An Explainable Evolutionary Grammar Approach for Interpretable Data Stream Classification with Concept Drift

Leandro Maciel Almeida and Leandro L. Minku, *Senior Member, IEEE*

***Abstract*—Data stream classification under concept drift presents fundamental challenges for machine learning systems, requiring models that adapt to evolving data distributions while maintaining interpretability. This paper introduces a novel rule-based classifier, termed EGIS, that leverages grammatical evolution to generate human-readable classification rules for non-stationary data streams. Unlike black-box ensemble methods that sacrifice transparency for accuracy, EGIS produces explicit IF-THEN rules that enable domain experts to understand, validate, and trust the model's predictions. The proposed method incorporates a multi-level self-adaptation framework that automatically adjusts evolutionary parameters in response to population diversity, performance history, and drift severity signals. A specialized mutation mechanism, termed Gene Therapy, injects discriminative patterns extracted from auxiliary decision trees directly into the evolutionary population, accelerating convergence during concept drift recovery. We introduce three novel transition metrics, namely Transition Change Score (TCS), Rule Instability Rate (RIR), and Average Modification Severity (AMS), that quantify rule evolution dynamics, revealing not only what the classifier predicts but how and why its decision logic changes over time. Comprehensive experiments across 48 datasets encompassing abrupt, gradual, recurring, and noisy drift scenarios demonstrate that EGIS achieves competitive predictive performance (average G-Mean of 0.78) while providing complete interpretability. Statistical analysis confirms that EGIS significantly outperforms ERulesD2S by 22-24 percentage points while maintaining a gap of only 2-4% relative to black-box ensemble methods. The proposed transition metrics provide unprecedented insight into classifier adaptation behavior, enabling practitioners to monitor how classification rules evolve in response to changing data distributions.**

***Index Terms*—Data stream classification, concept drift, explainable AI, grammatical evolution, rule-based learning, interpretable machine learning, self-adaptation**

I. Introduction

The proliferation of real-time data sources across diverse domains, ranging from financial transaction monitoring to industrial sensor networks, has created an urgent need for classification systems capable of processing continuous data streams while adapting to evolving patterns [1]. Unlike traditional batch learning paradigms where a model is trained once on a static dataset, data stream classification must address a constellation of interrelated challenges: concept drift manifests as changes in the underlying data distribution that render previously learned models obsolete; computational constraints demand single-pass processing with bounded memory; and the requirement for timely predictions precludes the luxury of storing data for retrospective analysis [2]. These challenges have motivated extensive research into adaptive learning algorithms that can detect distribution changes and update their models accordingly.

While the machine learning community has made substantial progress in developing adaptive classifiers for non-stationary environments, the dominant approaches rely on ensemble methods or complex tree structures that achieve high predictive accuracy at the cost of interpretability [3]. Methods such as Adaptive Random Forest (ARF) [3] and Streaming Random Patches (SRP) [15] combine hundreds of base learners whose individual predictions are aggregated through mechanisms that obscure the reasoning process from human understanding. This opacity is particularly problematic in high-stakes domains such as healthcare diagnostics, financial fraud detection, and critical infrastructure monitoring, where understanding *why* a model makes specific predictions is as important as the predictions themselves [4]. Domain experts in these fields cannot validate or trust a classifier whose decision logic remains hidden within an impenetrable ensemble, regardless of its accuracy on benchmark datasets.

The tension between predictive performance and interpretability represents a fundamental challenge in machine learning, yet the data streaming context introduces an additional dimension that has received insufficient attention: the need to understand not only *what* a model predicts at any given moment, but *how* and *why* its decision logic changes over time. When concept drift occurs, an adaptive classifier must modify its internal representation to track the evolving data distribution. For black-box methods, this adaptation process is entirely opaque; practitioners observe only that predictions have changed, with no insight into which aspects of the model were modified or why. In contrast, an interpretable classifier that maintains explicit decision rules offers the potential for transparency not only in individual predictions but also in the adaptation process itself. This capability would enable practitioners to verify that model changes align with their domain knowledge about expected distributional shifts, identify when the classifier has detected a genuine concept change versus responded to noise, and maintain an auditable history of how the decision logic has evolved throughout the data stream.

This paper introduces **EGIS** (Evolutionary Grammar for Interpretable Streams), a novel approach that addresses both interpretability requirements in data stream classification. EGIS

L. M. Almeida is with the Centro de Informática, Universidade Federal de Pernambuco, Recife, Brazil (e-mail: lma3@cin.ufpe.br).

L. L. Minku is with the School of Computer Science, University of Birmingham, Birmingham, B15 2TT, UK.

leverages grammatical evolution [5] to maintain a population of rule-based classifiers expressed as human-readable IF-THEN rules, enabling domain experts to understand model decisions without specialized machine learning knowledge. The evolutionary process adapts these rules as new data arrives, with a multi-level self-adaptation framework that automatically adjusts parameters based on population diversity, performance history, and drift severity. A specialized mutation mechanism called Gene Therapy accelerates drift recovery by injecting discriminative patterns extracted from auxiliary decision trees. Most importantly, EGIS introduces novel transition metrics that quantify how rules evolve between consecutive time periods, providing unprecedented insight into the adaptation process.

The research questions addressed in this paper are formulated as follows:

RQ1: Can a grammatical evolution approach achieve competitive predictive performance on data streams with concept drift while maintaining complete interpretability?

RQ2: How can the adaptation behavior of a rule-based classifier be quantified and analyzed to understand not only what changes occur but where, how, and when these changes manifest?

RQ3: Does a multi-level self-adaptation framework that responds to diversity, performance, and drift signals improve classifier robustness across different types of concept drift?

The main contributions of this paper are as follows:

1) **Interpretable Rule Evolution Framework**: We propose a grammatical evolution approach that generates explicit IF-THEN classification rules, enabling domain experts to understand model decisions and verify classifier behavior without specialized machine learning knowledge.
2) **Novel Transition Metrics**: We introduce three metrics, namely Transition Change Score (TCS), Rule Instability Rate (RIR), and Average Modification Severity (AMS), that quantify rule evolution dynamics, answering where, how, and when structural changes occur during concept drift adaptation.
3) **Multi-Level Self-Adaptation Framework**: We develop a comprehensive adaptation system with fifteen mechanisms that respond to population diversity, performance history, and drift severity signals, enabling automatic parameter adjustment without manual tuning.
4) **Gene Therapy Mutation**: We propose a knowledge-guided mutation operator that extracts discriminative patterns from specialized decision trees and injects them directly into the evolutionary population, accelerating convergence during drift recovery.
5) **Comprehensive Empirical Evaluation**: We conduct extensive experiments spanning 48 datasets across five drift categories (abrupt, gradual, noisy, stationary, and real-world streams), evaluating EGIS under six configurations that systematically vary chunk size (500, 1000, 2000 instances) and complexity penalty settings. The resulting experimental matrix comprises over 300 individual runs, compared against eight state-of-the-art methods with rigorous statistical validation including Friedman tests and pairwise Wilcoxon signed-rank tests with Bonferroni correction.

The remainder of this paper is organized as follows. Section II presents the formal problem formulation establishing the mathematical framework for data stream classification with interpretability requirements. Section III reviews related work on grammatical evolution, rule-based stream classifiers, and concept drift adaptation. Section IV describes the proposed EGIS method in detail, including the grammar-based representation, multi-objective fitness function, evolutionary operators, self-adaptation framework, and transition metrics. Section V presents the experimental setup, and Section VI discusses the results. Finally, Section VII concludes the paper and outlines directions for future work.

