycles of incremental temperature increases from 20 °C to 40 °C. The dataset exhibits an abrupt change between the end and beginning of each cycle of incremental changes. TABLE 4 shows the accuracy metrics of Experiment 4 for the adaptive classifiers, the three best combinations of XGBoost with concept drift detectors, and SABDM/RL. FIGURE 5 (top) shows the dynamic behavior of the accuracy metric for the adaptive classifiers and SABDM/RL. The bottom of

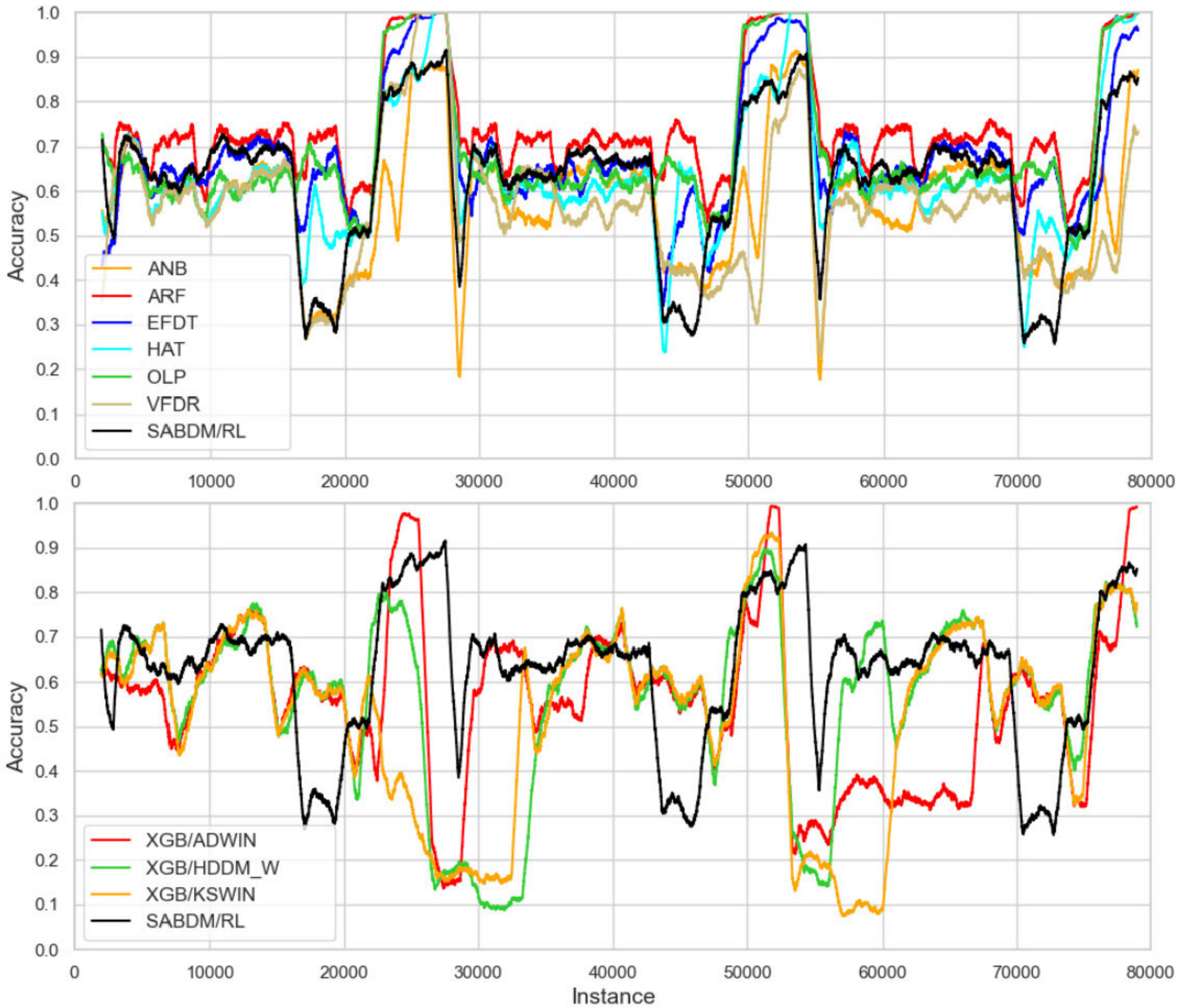


FIGURE 5. Dynamic accuracy of Experiment 4 with incremental-abrupt-reoccurring drift.

the figure shows XGBoost with concept drift detectors and SABDM/RL.

Experiment 5 used the incremental-reoccurring balanced insect dataset that provides 79,986 instances and six distinct insect classes, with three incremental temperature cycles, increasing from 20 °C to 40 °C, returning to 20 °C, and then increasing again to 40 °C. The dataset exhibits an abrupt change between the end and beginning of a cycle of incremental changes. TABLE 5 lists the accuracy metrics of Experiment 5 for the adaptive classifiers, the three best combinations of XGBoost with concept drift detectors, and SABDM/RL. FIGURE 6 (top) shows the dynamic behavior of the accuracy metric for the adaptive classifiers and SABDM/RL. The bottom of the figure shows XGBoost with concept drift and SABDM/RL.