II. Problem Formulation

We formalize the problem of interpretable data stream classification with concept drift adaptation, establishing the mathematical framework that underlies the proposed approach. Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots\}$ denote an infinite data stream where each instance $(\mathbf{x}_t, y_t)$ arrives at discrete time step t, with $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$ representing a d-dimensional feature vector and $y_t \in \mathcal{Y} = \{c_1, c_2, \ldots, c_K\}$ denoting the class label from a finite set of K classes. The stream is generated according to a time-varying joint probability distribution $P_t(\mathbf{x}, y)$ that may change at unknown time points, a phenomenon known as concept drift.

Concept drift occurs when the joint distribution changes over time, formally expressed as $\exists t_1, t_2 : P_{t_1}(\mathbf{x}, y) \neq P_{t_2}(\mathbf{x}, y)$. This change may manifest through different mechanisms: real concept drift affects the posterior probability $P(y|\mathbf{x})$, altering the true decision boundary between classes, while virtual drift affects only the feature distribution $P(\mathbf{x})$ without changing the class-conditional relationships [1]. The temporal dynamics of drift further distinguish abrupt changes, where the distribution shifts instantaneously at a single time point, from gradual drift, where the transition occurs smoothly over an extended period. Recurring concepts represent a special case where previously observed distributions reappear after an intervening period.

For computational tractability, the data stream is typically processed in discrete chunks $\mathcal{D}_i = \{(\mathbf{x}_t, y_t) : t \in [(i-1) \cdot n + 1, i \cdot n]\}$, where n denotes the chunk size and i indexes consecutive chunks. The chunk-based processing paradigm enables bounded memory usage while providing sufficient data for meaningful model updates. The classifier $f : \mathcal{X} \to \mathcal{Y}$ is trained on each chunk $\mathcal{D}_i$ and evaluated on the subsequent chunk $\mathcal{D}_{i+1}$, following a **train-then-test** protocol. Unlike the prequential (test-then-train) approach commonly used in online learning where each instance is first used for testing before training, the train-then-test protocol evaluates the classifier on completely unseen data from the next temporal window. This methodology provides a more realistic assessment of generalization to future data, as the test set represents genuinely held-out instances rather than individual samples interleaved with training.

The interpretability requirement constrains the hypothesis space to rule-based classifiers of the form $\mathcal{R} =$

$\{r_1, r_2, \ldots, r_m\}$, where each rule r_k takes the form:

$$r_k : \text{IF } \phi_k(\mathbf{x}) \text{ THEN } c_k \tag{1}$$

Here $\phi_k : \mathcal{X} \to \{0, 1\}$ is a Boolean function composed of conjunctions and disjunctions of atomic conditions, and $c_k \in \mathcal{Y}$ is the consequent class. Each atomic condition compares an attribute value against a threshold using a relational operator:

$$\text{attr}_j \circ v, \quad \circ \in \{<, >, \leq, \geq, =, \neq\} \tag{2}$$

This representation ensures that any evolved classifier can be directly inspected and understood by domain experts, as each rule expresses its decision logic in explicit terms that require no machine learning expertise to interpret.

The transition monitoring problem extends the standard classification task to encompass analysis of classifier evolution. Given consecutive rule sets $\mathcal{R}_{t-1}$ and $\mathcal{R}_t$ evolved for chunks $\mathcal{D}_{t-1}$ and $\mathcal{D}_t$ respectively, we seek to quantify: (1) the overall magnitude of structural change, (2) the proportion of rules that were added, modified, or deleted, and (3) the severity of modifications to individual rules. This quantification enables practitioners to understand not only the current state of the classifier but also how it arrived at that state through incremental adaptation.

III. Related Work

The literature relevant to this work spans grammatical evolution for classification, rule-based methods for data streams, concept drift detection and adaptation, and explainability in machine learning. Grammatical evolution represents a form of genetic programming that uses a context-free grammar to constrain the search space to syntactically valid programs [5]. The grammar specifies production rules that map genotype representations to phenotype expressions, ensuring that all evolved solutions conform to the desired structural properties. This approach has found application in symbolic regression, automatic programming, and classification rule learning, where the grammar can encode domain-specific constraints on valid rule structures. The separation of genotype and phenotype enables standard genetic operators to work on integer chromosomes while the grammar mapping produces well-formed classification rules. Research has demonstrated that grammatical evolution can discover interpretable classifiers that rival neural networks on certain problems while maintaining the transparency that neural approaches lack [10].

Rule-based classifiers for data streams have received increasing attention as the demand for interpretable machine learning grows. The Very Fast Decision Rules (VFDR) algorithm [6] extends Hoeffding bounds to rule learning, enabling incremental rule induction from streaming data with provable guarantees. G-eRules [7] employs grammatical evolution to learn rules from streams, demonstrating that evolutionary approaches can achieve competitive performance while maintaining interpretability. ERulesD2S [8] extends this line of work by incorporating mechanisms for concept drift detection and rule adaptation, representing the closest prior work to our approach. However, ERulesD2S lacks the comprehensive self-adaptation framework and transition analysis capabilities that distinguish EGIS. The ACDWM algorithm [9] employs a weighted majority ensemble of chunk-based classifiers, though its ensemble nature limits interpretability despite using rule learners as base classifiers.

Concept drift detection and adaptation mechanisms have evolved substantially since early work on the problem. The ADWIN algorithm [11] maintains a sliding window that automatically grows or shrinks based on detecting distribution changes, providing a principled approach to forgetting obsolete data. The Drift Detection Method (DDM) [12] monitors the online error rate and signals drift when the error exceeds statistical thresholds. More recent approaches employ ensemble diversity [13] or explicit concept representations [14] to detect and respond to drift. Recent work has also explored the interplay between concept drift and class imbalance, with CDCMS.CIL [19] employing clustering in model space to maintain ensemble diversity under non-stationary class distributions. The severity-based approach in EGIS differs from binary drift detection by quantifying the magnitude of change and calibrating the adaptation response accordingly, enabling fine-grained control over the exploration-exploitation balance during recovery.

Ensemble methods dominate current benchmarks for data stream classification, achieving state-of-the-art accuracy through the aggregation of diverse base learners. Adaptive Random Forest (ARF) [3] combines online bagging with adaptive Hoeffding trees, using drift detectors to trigger tree replacement when performance degrades. Streaming Random Patches (SRP) [15] employs random subspaces and online bagging to create diverse ensembles that are robust to various drift types. ROSE [16] specifically addresses imbalanced data streams through adaptive oversampling techniques within an ensemble framework. While these methods achieve impressive predictive performance, their ensemble nature renders individual predictions unexplainable, a fundamental limitation that EGIS addresses through its rule-based representation.

The broader field of explainable artificial intelligence (XAI) has emphasized the importance of intrinsically interpretable models over post-hoc explanations [4]. Post-hoc methods such as LIME and SHAP provide local explanations for black-box predictions but do not guarantee that the explanation faithfully represents the model's actual reasoning process. In contrast, intrinsically interpretable models like rule sets and decision lists produce predictions through explicit logic that can be directly inspected. The data streaming context adds a temporal dimension to interpretability that has received limited attention: beyond understanding individual predictions, practitioners need to understand how the model's decision logic evolves over time. The transition metrics proposed in this paper address this gap by quantifying rule evolution dynamics throughout the stream.

IV. Proposed Method: EGIS

EGIS employs grammatical evolution to maintain a population of rule-based classifiers that adapt to concept drift through a sophisticated interplay of selection, crossover, mutation, and self-adaptation mechanisms. This section presents

the method in detail, beginning with the grammar-based rule representation and proceeding through the fitness function, evolutionary operators, self-adaptation framework, and transition metrics. The complete processing pipeline is summarized in Algorithm 1, followed by detailed textual descriptions of each phase.

A. Grammar-Based Rule Representation

The foundational representation scheme in EGIS employs a context-free grammar $G = (V_N, V_T, P, S)$ to generate classification rules that are inherently interpretable. The non-terminal symbols V_N define the structural elements of rules, including logical operators and comparison constructs. The terminal symbols V_T comprise the concrete elements that appear in final rules: attribute names drawn from the feature space, comparison operators appropriate to each attribute type, and threshold values within valid ranges. The production rules P specify how non-terminals expand into combinations of terminals and non-terminals, while the start symbol S initiates the derivation process.

The grammar enforces syntactic validity through carefully designed production rules that combine logical conditions with comparison operators. In Backus-Naur Form (BNF), the grammar is expressed as:

$$\begin{aligned}
\langle \text{ruleset} \rangle &::= \langle \text{rule} \rangle \mid \langle \text{rule} \rangle \langle \text{ruleset} \rangle \\
\langle \text{rule} \rangle &::= \text{IF } \langle \text{cond} \rangle \text{ THEN } \langle \text{class} \rangle \\
\langle \text{cond} \rangle &::= \langle \text{term} \rangle \mid \langle \text{term} \rangle \langle \text{logop} \rangle \langle \text{cond} \rangle \\
\langle \text{term} \rangle &::= \langle \text{attr} \rangle \langle \text{compop} \rangle \langle \text{value} \rangle \\
\langle \text{logop} \rangle &::= \text{AND} \mid \text{OR} \\
\langle \text{compop} \rangle &::= < \mid > \mid \leq \mid \geq \mid = \mid \neq
\end{aligned} \tag{3}$$

Each individual in the population represents a complete rule set $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$ capable of classifying instances across all classes. The representation follows a tree-based structure where internal nodes contain non-terminal symbols (logical operators, comparisons) while leaf nodes contain terminals (attributes, values, operators). This structure enables natural genetic operations while maintaining syntactic validity; any subtree exchange or node mutation produces a rule that conforms to the grammar specification. The complexity of each rule is measured through tree depth $d(r)$ and condition count $|\phi(r)|$, providing explicit control over the expressiveness-interpretability trade-off.

The internal representation employs Abstract Syntax Trees (ASTs) to capture the hierarchical structure of rule conditions. Each rule is parsed into an AST where internal nodes represent logical operators (AND, OR) and leaf nodes contain atomic conditions of the form $\text{attr}_j \circ v$. This tree-based representation enables efficient rule manipulation during evolutionary operations and provides a canonical form for rule comparison. For commutative operators, operands are lexicographically sorted to ensure that logically equivalent rules receive identical representations; for instance, $(\text{attr}_1 > 5) \wedge (\text{attr}_2 < 3)$ and $(\text{attr}_2 < 3) \wedge (\text{attr}_1 > 5)$ yield the same canonical AST. This normalization facilitates accurate similarity computation in the transition metrics, as semantically equivalent rules are recognized as identical rather than spuriously different.

The grammar adapts to the feature space by instantiating attribute-specific productions. Numeric attributes use comparison operators $\{<, >, \leq, \geq\}$ with threshold values sampled from the observed range, while categorical attributes use equality operators $\{=, \neq\}$ with values from the attribute domain. This type-aware generation ensures that all evolved conditions are semantically meaningful for the given dataset. A bias toward conjunctive rules (80% probability for AND versus 20% for OR at each expansion) produces more specific rules that are typically easier for humans to interpret.

B. Multi-Objective Fitness Function

The fitness function balances three fundamental objectives that govern rule quality, with dynamic weighting that adapts based on drift context. The primary objective captures predictive accuracy through the geometric mean of class-specific recalls, a metric that ensures balanced performance across classes:

$$\text{G-Mean} = \left(\prod_{k=1}^{K} \text{Recall}_k \right)^{1/K} \tag{4}$$

where $\text{Recall}_k = \text{TP}_k / (\text{TP}_k + \text{FN}_k)$ for class k. The geometric mean penalizes classifiers that achieve high accuracy on majority classes while neglecting minority classes, an important consideration for imbalanced data streams.

The coverage objective incentivizes rules that classify a larger proportion of instances, preventing overly specific rules that match few examples:

$$\text{Coverage}(\mathcal{R}) = \frac{|\{(\mathbf{x}, y) \in \mathcal{D} : \exists r \in \mathcal{R}, \phi_r(\mathbf{x}) = 1\}|}{|\mathcal{D}|} \tag{5}$$

A class-weighted coverage bonus further rewards rules that cover underrepresented classes, with weights inversely proportional to class frequencies.

The complexity penalization objective favors simpler, more interpretable structures:

$$\text{Complexity}(\mathcal{R}) = \lambda_r \cdot |\mathcal{R}| + \lambda_c \cdot \sum_{r \in \mathcal{R}} |\phi(r)| + \lambda_f \cdot |A(\mathcal{R})| \tag{6}$$

where $|\mathcal{R}|$ counts total rules, $|\phi(r)|$ counts conditions in rule r, and $|A(\mathcal{R})|$ counts distinct attributes used across all rules. The coefficients λ_r, λ_c, and λ_f control the relative importance of each complexity component.

The complete fitness function combines these objectives with stability penalties that discourage unnecessary structural changes:

$$F(\mathcal{R}) = \alpha \cdot \text{G-Mean} + \beta_c \cdot \text{Cov} - \gamma \cdot \text{Cplx} - \beta_s \cdot \text{Stab} \tag{7}$$

where Cov, Cplx, and Stab denote Coverage, Complexity, and Stability respectively. The stability penalty measures feature distance from a reference configuration using the Jaccard distance:

$$\text{Stability}(\mathcal{R}) = d_J(A(\mathcal{R}), A_{\text{ref}}) = \frac{|A(\mathcal{R}) \triangle A_{\text{ref}}|}{|A(\mathcal{R}) \cup A_{\text{ref}}|} \tag{8}$$

where A_{ref} denotes the features used by the best individual from the previous generation.

The coefficients governing these objectives are not static but adapt dynamically based on drift severity. When significant concept change is detected, stability constraints are relaxed to enable exploration of new rule structures. This adaptive weighting represents a key mechanism for balancing exploitation of learned knowledge with exploration of emerging patterns. Specifically, when drift severity exceeds a threshold, the stability penalty coefficient β_s is reduced to zero, and the complexity penalty is suspended, allowing the evolutionary process to discover radically different rule structures if needed.

An early termination mechanism conserves computational resources by interrupting evaluation of clearly inferior individuals. The first 20% of training instances (minimum 100) are evaluated, and if the partial G-Mean falls below 50% of the median fitness among elite individuals, evaluation terminates immediately. This heuristic reduces wasted computation on non-viable candidates, enabling more thorough evaluation of promising individuals.

C. Evolutionary Operators

The evolutionary process relies on carefully designed operators that balance exploration of the search space with exploitation of discovered solutions. Selection operates through a tournament mechanism with dynamically adjusted pressure. When population diversity is high, indicating risk of premature convergence, stronger selection pressure intensifies competition among individuals. When diversity diminishes, reduced pressure preserves alternative solutions that may prove valuable if conditions change. The tournament size varies according to:

$$k_t = k_{\min} + \delta \cdot (k_{\max} - k_{\min}) \tag{9}$$

where δ measures population diversity as the normalized fitness dispersion, and $k_{\min} = 2$, $k_{\max} = 5$ define the range.

The crossover operator implements a two-mode strategy that adapts throughout evolution. During early generations when class coverage is incomplete, *expansion mode* performs subtree exchanges that promote broad exploration of the search space. Two parent individuals are selected, random rules are chosen from each, and subtrees are swapped at compatible crossover points. This aggressive exploration helps discover rules covering classes that neither parent adequately addresses. As evolution progresses and populations achieve complete class coverage, the system transitions to *refinement mode*, which transfers high-quality rule components between individuals to intensify search in promising regions. For each class, the worst-performing rule in the child is identified along with the best-performing rule from the second parent; if the parent's rule exceeds the child's rule in quality, replacement occurs. This quality-directed transfer mechanism preserves good components while improving weak spots.