V. DISCUSSION

This study evaluates reinforcement learning as a one-step, online, adaptive machine learning method for automatic and continuous adjustment of situation-awareness-based decision-making application's cognitive map belief parameters, named SABDM/RL. It compares the results to non-situation-awareness-based decision-making using adaptive machine learning algorithms and the XGBoost algorithm with several concept drift detection methods.

The evaluation experiments used the insect datasets, a multiclass streaming dataset with five distinct concept drift conditions: incremental, abrupt, gradual, incremental-abrupt-reoccurring, and incremental-reoccurring. The experiments evaluated six online adaptive machine learning algorithms, XGBoost with seven concept drift detection methods, and the SABDM/RL method. The evaluation compares the overall and dynamic performances of the methods. The overall weighted accuracy and weighted F1-score metrics allow an understanding of the expected performance of the methods when running an application. The dynamics of the method were measured using 1000 events moving windows. The dynamics are shown in the dynamic accuracy graphics of

FIGURE 6. Dynamic accuracy of Experiment 5 with incremental-reoccurring drift.

the experiments (FIGURES 2–6). Using the same strategy, we measured the minimum dynamic accuracy metric, which indicates the expected worst-case results for the methods under each evolving condition.

TABLES 6–9 list the experimental results. The tables show that the SABDM/RL method achieves accuracy and MCC results above the average in all drift conditions tested compared to the machine learning classification methods. The experiments also confirm that the best overall and dynamic results are obtained using the Adaptive Random Forest (ARF) algorithm as in [44].

The main strength of the SABDM/RL method is the interpretability and explainability provided by the cognitive maps and goal-direct task analysis techniques (SA-GDTA) used by the SABDM model. The method, using a Bayesian-based reinforcement learning technique, automatically adapts the parameters of the SA-GDTA goal-direct tree. The experiments show that in the direct comparison of the weighted average accuracy (TABLE 6), weighted average F1-score (TABLE 7), MCC (TABLE 8), and minimum accuracy (TABLE 9) with the Bayesian, tree-based, and rule-based methods, also considered explainable, the SABDM/RL method: (i) overpasses the adaptive Bayesian machine learning method, the Adaptive Naïve Bayes (ANB), under all drift conditions, (ii) overpasses the adaptive rule-based machine learning method (VFDR) under all drift conditions, and (iii) obtains similar results to the adaptive tree-based methods, being slightly better than Hoeffding's Adaptive Tree (HAT) and slightly worse than the Extremely Fast Decision Tree (EFDT) methods. In addition, both Adaptive Random Forest (ARF) and Online Perception (OLP) obtain better results showing that they collect the distinct drift variations of the datasets better. Compared to online adaptive methods, the SABDM/RL method performs better with incremental drift types than with abrupt drift types. In Experiment 1, with incremental drift, the method performed better in five out of

TABLE 5. Metrics of Experiment 5 with incremental-reoccurring drift.

Experiment 5: Incremental Reoccurring Drift	Adaptive Naive Bayes (ANB)	Adaptive Random Forest (ARF)	Extremely Fast Decision Tree (EFDT)	Hoeffding Adaptive Tree (HAT)	Online Perception (OLP)	Very Fast Decision Rules (VFDR)	XGBoost + ADWIN	XGBoost + HDDM_W	XGBoost + KSWIN	Mean	Std Deviation	SABDM/RL
Weighted Accuracy (%)	47.93	**78.16**	68.95	66.34	72.64	59.80	53.27	56.38	53.77	61.92	9.54	62.47
Weighted F1-score (%)	44.39	**78.24**	68.74	65.56	72.64	58.45	53.66	56.73	53.94	61.37	10.09	59.79
Matthews MCC (%)	39.19	**73.39**	62.80	59.83	67.17	52.29	43.96	47.72	44.58	54.55	11.10	55.77
Minimum Accuracy (%)	7.50	**52.20**	36.10	30.70	47.50	27.40	4.60	3.70	2.90	23.62	18.42	20.70

TABLE 6. Weighted average accuracy results.

ML methods	Exp1 Acc	Exp1 #	Exp2 Acc	Exp2 #	Exp3 Acc	Exp3 #	Exp4 Acc	Exp4 #	Exp5 Acc	Exp5 #	Mean Acc	Mean #
ANB	47.5	10	50.3	10	53.0	10	58.6	7	47.9	10	51.5	10
ARF	**64.9**	**1**	**75.0**	**1**	**77.0**	**1**	**75.4**	**1**	**78.2**	**1**	**74.1**	**1**
EFDT	57.1	6	65.0	3	65.4	4	69.0	3	69.0	3	65.1	3
HAT	53.9	7	61.3	5	61.6	5	65.1	4	66.3	4	61.6	5
OLP	53.2	8	67.8	2	72.8	2	69.7	2	72.6	2	67.2	2
VFDR	52.1	9	54.0	9	60.3	6	57.4	8	59.8	6	56.7	9
XGB/ADWIN	60.9	4	58.9	6	58.6	7	55.3	10	53.3	9	57.4	7
XGB/HDDM_W	64.6	2	57.1	7	57.7	8	56.7	9	56.4	7	58.5	6
XGB/KSMIN	63.2	3	56.3	8	53.1	9	59.1	6	53.8	8	57.1	8
SABDM/RL	**60.8**	**5**	**63.4**	**4**	**69.6**	**3**	**63.6**	**5**	**62.5**	**5**	**64.0**	**4**
Mean	57.5		60.6		62.2		62.9		61.9		61.0	
Std. dev	5.9		7.2		7.8		6.7		9.5		6.4	

SABDM/RL Better Acc	SABDM/RL Worse Acc
[1,2,3,4,5]	
	[1,2,3,4,5]
[1,3]	[2,4,5]
[1,2,3]	[4,5]
[1]	[2,3,4,5]
[1,2,3,4,5]	
[2,3,4,5]	[1]
[2,3,4,5]	[1]
[2,3,4,5]	[1]
total=28	total=17

six online adaptive machine learning methods regarding the weighted average accuracy and MCC correlation coefficient. In Experiment 3, with incremental-gradual drift, the method was better in four out of six online adaptive machine learning methods regarding the weighted average accuracy and MCC correlation coefficient.

The XGBoost, with the support of concept drift detection methods, obtained better results in Experiment 1 with incremental drift, being among the best, but not in the other experiments. The main reason is that XGBoost requires larger training files to observe all the predicted classes during the training period. Consequently, the method adapts better to incremental drifts than abrupt drifts.

Compared to the other machine learning methods, the SABDM/RL method was among the best in terms of dynamic performance, as measured by the minimum accuracy (TABLE 9), being over the average in all experiments except in Experiment 5.

FIGURE 7 shows how the SABDM/RL method compares dynamically to the Adaptive Random Forest (ARF), which obtains the best results, and the Adaptive Naïve Bayes (ANB), which is the Bayesian reference.

The SABDM/RL method followed the performance of the ARF well, mainly under incremental drifts (Experiments 1 and 3), but it took longer to recover from abrupt drifts, as in Experiments 2, 4, and 5. Compared with ANB, the SABDM/RL method adjusted to both incremental and abrupt drifts better, showing that the Bayesian inference of the cognitive map belief parameters can provide good results in all drift conditions.

TABLE 7. Weighted average F1-score results.

ML methods	Exp1		Exp2		Exp3		Exp4		Exp5		Mean		SABDM/RL	
	F1 Score	#	F1 Score	#	F1 Score	#	F1 Score	#	F1 Score	#	F1 Score	#	Better F1-score	Worse F1-score
ANB	46.4	10	48.9	10	51.2	10	55.8	10	44.4	10	49.3	10	[1,2,3,4,5]	
ARF	64.3	2	**74.8**	**1**	**76.4**	**1**	**75.1**	**1**	**78.2**	**1**	**73.8**	**1**		[1,2,3,4,5]
EFDT	56.0	6	64.8	3	65.2	4	68.3	3	68.7	3	64.6	3	[1,3]	[2,4,5]
HAT	52.6	8	60.4	5	60.6	5	62.9	4	65.6	4	60.4	5	[1,2,3]	[4,5]
OLP	53.1	7	67.8	2	72.0	2	70.0	2	72.6	2	67.1	2	[1]	[2,3,4,5]
VFDR	50.8	9	52.8	9	60.0	6	55.9	9	58.5	6	55.6	9	[1,2,3,4,5]	
XGB/ADWIN	61.1	4	59.0	6	59.1	7	56.1	8	53.7	9	57.8	7	[2,3,4,5]	[1]
XGB/HDDM_W	64.7	1	57.0	7	57.4	8	57.6	7	56.7	7	58.7	6	[2,3,4,5]	[1]
XGB/KSMIN	**63.3**	**3**	55.9	8	52.9	9	59.9	6	53.9	8	57.2	8	[2,3,4,5]	[1]
SABDM/RL	**59.0**	**5**	**62.1**	**4**	**68.9**	**3**	**59.9**	**5**	**59.8**	**5**	**61.9**	**4**	total=28	total=17
Mean	56.9		60.2		61.6		62.4		61.4		60.5			
Std. dev	6.3		7.5		7.8		6.7		10.1		6.7			

TABLE 8. Matthews Correlation Coefficient (MCC) results.