Mutation operates at three structural levels within rules. Operator mutation alters comparison operators with probability p_{op}, enabling fine-grained adjustments to decision boundaries, for example, changing "$\text{attr}_j > v$" to "$\text{attr}_j \geq v$". Value mutation perturbs thresholds within valid ranges, allowing smooth adaptation to shifting data distributions. A mutation of magnitude ϵ drawn from a Gaussian distribution centered at zero adjusts the threshold while respecting attribute bounds. Subtree mutation replaces entire rule components with newly generated structures, introducing substantial novelty when incremental adjustments prove insufficient. The overall mutation rate adapts inversely to population diversity:

$$p_m = p_m^{\text{base}} + (1 - \delta) \cdot (p_m^{\max} - p_m^{\text{base}}) \tag{10}$$

ensuring that mutation intensity increases as the population converges.

A specialized mutation mechanism, termed **Gene Therapy**, leverages knowledge extracted directly from training data to accelerate convergence. This approach trains a specialized decision tree for each class, creating a focused binary classifier that distinguishes the target class from all others. The tree is constrained to shallow depth (maximum 5 levels) to produce interpretable rules. Paths from root to leaf in this auxiliary tree are extracted as candidate rules, each representing a conjunction of conditions that discriminate the target class. These candidate rules are scored by a function combining classification confidence with instance coverage:

$$\text{Score}(r) = \text{Purity}(r) \cdot \log(1 + \text{Support}(r)) \tag{11}$$

where $\text{Purity}(r)$ measures the proportion of correctly classified instances among those matching the rule, and $\text{Support}(r)$ counts the matching instances. The highest-scoring rule replaces the worst-performing rule for the corresponding class in the target individual. This mechanism injects discriminative knowledge directly into the evolutionary population, providing a shortcut to good solutions that pure random mutation might take many generations to discover. Gene Therapy is particularly valuable during drift recovery, when new patterns must be rapidly incorporated into the classifier.

An additional intensification mechanism, **Hill Climbing**, applies local search to elite individuals. Three strategies operate based on performance classification: aggressive intensification for poor performers (G-Mean below 85%) generates many variants through error-focused decision tree extraction; moderate intensification (85-96%) combines boosted ensemble predictions; fine-tuning (above 96%) applies guided mutations weighted by feature importance. This hierarchical approach allocates computational effort according to the improvement potential of each individual.

D. Self-Adaptation Framework

EGIS implements a multi-level self-adaptation system that automatically adjusts parameters and strategies in response to three complementary signals: population diversity, performance history, and drift severity. This framework eliminates the need for manual parameter tuning across different datasets and drift scenarios, enabling robust performance without domain-specific configuration.

Diversity-based adaptation monitors population convergence through fitness value dispersion. The diversity metric δ is computed as the normalized range of fitness values among population members. When diversity diminishes below a threshold, indicating premature convergence risk,

mutation rate increases automatically to reintroduce variability. Simultaneously, selection pressure decreases to preserve alternative solutions that might otherwise be eliminated by strong competition. The crossover mode transition from expansion to refinement also responds to diversity signals: high diversity suggests the population has not yet converged on a promising region, warranting continued exploration.

Performance-based adaptation classifies predictive performance into three levels based on recent history and absolute thresholds. Good performance (G-Mean exceeds the historical mean plus one standard deviation, or exceeds an absolute threshold of 0.85) indicates the current rule set effectively captures the underlying concept, warranting conservative exploration with substantial inheritance from previous populations. Medium performance (within one standard deviation of the mean) suggests the classifier is adequate but could benefit from moderate exploration. Poor performance (below the mean minus one standard deviation, or below an absolute threshold of 0.30) signals potential concept obsolescence, triggering increased injection of new individuals and reduced reliance on inherited knowledge. The stability penalty coefficient β_s adapts according to performance classification: high values (0.05) for good performance encourage stability, while low values (0.01) for poor performance permit structural change.

Drift-based adaptation employs a severity analysis framework that classifies detected changes into four levels. The drift severity metric combines three complementary signals:

$$S_{\text{drift}} = 1 - (0.5 \cdot \text{sim}_{\mu} + 0.3 \cdot \text{sim}_{\pi} + 0.2 \cdot \text{sim}_{\sigma}) \qquad (12)$$

where sim_{μ} measures cosine similarity between feature means across chunks, sim_{π} measures L_1 similarity between class distributions, and sim_{σ} measures cosine similarity between feature standard deviations. This multi-faceted metric captures different aspects of distribution change: feature location shift, class proportion change, and feature dispersion change.

Based on the computed severity, the system classifies drift into four levels and triggers corresponding responses:

- **Stable** ($S_{\text{drift}} < 0.05$): Normal operation with full penalty enforcement and standard parameters.
- **Mild** ($0.05 \leq S_{\text{drift}} < 0.10$): Subtle parameter adjustments without disrupting the learned model.
- **Moderate** ($0.10 \leq S_{\text{drift}} < 0.25$): Stability penalties removed, evolutionary budget potentially increased to 15 additional generations.
- **Severe** ($S_{\text{drift}} \geq 0.25$): Partial knowledge reset with mutation rate elevated to 0.5, 60% of population reinitialized randomly, and up to 25 additional recovery generations.

The adaptive memory management system maintains two complementary structures. A best-solutions memory stores high-fitness individuals encountered throughout the stream, pruned periodically to retain the most recent and highest-quality solutions. When critical performance degradation indicates stored knowledge has become obsolete (performance drop exceeding 55% over consecutive chunks), this memory is automatically cleared through an abandonment mechanism. A concept fingerprint memory stores statistical summaries of processed concepts, enabling detection of recurring patterns. Each fingerprint captures the mean vector, standard deviation vector, and class distribution of a chunk. When a new chunk arrives, its fingerprint is compared against stored fingerprints; if similarity exceeds 0.85, a recurring concept is identified, and previously successful solutions are restored from concept memory. This mechanism dramatically accelerates re-adaptation when concepts recur, a common phenomenon in many real-world streams.

Population initialization follows an adaptive recipe determined by problem complexity and drift context. A rapid probe using a shallow decision tree (maximum depth 3) estimates current problem complexity by measuring the probe's accuracy on the current chunk. High probe accuracy (above 0.90) indicates a simple problem where decision tree seeding should dominate (80% of population). Medium probe accuracy (0.75-0.90) suggests moderate complexity with 60% seeding. Low probe accuracy (below 0.75) indicates a complex problem where random initialization should predominate (60%) to ensure adequate exploration. The final recipe balances inherited knowledge (from memory and previous population elite) with exploration (seeded and random individuals) according to both complexity assessment and current drift context.

A distinguishing characteristic of EGIS is its reliance on **implicit drift detection** rather than explicit statistical tests. Unlike methods that employ dedicated drift detectors such as ADWIN [11] or DDM [12] to trigger model updates, EGIS continuously adapts through the interplay of fingerprint comparison and performance monitoring. The severity framework (Equation 7) quantifies distributional change without requiring binary drift/no-drift decisions, enabling graduated responses proportional to the magnitude of change. This design philosophy avoids the sensitivity-specificity trade-off inherent in threshold-based detection: excessively sensitive detectors trigger frequent false alarms and unnecessary adaptation, while conservative thresholds may delay response to genuine drift. By treating drift detection and adaptation as a unified continuous process, EGIS achieves robust performance across drift scenarios without detector tuning.

E. Transition Metrics

The transition metrics constitute a central contribution of this work, quantifying rule evolution dynamics between consecutive chunks and answering three fundamental questions about classifier adaptation. Unlike standard accuracy metrics that reveal only predictive performance at a single time point, these metrics characterize the adaptation process itself, enabling practitioners to understand how and why the classifier changes over time.