ML methods	Exp1		Exp2		Exp3		Exp4		Exp5		Mean		SABDM/RL	
	MCC	#	MCC	#	MCC	#	MCC	#	MCC	#	MCC	#	Better MCC	Worse MCC
ANB	38.1	10	41.2	10	44.0	9	51.2	6	39.2	10	42.7	10	[1,2,3,4,5]	
ARF	**58,0**	**1**	**70.0**	**1**	**72.5**	**1**	**70.6**	**1**	**73.4**	**1**	**68.9**	**1**		[1,2,3,4,5]
EFDT	48.8	6	58.2	3	58.5	4	63.1	3	62.8	3	58.3	3	[1,3]	[2,4,5]
HAT	45.6	7	53.9	5	54.1	5	58.7	4	59.8	4	54.4	5	[1,2,3]	[4,5]
OLP	43.8	8	61.4	2	67.4	2	63.4	2	67.2	2	60.6	2	[1]	[2,3,4,5]
VFDR	43.5	9	45.3	9	52.5	6	49.4	7	52.3	6	48.6	8	[1,2,3,4,5]	
XGB/ADWIN	53.1	5	50.7	6	50.7	7	46.7	10	44.0	9	49.0	7	[1,2,3,4,5]	
XGB/HDDM_W	57.6	2	48.7	7	49.7	8	48.3	9	47.8	7	50.4	6	[2,3,4,5]	[1]
XGB/KSMIN	55.8	3	47.6	8	43.7	10	51.1	8	44.6	8	48.6	9	[2,3,4,5]	[1]
SABDM/RL	**53.9**	**4**	**56.6**	**4**	**63.6**	**3**	**57.4**	**5**	**55.8**	**5**	**57.5**	**4**	total=29	total=16
Mean	49.4		53.0		54.8		55.8		54.6		53.5			
Std. dev	6.7		8.4		9.3		7.9		11.1		7.5			

The results of the experiments with concept drift evolving conditions from incremental to abrupt to reoccurring show that the SABDM/RL method generalizes well under different evolving conditions. Additional experiments on other streaming datasets with concept drift conditions should be performed to verify its generalization further. In the experiments, the method was more effective under incremental than under abrupt conditions. It should be used with caution when abrupt concept drift conditions are expected in practical applications. Additional information may be necessary to advise on these situations and avoid unacceptable inaccuracies. Although using the SABDM/RL method in the initial configuration of regulated applications seems straightforward, caution should be taken regarding any automatism in their evolutionary maintenance because uncontrolled changes in the behavior of these systems are unacceptable. One suggestion is to use the SABDM/RL method results as a prediction to provide information to advise users of possible environmental changes requiring system maintenance.

TABLE 9. Minimum dynamic accuracy results.

ML methods	Exp1		Exp2		Exp3		Exp4		Exp5		Mean		SABDM/RL	
	Min Acc	#	Min Acc	#	Min Acc	#	Min Acc	#	Min Acc	#	Min Acc	#	Better Min Acc	Worse Min Acc
ANB	35.4	9	7.1	7	21.8	7	17.6	7	7.5	7	17.9	8	[1,2,3,4,5]	
ARF	36.1	8	**52.0**	**1**	**60.6**	**1**	**52.9**	**1**	**52.2**	**1**	**50.8**	**1**	[1]	[2,3,4,5]
EFDT	38.9	5	27.8	4	37.7	4	34.2	3	36.1	3	34.9	4	[1,2,3]	[4,5]
HAT	38.8	6	7.3	6	22.8	6	25.0	5	30.7	4	24.9	5	[1,2,3,4]	[5]
OLP	38.3	7	41.9	2	45.8	3	46.8	2	47.5	2	44.1	2	[1,3]	[2,4,5]
VFDR	33.7	10	7.0	8	21.7	8	22.7	6	27.4	5	22.5	7	[1,2,3,4]	[5]
XGB/ADWIN	45.8	3	14.1	5	35.2	5	13.6	8	4.6	8	22.7	6	[2,3,4,5]	[1]
XGB/HDDM_W	48.4	2	6.0	10	4.2	10	8.7	10	3.7	9	14.2	10	[2,3,4,5]	[1]
XGB/KSMIN	**50.1**	**1**	6.1	9	21.8	9	11.3	9	2.9	10	18.4	9	[2,3,4,5]	[1]
SABDM/RL	**44.8**	**4**	**41.5**	**3**	**53.9**	**2**	**25.6**	**4**	**17.4**	**6**	**36.6**	**3**	total=31	total=14
Mean	40.6		18.8		30.2		25.9		23.6		27.8			
Std. dev	5.6		16.6		15.6		14.8		18.4		12.0			

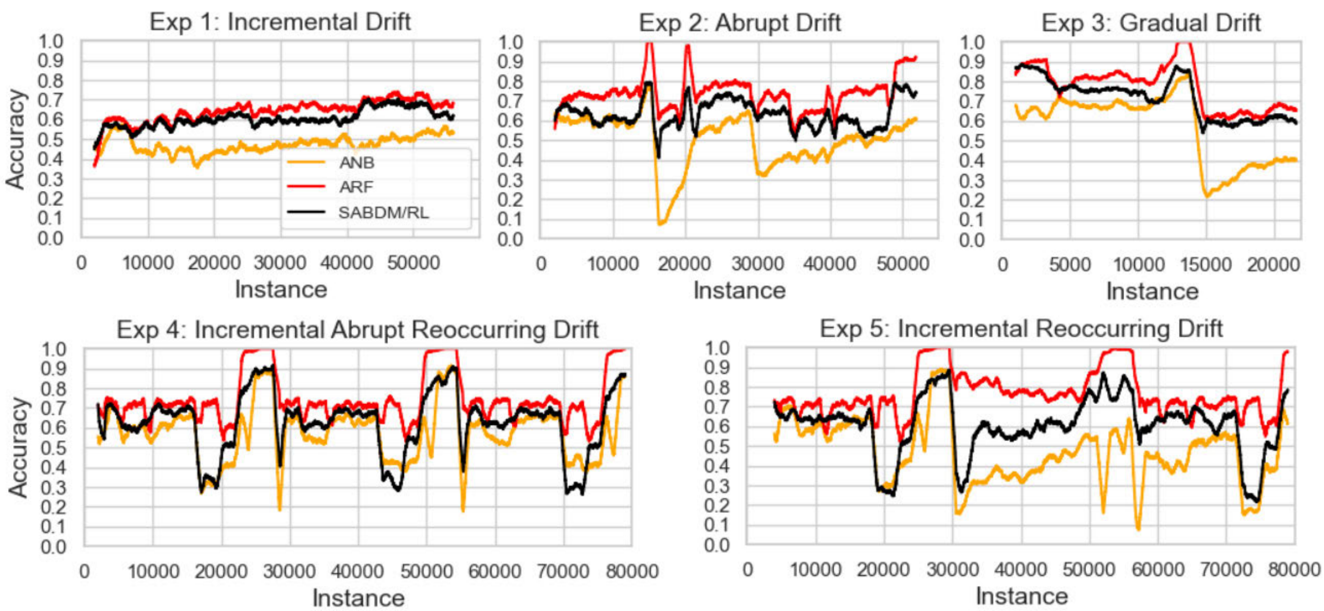


FIGURE 7. Dynamics comparative analysis.

VI. CONCLUSION

This study evaluated reinforcement learning as an online adaptive machine learning method for the automatic and continuous adjustment of situation-awareness-based decision-making applications.

The SABDM/RL method uses a Bayesian-based reinforcement learning technique (BA-C-MAB) to automatically adapt SA-GDTA goal-direct tree parameters. We verified that the method could recover accuracy conditions in distinct concept drift conditions by running experiments with incremental, abrupt, gradual, and reoccurring drift conditions.

We compared the results with those of six online adaptive machine learning algorithms and the XGBoost algorithm with the support of seven concept drift detection methods. The SABDM/RL method shows overall and dynamic results that are better than the average.

The experiments showed that, under diverse evolving conditions, the SABDM/RL method obtains better or similar results in direct comparison with the explainable Bayesian, tree-based, and rule-based methods. They also showed that the SABDM/RL method adjusts better under incremental drift types than under abrupt ones.

In future work, we will evaluate the integration of concept drift detectors with the SABDM/RL method to better understand abrupt drift conditions and adjust the algorithm accordingly. A second proposal is to evaluate the impact of SABDM/RL hyperparameters, such as impact factors and the BA-C-MAB learning rate, on the method performance using hyperparameter optimization techniques. We also intend to evaluate the method when teams of decision-makers are involved.

REFERENCES

[1] M. Hoogendoorn, R. M. van Lambalgen, and J. Treur, "Modeling situation awareness in human-like agents using mental models," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1697–1704, doi: 10.5591/978-1-57735-516-8/IJCAI11-285.

[2] M. R. Endsley and D. Jones, *Designing for Situation Awareness: An Approach to User-Centered Design*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2012, doi: 10.1201/b11371.

[3] M. R. Endsley, "Design and evaluation for situation awareness enhancement," *Proc. Hum. Factors Soc. Annu. Meeting*, vol. 32, no. 2, pp. 97–101, Oct. 1988, doi: 10.1177/154193128803200221.

[4] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Hum. Factors*, vol. 37, pp. 32–64, Sep. 1995, doi: 10.1518/001872095779049543.

[5] M. R. Endsley, "Situation awareness in future autonomous vehicles: Beware of the unexpected," in *Proc. 20th Congr. Int. Ergonom. Assoc.* (Advances in Intelligent Systems and Computing), vol. 824. Cham, Switzerland: Springer, Aug. 2018, pp. 303–309, doi: 10.1007/978-3-319-96071-5_32.

[6] N. Kim and N. Bodunkov, "Decision-making related to UAV control under uncertainty," in *Proc. 10th Int. Conf. Recent Adv. Civil Aviation*, 2023, pp. 395–403, doi: 10.1007/978-981-19-3788-0_35.

[7] B. Shen, L. D. Xu, H. Cai, H. Yu, P. Hu, L. Jiang, and J. Guo, "Enhancing context-aware reactive planning for unexpected situations of on-orbit spacecraft," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 6, pp. 4965–4983, Dec. 2022, doi: 10.1109/TAES.2022.3172022.

[8] R. Koopmanschap, M. Hoogendoorn, and J. J. Roessingh, "Tailoring a cognitive model for situation awareness using machine learning," *Int. J. Speech Technol.*, vol. 42, no. 1, pp. 36–48, Jan. 2015, doi: 10.1007/s10489-014-0584-3.

[9] G. D'Aniello, V. Loia, and F. Orciuoli, "Adaptive goal selection for improving situation awareness: The fleet management case study," *Proc. Comput. Sci.*, vol. 109, pp. 529–536, Jan. 2017, doi: 10.1016/j.procs.2017.05.332.