The **Rule Instability Rate (RIR)** measures the proportion of rules that were structurally modified between consecutive time points. To compute RIR, rules from consecutive chunks $\mathcal{R}_{t-1}$ and $\mathcal{R}_t$ are matched using a similarity function based on normalized Levenshtein distance between rule string representations. Rules exceeding a similarity threshold τ (typically 0.35) are considered modifications of the same

underlying rule; below this threshold, they are treated as complete replacements. RIR then counts the proportion of rules that were either added (present in $\mathcal{R}_t$ but with no similar match in $\mathcal{R}_{t-1}$) or deleted (present in $\mathcal{R}_{t-1}$ but with no similar match in $\mathcal{R}_t$):

$$\text{RIR} = \frac{|\text{Added}| + |\text{Deleted}|}{|\mathcal{R}_{t-1}| + |\mathcal{R}_t|} \tag{13}$$

This metric indicates *where* structural changes occurred in the rule set, identifying which classifier components were replaced during adaptation. High RIR values indicate substantial structural turnover, while low values indicate stability in the rule set composition.

The **Average Modification Severity (AMS)** quantifies the degree of alteration for rules that were modified rather than completely replaced. For each matched rule pair (r_{t-1}, r_t) where similarity exceeds τ, the modification severity is computed as one minus the similarity:

$$\text{AMS} = \frac{1}{|\text{Modified}|} \sum_{(r_{t-1}, r_t) \in \text{Modified}} (1 - \text{sim}(r_{t-1}, r_t)) \tag{14}$$

The similarity function employs normalized Levenshtein edit distance:

$$\text{sim}(r_1, r_2) = 1 - \frac{\text{edit\_distance}(\text{str}(r_1), \text{str}(r_2))}{\max(|\text{str}(r_1)|, |\text{str}(r_2)|)} \tag{15}$$

AMS indicates *how* individual rules changed, capturing the magnitude of modifications realized during adaptation. High AMS with low RIR suggests gradual refinement of existing rules; low AMS with high RIR suggests complete rule replacement.

The **Transition Change Score (TCS)** combines instability and severity into a unified metric that characterizes overall adaptation intensity:

$$\text{TCS} = w_1 \cdot \text{RIR} + w_2 \cdot (1 - \text{RIR}) \cdot \text{AMS} \tag{16}$$

where default weights $w_1 = 0.6$ and $w_2 = 0.4$ balance the contributions of complete replacements (captured by RIR) and partial modifications (captured by the product of retention proportion and AMS). TCS indicates *when* significant transitions occurred, enabling identification of inflection points in classifier adaptation. A TCS near 1.0 indicates radical transformation of the rule set; TCS near 0.0 indicates minimal change.

Joint interpretation of these metrics enables rich analysis of adaptation behavior. Peaks in TCS signal moments of significant change that may correspond to concept drift events. High RIR with low AMS indicates a rule replacement strategy where entire rules are swapped rather than modified. Low RIR with high AMS indicates a gradual refinement strategy where existing rules are incrementally adjusted. Sequences of low TCS indicate stable concepts where the classifier requires minimal adaptation. When TCS spikes following a period of stability, practitioners can examine which rules changed and how, correlating these changes with domain knowledge about expected distributional shifts.

F. Algorithm Description

Algorithm 1 presents the complete EGIS processing pipeline. The algorithm processes the data stream in discrete chunks, updating the classifier as each chunk arrives.

Algorithm 1 EGIS: Evolutionary Grammar for Interpretable Streams

Require: Data stream $\mathcal{S}$, chunk size n, population size N
Ensure: Sequence of classifiers $\{f_1, f_2, \ldots\}$ and transition metrics
Initialize grammar G from feature space
Initialize empty memories: $\mathcal{M}_{\text{best}} \leftarrow \emptyset$, $\mathcal{M}_{\text{concept}} \leftarrow \emptyset$
Initialize previous rules $\mathcal{R}_{\text{prev}} \leftarrow \emptyset$
for each chunk $\mathcal{D}_i$ from stream $\mathcal{S}$ **do**
  // Phase 1: Fingerprint and Recurrence Detection
  $\text{fp}_i \leftarrow \text{ComputeFingerprint}(\mathcal{D}_i)$
  $\text{recurring} \leftarrow \text{DetectRecurrence}(\text{fp}_i, \mathcal{M}_{\text{concept}})$
  // Phase 2: Drift Severity Classification
  $S_{\text{drift}} \leftarrow \text{ComputeDriftSeverity}(\text{fp}_i, \text{fp}_{i-1})$
  $\text{level} \leftarrow \text{ClassifyDrift}(S_{\text{drift}})$
  Adjust penalties and parameters based on level
  // Phase 3: Population Initialization
  $\text{complexity} \leftarrow \text{ProbeComplexity}(\mathcal{D}_i)$
  $\mathcal{P}_0 \leftarrow \text{InitializePopulation}(N, \text{complexity}, \text{recurring}, \mathcal{M}_{\text{best}})$
  // Phase 4: Evolutionary Optimization
  for generation $g = 1$ to $g_{\max}$ **do**
    Evaluate fitness: $F(p) \leftarrow$ Equation 7 for all $p \in \mathcal{P}_{g-1}$
    $\delta \leftarrow \text{ComputeDiversity}(\mathcal{P}_{g-1})$
    Adjust tournament size and mutation rate based on δ
    $\mathcal{P}_{\text{elite}} \leftarrow \text{SelectElite}(\mathcal{P}_{g-1}, 0.1 \cdot N)$
    $\mathcal{P}_{\text{parents}} \leftarrow \text{TournamentSelection}(\mathcal{P}_{g-1})$
    $\mathcal{P}_{\text{offspring}} \leftarrow \text{AdaptiveCrossover}(\mathcal{P}_{\text{parents}}, \delta)$
    $\mathcal{P}_{\text{mutated}} \leftarrow \text{Mutation}(\mathcal{P}_{\text{offspring}}, p_m)$
    $\mathcal{P}_{\text{mutated}} \leftarrow \text{GeneTherapy}(\mathcal{P}_{\text{mutated}}, \mathcal{D}_i)$
    $\mathcal{P}_g \leftarrow \mathcal{P}_{\text{elite}} \cup \mathcal{P}_{\text{mutated}}$
    if stagnation detected **then**
      Apply Hill Climbing to elite individuals
    end if
  end for
  // Phase 5: Recovery (if severe drift)
  if level = SEVERE **then**
    Execute additional recovery generations with elevated mutation
  end if
  // Phase 6: Output and Memory Update
  $f_i \leftarrow \text{BestIndividual}(\mathcal{P}_{g_{\max}})$
  $\mathcal{R}_i \leftarrow \text{ExtractRules}(f_i)$
  // Phase 7: Transition Metrics
  Compute RIR, AMS, TCS using Equations 13-16
  $\mathcal{R}_{\text{prev}} \leftarrow \mathcal{R}_i$
  // Phase 8: Memory Management
  Update $\mathcal{M}_{\text{best}}$ with elite individuals
  Store $(\text{fp}_i, \text{elite})$ in $\mathcal{M}_{\text{concept}}$
  if abandonment criteria met **then**
    $\mathcal{M}_{\text{best}} \leftarrow \emptyset$
  end if
  Output: Classifier f_i, Rules $\mathcal{R}_i$, Metrics (RIR, AMS, TCS)
end for

The algorithm begins by initializing the grammar from the feature space specification, creating production rules appropriate for each attribute type. Empty memory structures are created for storing high-fitness individuals and concept fingerprints. Processing then proceeds through eight phases for each chunk.

Phase 1 computes a statistical fingerprint of the current chunk and checks for recurring concepts by comparing against

stored fingerprints. If a recurring concept is detected (similarity exceeding 0.85), previously successful solutions are retrieved from concept memory to accelerate re-adaptation.