[10] M. R. Endsley, "Situation awareness global assessment technique (SAGAT)," in *Proc. IEEE Nat. Aerosp. Electron. Conf.*, Apr. 1988, pp. 789–795, doi: 10.1109/NAECON.1988.195097.

[11] R. M. Taylor, "Situational awareness rating technique (SART): The development of a tool for aircrew systems design," in *Situational Awareness*, London, U.K.: Routledge, 2018, pp. 111–128, doi: 10.4324/9781315087924-8.

[12] M. R. Endsley, "Direct measurement of situation awareness: Validity and use of SAGAT," in *Situational Awareness*. Boca Raton, FL, USA: CRC Press, 2017, pp. 129–155, doi: 10.4324/9781315087924-9.

[13] M. L. Gini, M. Hoogendoorn, and R. van Lambalgen, "Learning belief connections in a model for situation awareness," in *Agents in Principle, Agents in Practice* (Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7047. Berlin, Germany: Springer, Nov. 2011, pp. 373–384, doi: 10.1007/978-3-642-25044-6_30.

[14] X. Y. Zhou, Z. J. Liu, Z. L. Wu, and F. W. Wang, "Quantitative processing of situation awareness for autonomous ships navigation," *TransNav, Int. J. Mar. Navigat. Saf. Sea Transp.*, vol. 13, no. 1, pp. 25–31, 2019, doi: 10.12716/1001.13.01.01.

[15] Y. Wang and G. Ruhe, "The cognitive process of decision making," *Int. J. Cognit. Informat. Natural Intell.*, vol. 1, no. 2, pp. 73–85, Apr. 2007, doi: 10.4018/jcini.2007040105.

[16] P. N. Johnson-Laird, "Mental models in cognitive science," *Cognit. Sci.*, vol. 4, no. 1, pp. 71–115, Jan. 1980, doi: 10.1207/s15516709cog0401_4.

[17] P. N. Johnson-Laird, "Mental models and human reasoning," *Proc. Nat. Acad. Sci. USA*, vol. 107, no. 43, pp. 18243–18250, Oct. 2010, doi: 10.1073/pnas.1012933107.

[18] K. Sedki and L. B. de Beaufort, "Cognitive maps for knowledge represenation and reasoning," in *Proc. IEEE 24th Int. Conf. Tools Artif. Intell.*, Nov. 2012, pp. 1035–1040, doi: 10.1109/ICTAI.2012.175.

[19] T. Sammut-Bonnici and J. McGee, "Cognitive map," in *Wiley Encyclopedia of Management*. Hoboken, NJ, USA: Wiley, 2015, pp. 1–3, doi: 10.1002/9781118785317.weom120127.

[20] A. Sharma, S. Nazir, and J. Ernstsen, "Situation awareness information requirements for maritime navigation: A goal directed task analysis," *Saf. Sci.*, vol. 120, pp. 745–752, Dec. 2019, doi: 10.1016/j.ssci.2019.08.016.

[21] S. Rezaeifam, E. Ergen, and H. M. Günaydın, "Fire emergency response systems information requirements' data model for situational awareness of responders: A goal-directed task analysis," *J. Building Eng.*, vol. 63, Jan. 2023, Art. no. 105531, doi: 10.1016/J.JOBE.2022.105531.

[22] X. Geng and K. Smith-Miles, "Incremental learning," in *Encyclopedia of Biometrics*. Boston, MA, USA: Springer, 2009, pp. 731–735, doi: 10.1007/978-0-387-73003-5_304.

[23] Y. Luo, L. Yin, W. Bai, and K. Mao, "An appraisal of incremental learning methods," *Entropy*, vol. 22, no. 11, p. 1190, Oct. 2020, doi: 10.3390/E22111190.

[24] A. Shahraki, M. Abbasi, A. Taherkordi, and A. D. Jurcut, "A comparative study on online machine learning techniques for network traffic streams analysis," *Comput. Netw.*, vol. 207, Apr. 2022, Art. no. 108836, doi: 10.1016/J.COMNET.2022.108836.

[25] A. Blum, A. Kalai, and J. Langford, "Beating the hold-out: Bounds for K-fold and progressive cross-validation," in *Proc. 12th Annu. Conf. Comput. Learn. theory*, Jul. 1999, pp. 203–208, doi: 10.1145/307400.307439.

[26] J. Gama, R. Sebastião, and P. P. Rodrigues, "On evaluating stream learning algorithms," *Mach. Learn.*, vol. 90, no. 3, pp. 317–346, 2013, doi: 10.1007/s10994-012-5320-9.

[27] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–37, Mar. 2014, doi: 10.1145/2523813.

[28] S. C. H. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online learning: A comprehensive survey," *Neurocomputing*, vol. 459, pp. 249–289, Oct. 2021, doi: 10.1016/j.neucom.2021.04.112.

[29] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Mach. Learn.*, vol. 23, no. 1, pp. 69–101, Apr. 1996, doi: 10.1007/BF00116900.

[30] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2346–2363, Dec. 2019, doi: 10.1109/TKDE.2018.2876857.