Phase 2 computes drift severity by comparing the current fingerprint against the previous chunk's fingerprint. The severity score determines the adaptation level (stable, mild, moderate, or severe), which in turn controls penalty coefficients and evolutionary parameters. Severe drift triggers penalty suspension and parameter adjustments that favor exploration over stability.

Phase 3 initializes the population according to an adaptive recipe. A shallow decision tree probe estimates problem complexity, informing the proportion of seeded versus random individuals. Recurring concept detection influences whether to restore previously successful solutions. The final population combines inherited knowledge (from memory and previous elite) with exploration (seeded rules and random individuals) in proportions determined by the current context.

Phase 4 executes the main evolutionary loop. Each generation evaluates fitness using Equation 7, computes population diversity, and adjusts tournament size and mutation rate accordingly. Elite individuals (top 10%) are preserved unchanged. Parents are selected through adaptive tournament selection, recombined through adaptive crossover (expansion or refinement mode based on class coverage), and mutated at the adjusted rate. Gene Therapy injects knowledge from specialized decision trees. If stagnation is detected (no fitness improvement for multiple generations), Hill Climbing applies intensive local search to elite individuals.

Phase 5 handles severe drift recovery. When drift classification indicates severe change, additional generations with elevated mutation rate (0.5) and increased random individual proportion (60%) enable rapid exploration of the search space. This recovery mode continues until performance stabilizes or a maximum generation budget is exhausted.

Phases 6-8 finalize chunk processing. The best individual becomes the current classifier, and its rules are extracted for output and metric computation. Transition metrics (RIR, AMS, TCS) are computed by comparing current rules against the previous chunk's rules using Equations 13-16. Memory structures are updated with elite individuals and the current fingerprint. If abandonment criteria are met (severe performance degradation over consecutive chunks), the best-solutions memory is cleared to prevent obsolete knowledge from contaminating future evolution.

G. Explainability Analysis Tools

EGIS provides a suite of analysis tools for examining adaptation dynamics beyond the transition metrics. The rule difference analyzer compares consecutive rule sets, identifying unchanged, modified, added, and deleted rules through Levenshtein similarity with configurable thresholds. It generates detailed textual reports listing each rule change with its similarity score, enabling practitioners to trace exactly how the classifier evolved between chunks. Evolution matrices visualize the modification patterns as heatmaps, with rows representing previous rules, columns representing current rules, and cell values indicating pairwise similarities.

The performance-based drift detector monitors accuracy throughout the stream, identifying significant drops that correlate with concept changes. When accuracy falls below the historical mean by more than one standard deviation, a drift warning is triggered. This detector operates independently of the statistical fingerprinting mechanism, providing complementary evidence for concept change.

The concept difference analyzer quantifies differences between concept definitions through label disagreement rates. Given two classifiers trained on different chunks, the analyzer measures how often they disagree on predictions for a held-out sample. This disagreement rate indicates the magnitude of concept change and informs the severity analysis framework. Disagreement matrices produced by this analyzer reveal which class pairs experience the most decision boundary shift, helping practitioners understand not just that drift occurred but which concepts were most affected.

V. Experimental Setup

We evaluate EGIS through comprehensive experiments designed to assess both predictive performance and adaptation dynamics across diverse concept drift scenarios. The experimental corpus encompasses 48 datasets spanning abrupt, gradual, recurring, and noisy drift patterns, enabling systematic analysis of how the self-adaptation framework responds to different drift characteristics. This section describes the datasets employed, the comparative methods against which EGIS is benchmarked, the experimental configurations that vary chunk size and complexity penalty, and the evaluation metrics used to quantify performance and classifier stability.

A. Datasets

The experimental corpus comprises 48 data streams spanning five distinct categories designed to stress-test different aspects of adaptive classification. Synthetic datasets generated through the MOA framework enable precise evaluation under controlled drift conditions, while real-world datasets validate performance on authentic streaming scenarios with unknown distributional dynamics. Table I organizes these datasets by drift category, distinguishing abrupt transitions (instantaneous distribution changes), gradual drift (extended transition periods), noisy drift (label corruption during transitions), stationary streams (baseline without drift), and real-world scenarios with undocumented drift characteristics.

All synthetic datasets were generated once using the MOA framework [2] with fixed random seeds to ensure reproducibility. Each dataset comprises 12,000 instances, subsequently partitioned into non-overlapping temporal chunks of 500, 1,000, or 2,000 instances depending on the experimental configuration. This pre-generation strategy guarantees that all comparative methods process identical data sequences, eliminating any confounding effects from stochastic data generation during evaluation. Concept drifts in synthetic streams occur at predetermined chunk boundaries, enabling precise analysis of adaptation behavior at known transition points.

TABLE I
DATASET CATEGORIES AND CHARACTERISTICS

Category	Datasets
Abrupt Drift	SEA_Abrupt_Simple, SEA_Abrupt_Chain, SEA_Abrupt_Recurring, AGRAWAL_Abrupt_Simple_Mild, AGRAWAL_Abrupt_Simple_Severe, AGRAWAL_Abrupt_Chain_Long, RBF_Abrupt_Blip, RBF_Abrupt_Severe, HYPERPLANE_Abrupt_Simple, STAGGER_Abrupt_Chain, STAGGER_Abrupt_Recurring, RANDOMTREE_Abrupt_Simple, RANDOMTREE_Abrupt_Recurring, SINE_Abrupt_Simple, LED_Abrupt_Simple, WAVEFORM_Abrupt_Simple
Gradual Drift	SEA_Gradual_Simple_Fast, SEA_Gradual_Simple_Slow, SEA_Gradual_Recurring, RBF_Gradual_Moderate, RBF_Gradual_Severe, HYPERPLANE_Gradual_Simple, STAGGER_Gradual_Chain, RANDOMTREE_Gradual_Simple, SINE_Gradual_Recurring, LED_Gradual_Simple, WAVEFORM_Gradual_Simple
Noisy Drift	AGRAWAL_Abrupt_Simple_Severe_Noise, RBF_Abrupt_Blip_Noise, RBF_Gradual_Severe_Noise, HYPERPLANE_Gradual_Noise, RANDOMTREE_Gradual_Noise, SEA_Abrupt_Chain_Noise, STAGGER_Abrupt_Chain_Noise, SINE_Abrupt_Recurring_Noise
Stationary	AGRAWAL_Stationary, SEA_Stationary, RBF_Stationary, HYPERPLANE_Stationary, STAGGER_Stationary, RANDOMTREE_Stationary, SINE_Stationary, LED_Stationary, WAVEFORM_Stationary
Real-World	Electricity, AssetNegotiation_F2, AssetNegotiation_F3, AssetNegotiation_F4

TABLE II
DATASET DIMENSIONS AND CHUNK STRUCTURE

Chunk Size	Instances	Chunks	Evals	Runs
500	12,000	24	23	$48 \times 2 = 96$
1,000	12,000	12	11	$48 \times 2 = 96$
2,000	12,000	6	5	$48 \times 2 = 96$
Total experimental runs:				**288**

Table II summarizes the chunk structure and resulting evaluation counts for each configuration.

Abrupt drift streams feature instantaneous distribution changes at predetermined chunk boundaries, simulating scenarios such as sudden policy changes or system failures. The severity varies from mild (function parameter changes in AGRAWAL) to severe (complete concept replacement in RBF). **Gradual drift** streams transition between concepts over extended periods spanning multiple chunks, with transition speeds ranging from fast (100 instances) to slow (500 instances). **Recurring drift** streams cycle through previously observed concepts, testing the classifier's ability to recognize and rapidly re-adapt to familiar distributions. **Noisy drift** streams add 10-20% label noise during transitions, stress-testing robustness against imperfect drift signals. **Stationary** streams provide baseline performance without distribution change, verifying that adaptive mechanisms do not degrade performance when adaptation is unnecessary. Real-world datasets represent genuine streaming scenarios from domains including electricity demand and asset negotiation.