[31] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, "Scikit-multiflow: A multi-output streaming framework," *J. Mach. Learn. Res.*, vol. 19, no. 72, pp. 1–5, 2018.

[32] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3171. Berlin, Germany: Springer, 2004, pp. 286–295, doi: 10.1007/978-3-540-28645-5_29.

[33] M. Baena-Garcìa, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno, "Early drift detection method," in *Proc. 4th Int. Workshop Knowl. Discovery Data Streams*, vol. 6, Sep. 2006, pp. 77–86.

[34] I. Frías-Blanco, J. del Campo-Ávila, G. Ramos-Jiménez, R. Morales-Bueno, A. Ortiz-Díaz, and Y. Caballero-Mota, "Online and non-parametric drift detection methods based on Hoeffding's bounds," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 3, pp. 810–823, Mar. 2015, doi: 10.1109/TKDE.2014.2345382.

[35] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2007, pp. 443–448, doi: 10.1137/1.9781611972771.42.

[36] C. Raab, M. Heusinger, and F.-M. Schleif, "Reactive soft prototype computing for concept drift streams," *Neurocomputing*, vol. 416, pp. 340–351, Nov. 2020.

[37] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, nos. 1–2, pp. 100–115, 1954, doi: 10.1093/biomet/41.1-2.100.

[38] R. S. Sutton and A. G. Barto, *Reinforcement Learning*, no. 6685, 2nd ed. Cambridge, MA, USA: MIT Press, 2018. [Online]. Available: https://mitpress.mit.edu/books/reinforcement-learning-second-edition

[39] R. A. Howard, *Dynamic Programming and Markov Processes*. Hoboken, NJ, USA: Wiley, 1960.

[40] M. N. Katehakis and A. F. Veinott, "The multi-armed bandit problem: Decomposition and computation," *Math. Oper. Res.*, vol. 12, no. 2, pp. 262–268, May 1987, doi: 10.1287/moor.12.2.262.

[41] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and non-stochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012, doi: 10.1561/2200000024.

[42] L. Zhou, "A survey on contextual multi-armed bandits," 2015, *arXiv:1508.03326*.

[43] M. Gan and O.-C. Kwon, "A knowledge-enhanced contextual bandit approach for personalized recommendation in dynamic domains," *Knowl.-Based Syst.*, vol. 251, Sep. 2022, Art. no. 109158, doi: 10.1016/j.knosys.2022.109158.

[44] C. Yan, H. Han, Y. Zhang, D. Zhu, and Y. Wan, "Dynamic clustering based contextual combinatorial multi-armed bandit for online recommendation," *Knowl.-Based Syst.*, vol. 257, Dec. 2022, Art. no. 109927, doi: 10.1016/J.KNOSYS.2022.109927.

[45] E. A. Silver, "Markovian decision processes with uncertain transition probabilities or rewards," M.S. thesis, MIT, Cambridge, MA, USA, 1963. Accessed: Nov. 2, 2021. [Online]. Available: https://www.researchgate.net/publication/34560132_Markovian_Decision_Processes_with_Uncertain_Transition_Probabilities_or_Rewards

[46] S. Ross, J. Pineau, B. Chaib-draa, and P. Kreitmann. (2011). *A Bayesian Approach for Learning and Planning in Partially Observable Markov Decision Processes*. Accessed: May 28, 2021. [Online]. Available: http://jmlr.org/papers/v12/ross11a.html

[47] V. Loia, G. D'Aniello, A. Gaeta, and F. Orciuoli, "Enforcing situation awareness with granular computing: A systematic overview and new perspectives," *Granular Comput.*, vol. 1, no. 2, pp. 127–143, Jun. 2016, doi: 10.1007/s41066-015-0005-y.

[48] A. Gaeta, V. Loia, and F. Orciuoli, "A comprehensive model and computational methods to improve situation awareness in intelligence scenarios," *Appl. Intell.*, vol. 51, no. 9, pp. 6585–6608, Sep. 2021.

[49] S. Agrahari and A. K. Singh, "Concept drift detection in data stream mining: A literature review," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9523–9540, Nov. 2022, doi: 10.1016/J.JKSUCI.2021.11.006.

[50] M. Vakili and M. Rezaei, "Incremental learning techniques for online human activity recognition," 2021, *arXiv:2109.09435*.

[51] D. Nallaperuma, R. Nawaratne, T. Bandaragoda, A. Adikari, S. Nguyen, T. Kempitiya, D. De Silva, D. Alahakoon, and D. Pothuher, "Online incremental machine learning platform for big data-driven smart traffic management," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4679–4690, Dec. 2019, doi: 10.1109/TITS.2019.2924883.

[52] J. Pearl, "Fusion, propagation, and structuring in belief networks," *Artif. Intell.*, vol. 29, no. 3, pp. 241–288, 1986, doi: 10.1016/0004-3702(86)90072-X.