B. Comparative Methods

We compare EGIS against eight state-of-the-art methods spanning ensemble approaches, tree-based methods, and interpretable classifiers:

- **ROSE** [16]: Robust Online Self-adjusting Ensemble for imbalanced streams, evaluated in both original and chunk-based evaluation configurations.
- **ARF** [3]: Adaptive Random Forest combining online bagging with adaptive Hoeffding trees.
- **SRP** [15]: Streaming Random Patches employing random subspaces.
- **CDCMS.CIL** [19]: Concept Drift handling based on Clustering in Model Space for Class-Imbalanced Learning, a heterogeneous ensemble that maintains diversity through model-space clustering with G-Mean weighted voting.
- **HAT** [18]: Hoeffding Adaptive Tree with ADWIN change detection.
- **ACDWM** [9]: Adaptive Chunk-based Dynamic Weighted Majority.
- **ERulesD2S** [8]: Evolutionary Rules for Data Streams, the only other interpretable baseline.

Model Limitations and Scope. Several comparative methods exhibit inherent limitations that affect their applicability across the full dataset corpus. ACDWM is designed exclusively for binary classification problems; when applied to multi-class datasets (LED and WAVEFORM generators), the algorithm encounters numerical instability during training in its under-bagging component, which assumes strictly binary class labels. Following established practice in comparative studies [20], we assign G-Mean=0.0 to these failed cases, reflecting the method's inability to handle multi-class problems while maintaining transparency in ranking comparisons.

HAT exhibits systematic underfitting on smaller chunk sizes, with training G-Mean approximately 8% lower than EGIS. This outcome is consistent with documented limitations of Hoeffding tree estimation, which requires substantial instance counts to reliably compute split statistics. ERulesD2S, while designed for interpretability, frequently produced collapsed classifiers assigning uniform predictions to all instances, generating no valid rules in our experimental configuration.

Among ensemble methods, ARF, SRP, ROSE, and CDCMS.CIL demonstrated robust performance across most dataset types but provide no interpretability. Their predictions emerge from aggregating multiple base learners through mechanisms that preclude explanation of individual decisions, representing the fundamental accuracy-interpretability trade-off that motivates EGIS development. CDCMS.CIL stands out among ensembles for its native G-Mean optimization and full

multiclass support, making it a particularly strong baseline for class-imbalanced scenarios.

Table III summarizes the capabilities and limitations of each method.

TABLE III
COMPARATIVE METHODS: CAPABILITIES AND LIMITATIONS

Method	Multi.	Imbal.	Interp.	Type
EGIS	✓	✓	Full	Rules
ARF	✓	Partial	None	Ensemble
SRP	✓	Partial	None	Ensemble
CDCMS.CIL	✓	Focus	None	Ensemble
HAT	✓	Limited	Low	Tree
ROSE	✓	✓	None	Ensemble
ACDWM	–	✓	None	Ensemble
ERulesD2S	✓	Limited	Medium	Rules

C. Experimental Configurations

The experimental design systematically varies chunk size and complexity penalty to evaluate adaptation behavior across different temporal granularities and interpretability constraints. Table IV presents the six experimental configurations.

TABLE IV
EXPERIMENTAL CONFIGURATIONS

Config	Chunk	Pen.	Sets	Focus
EXP-500-NP	500	0.0	48	Fine-grained
EXP-500-P	500	0.1	48	Interpretability
EXP-1000-NP	1000	0.0	48	Balanced
EXP-1000-P	1000	0.1	48	Complexity ctrl
EXP-2000-NP	2000	0.0	48	Larger windows
EXP-2000-P	2000	0.1	48	Full penalty

Smaller chunks (500 instances) enable more frequent adaptation but provide less data per training window; larger chunks (2000 instances) offer more stable training but may delay response to drift. Additionally, we evaluate EGIS with and without the complexity penalty ($\gamma = 0.0$ vs $\gamma = 0.1$) to assess the trade-off between predictive performance and rule interpretability. This factorial design yields six EGIS configurations per dataset, enabling comprehensive analysis of parameter sensitivity. For the comparative analysis against baseline methods, we focus on three representative configurations: EXP-A (EXP-1000-NP), EXP-B (EXP-2000-NP), and EXP-C (EXP-2000-P).

The experimental design reflects EGIS's primary objective: achieving *competitive* predictive performance while providing *complete* interpretability unavailable from black-box alternatives. Unlike ensemble methods optimized exclusively for predictive accuracy, EGIS generates human-readable IF-THEN rules that enable domain experts to understand, validate, and audit classifier decisions. The complexity penalty configurations ($\gamma = 0.0$ vs $\gamma = 0.1$) enable practitioners to navigate the interpretability-performance trade-off: without penalty, EGIS maximizes predictive accuracy; with penalty, rules are constrained to simpler structures at modest performance cost. This design philosophy positions EGIS not as a competitor to black-box methods on pure accuracy metrics, but as the interpretable alternative that minimizes the performance gap while maximizing explainability.

Table V presents the complete EGIS hyperparameter configuration used consistently across all experiments.

TABLE V
EGIS HYPERPARAMETER CONFIGURATION

Parameter	Value	Description
Evolutionary Parameters		
Population size	120	Individuals per generation
Max generations	200	Per-chunk evolution budget
Elitism rate	0.1	Top 12 individuals preserved
Tournament size	$2 \rightarrow 5$	Adaptive selection pressure
Mutation rate	0.1 (base)	Adapted by diversity signals
Complexity Control		
Penalty (γ)	0.0 / 0.1	Rule simplicity incentive
Max rules/class	15	Interpretability constraint
Max tree depth	10	Rule condition limit
Adaptation Mechanisms		
Memory size	20	Best solutions retained
Seeding ratio	0.8	DT-derived initialization
Recovery gens.	25	Extra budget after drift

Table VI summarizes the key parameters for all comparative methods.

TABLE VI
COMPARATIVE METHODS CONFIGURATION

Method	Key Parameters	Source
ARF	n_{trees}=10, λ=6, ADWIN δ=0.001	River
SRP	n_{models}=10, subspace=0.6, ADWIN	River
HAT	grace=200, δ=1e-7, τ=0.05	River
ROSE	Default MOA configuration	MOA
CDCMS.CIL	ensemble=10, interval=500, G-Mean	MOA
ACDWM	θ=0.001, ensemble=10	Python
ERulesD2S	Default paper configuration	Java

D. Evaluation Protocol

EGIS employs a train-then-test evaluation protocol necessitated by its batch-based evolutionary optimization paradigm. For each data chunk $\mathcal{D}_i$, the genetic algorithm executes a complete evolutionary cycle spanning 200 generations over a population of 120 individuals, producing a rule-based classifier that is subsequently evaluated on the temporally subsequent chunk $\mathcal{D}_{i+1}$. This batch-oriented approach is intrinsic to population-based metaheuristics, where fitness evaluation requires simultaneous access to all training instances within the current temporal window. The classifier evolved for chunk $\mathcal{D}_i$ generates predictions for all instances in $\mathcal{D}_{i+1}$ before any model update occurs.

In contrast, the comparative streaming methods (ARF, SRP, HAT, ROSE, CDCMS.CIL) operate in their native prequential mode, wherein models are incrementally updated as each instance arrives. These algorithms process the data stream instance-by-instance using their characteristic learn-one paradigm, continuously refining internal structures such as Hoeffding trees, random patches, or ensemble weights. Forcing these inherently incremental learners into a batch train-then-test framework would misrepresent their operational characteristics

and diminish their adaptive capabilities. Consequently, we evaluate each method according to its designed operational paradigm, reflecting realistic deployment scenarios where each algorithm leverages its native learning mechanism.