[53] M. O. Duff, "Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes," Ph.D. thesis, Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, 2002. [Online]. Available: https://scholarworks.umass.edu/dissertations/AAI3039353

[54] R. Bellman, "A problem in the sequential design of experiments," *Sankhyā, Indian J. Statist.*, vol. 16, nos. 3–4, pp. 221–229, 1956, Accessed: May 28, 2021. [Online]. Available: http://www.jstor.org/stable/25048278

[55] V. M. A. Souza, D. M. dos Reis, A. G. Maletzke, and G. E. A. P. A. Batista, "Challenges in benchmarking stream learning algorithms with real-world data," *Data Mining Knowl. Discovery*, vol. 34, no. 6, pp. 1805–1858, Nov. 2020, doi: 10.1007/S10618-020-00698-5.

[56] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," *Mach. Learn.*, vol. 106, nos. 9–10, pp. 1469–1495, Oct. 2017, doi: 10.1007/S10994-017-5642-8.

[57] C. Manapragada, G. I. Webb, and M. Salehi, "Extremely fast decision tree," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1953–1962, doi: 10.1145/3219819.3220005.

[58] A. Bifet and R. Gavaldà, "Adaptive learning from evolving data streams," in *Proc. Int. Symp. Intell. Data Anal.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 5772, 2009, pp. 249–260, doi: 10.1007/978-3-642-03915-7_22.

[59] J. K. Anlauf and M. Biehl, "The AdaTron: An adaptive perceptron algorithm," *Europhys. Lett.*, vol. 11, no. 4, p. 387, Feb. 1990, doi: 10.1209/0295-5075/11/4/016.

[60] P. Kosina and J. Gama, "Very fast decision rules for classification in data streams," *Data Mining Knowl. Discovery*, vol. 29, no. 1, pp. 168–202, Jan. 2015, doi: 10.1007/S10618-013-0340-z.

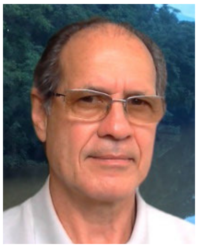
[61] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, vols. 13–17, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[62] V. U. Pugliese, R. D. Costa, and C. M. Hirata, "Comparative evaluation of the supervised machine learning classification methods and the concept drift detection methods in the financial business problems," in *Enterprise Information Systems*, J. Filipe, M. Śmiałek, A. Brodsky, and S. Hammoudi, Eds. Cham, Switzerland: Springer, 2021, pp. 268–292, doi: 10.1007/978-3-030-75418-1_13.

[63] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: An overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, May 2000, doi: 10.1093/BIOINFORMATICS/16.5.412.

[64] G. Jurman, S. Riccadonna, and C. Furlanello, "A comparison of MCC and CEN error measures in multi-class prediction," *PLoS ONE*, vol. 7, no. 8, Aug. 2012, Art. no. e41882, doi: 10.1371/JOURNAL.PONE.0041882.

[65] J. Gorodkin, "Comparing two K-category assignments by a K-category correlation coefficient," *Comput. Biol. Chem.*, vol. 28, nos. 5–6, pp. 367–374, Dec. 2004, doi: 10.1016/J.COMPBIOLCHEM.2004.09.006.

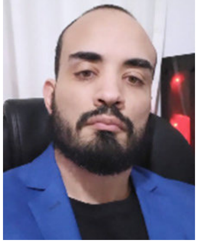


RENATO D. COSTA (Member, IEEE) received the B.Eng. degree in electronic engineering and the M.E.E. degree in electronic and computer engineering from the Instituto Tecnológico de Aeronáutica (ITA), Brazil, in 1977 and 1982, respectively, and the M.B.A. degree in enterprise management from the Fundação Getúlio Vargas (FGV), Brazil, in 1999. He is currently pursuing the Ph.D. degree in electronic and computer engineering with ITA.

From 1978 to 1985, he worked as an Assistant Researcher at the Spatial Activity Institute (IAE/CTA) and the National Spatial Research Institute (INPE), Brazil. From 1985 to 1995, he was an Assistant Professor with the Department of Electronic Engineering, ITA. He has extended professional experience in the computer industry, working mainly with critical and complex systems in aeronautics, industrial, and intelligent transportation applications. His research interests include situation awareness, explainable artificial intelligence, and reinforcement learning.

CELSO M. HIRATA received the B.S. and M.S. degrees in mechanical engineering and operations research from the Instituto Tecnológico de Aeronáutica (ITA), Brazil, and the Ph.D. degree in computer science from Imperial College London.

Since 2014, he has been a Full Professor with the Department of Computer Science, ITA. His research interests include distributed systems, cyber-physical systems, machine learning, and big data.



VICTOR U. PUGLIESE received the Technologist degree in analysis and systems development from the Instituto Federal de São Paulo (IFSP), Brazil, in 2015, the M.B.A. degree in software engineering from the Institute of Management and Information Technology (IGTI), Brazil, in 2017, and the M.E.E. degree in electronic and computer engineering from the Instituto Tecnológico de Aeronáutica (ITA), Brazil, in 2022. He is currently pursuing the Ph.D. degree in computer science with the Universidade Federal de São Paulo (UNIFESP), Brazil.

He has professional experience in the computer industry and works in artificial intelligence. His research interests include supervised learning, concept drift, and reinforcement learning.