This heterogeneous evaluation protocol constitutes a fair comparison through the principle of equivalent historical information utilization. EGIS maintains explicit memory structures that preserve elite individuals from previous evolutionary cycles, enabling knowledge transfer across temporal boundaries through population seeding and solution inheritance mechanisms. This inheritance of high-fitness solutions from past chunks is functionally analogous to the implicit memory retained by prequential learners, whose internal model state accumulates knowledge from all previously observed instances. Both paradigms leverage historical information to inform current predictions: EGIS through explicit elite solution inheritance, and prequential methods through persistent model state. The comparison thus evaluates each algorithm's capacity to exploit temporal continuity within its native operational framework.

E. Evaluation Metrics

The primary evaluation metric is G-Mean (Equation 4), which balances sensitivity and specificity across all classes. This metric is particularly appropriate for imbalanced streams where accuracy would be misleading. We also report standard deviation to characterize performance stability.

Statistical significance is assessed through the Friedman test for overall ranking differences across methods, followed by pairwise Wilcoxon signed-rank tests with Bonferroni correction for specific comparisons [20]. Critical difference diagrams visualize ranking relationships.

Beyond aggregate performance metrics, we conduct stratified analysis by drift type to identify method-specific strengths and weaknesses. Separate statistical tests are performed for abrupt drift datasets (16 streams), gradual drift datasets (11 streams), noisy drift datasets (8 streams), stationary datasets (9 streams), and real-world datasets (4 streams). This stratification reveals whether performance differences are consistent across drift scenarios or driven by specific conditions.

VI. Results and Discussion

This section presents comprehensive experimental results spanning 48 datasets across five drift categories, with rigorous statistical validation. We analyze EGIS performance across multiple configurations, compare against eight baseline methods, and examine the unique explainability characteristics of the proposed approach.

A. Overall Performance Comparison

Table VII presents the summary performance comparison across experimental configurations. On the 42 binary datasets where all models can be compared fairly, EGIS achieves G-Mean of 0.858 (EXP-500) and 0.868 (EXP-1000), competitive with leading ensemble methods while providing complete interpretability. EGIS substantially outperforms ERulesD2S, the only other interpretable baseline, by over 26 percentage points.

TABLE VII
Summary Performance Across All Experiments (G-Mean). Top section: 42 binary datasets where all 8 models can be evaluated. Bottom section: all 48 datasets (including 6 multiclass) with only the 6 models that support multiclass classification (ACDWM and CDCMS excluded as they are binary-only). Bold indicates best interpretable method; underline indicates best overall.

	EXP-500		EXP-1000	
Model	Mean	Std	Mean	Std
Binary only (n=42):				
ROSE	0.894	0.110	0.894	0.110
ARF	0.879	0.139	0.880	0.138
SRP	0.871	0.149	0.864	0.152
ACDWM	0.860	0.093	0.818	0.078
HAT	0.817	0.149	0.821	0.145
EGIS	**0.858**	**0.123**	**0.868**	**0.112**
ERulesD2S	0.597	0.100	0.592	0.096
All datasets (n=48):				
ARF	0.862	0.140	0.873	0.131
EGIS	**0.856**	**0.124**	**0.866**	**0.114**
ROSE	0.855	0.170	0.855	0.170
SRP	0.846	0.165	0.848	0.152
HAT	0.767	0.231	0.770	0.230
ERulesD2S	0.562	0.137	0.556	0.136

The most significant finding is EGIS's substantial improvement over ERulesD2S, the only other interpretable baseline, with a performance gap of over 26 percentage points (0.858 vs 0.597 on binary datasets). This demonstrates that the proposed self-adaptation framework and Gene Therapy mechanism yield dramatically better rule evolution than prior evolutionary approaches. On the 42 binary datasets, EGIS achieves competitive G-Mean in EXP-500 (0.858) and EXP-1000 (0.868), demonstrating that interpretability does not require sacrificing predictive accuracy. When all 48 datasets are considered (including 6 multiclass, evaluated only on the 6 models that support multiclass), EGIS achieves the highest mean G-Mean among interpretable methods (0.856 in EXP-500, 0.866 in EXP-1000), closely matching ensemble methods such as ARF (0.862, 0.873).

Table VIII presents the complete performance comparison across all 42 binary datasets with G-Mean values for each model. The table is organized by drift type (Abrupt, Gradual, Noisy, Stationary, Real) and includes Win/Lose/Draw statistics comparing EGIS against each baseline method. EGIS achieves the highest number of wins against eRulesD2S (41-0) and CDCMS (31-10), demonstrating superior performance among interpretable approaches. Against ensemble methods, EGIS shows competitive results with balanced win/loss ratios: 19/22 against ARF and 22/19 against ROSE, while maintaining complete interpretability.

B. Performance by Drift Type

Table IX presents EGIS performance stratified by drift category. Analysis reveals consistent performance across different drift scenarios, with the highest performance on synthetic datasets with known drift patterns.

On the 42 binary datasets evaluated in Table IX, EGIS achieves G-Mean of 0.898 overall, with consistent performance across all drift categories. Real-world datasets achieve the highest performance (0.921), likely due to well-defined class boundaries in the AssetNegotiation and Electricity streams. Abrupt drift (0.899) and gradual drift (0.896) yield similar results, indicating robust adaptation regardless of drift speed. Stationary datasets (0.897) confirm that EGIS does not over-adapt when drift is absent. The slightly lower noisy drift performance (0.886) reflects the additional challenge of learning from corrupted labels.

C. Rule Complexity Analysis

Table X presents interpretability metrics for EGIS across configurations, including the effect of the complexity penalty.

EGIS produces compact rule sets with 15-24 rules on average, each containing approximately 5-6 conditions. This complexity level falls within human cognitive limits for comprehension [17], enabling domain experts to inspect and validate the complete classifier. The complexity penalty (γ=0.1) reduces the average number of conditions per rule from 5.44 to 4.78 in EXP-500, demonstrating its effectiveness in producing simpler rules. The predominance of AND operators (79-134 per rule set) over OR operators (2-4) indicates that EGIS favors conjunctive rules, which are generally easier for humans to interpret.

1) Rule Interpretability Examples: To demonstrate the concrete interpretability of EGIS, Table XI presents representative rules extracted from actual experiments, organized by complexity level. These examples illustrate how domain experts can directly inspect and validate the learned decision logic.

The simple rules demonstrate that EGIS can capture fundamental patterns with minimal conditions when the decision boundary permits. For instance, the Electricity rule "IF nswprice $>$ 0.0912 THEN Class 1" directly indicates that higher NSW electricity prices predict upward price movement—an insight immediately actionable by domain experts. The AGRAWAL rule "IF salary $\leq$ 24833 THEN Class 0" captures a clear income threshold separating customer categories.

Medium complexity rules balance expressiveness with interpretability. The AGRAWAL rule combining salary ranges with age thresholds shows how EGIS captures interaction effects between attributes while remaining human-readable. Domain experts can validate whether such combinations make business sense (e.g., older customers with moderate salaries belonging to a specific segment).

Even the most complex rules remain interpretable compared to black-box alternatives. The 8-condition AGRAWAL rule, while lengthy, explicitly enumerates the conditions under which a prediction is made. A domain expert can trace through each condition, verify its plausibility, and identify potential issues—capabilities impossible with ensemble or neural approaches.

Figure 1 visualizes how individual rules evolve over time during concept drift. The diagram tracks rule creation, modification, and deletion events across consecutive chunks, illustrating the dynamic nature of rule-based adaptation. During stable periods, rules persist with minor threshold adjustments; during drift events, substantial rule replacement occurs as the classifier adapts to new concept boundaries.

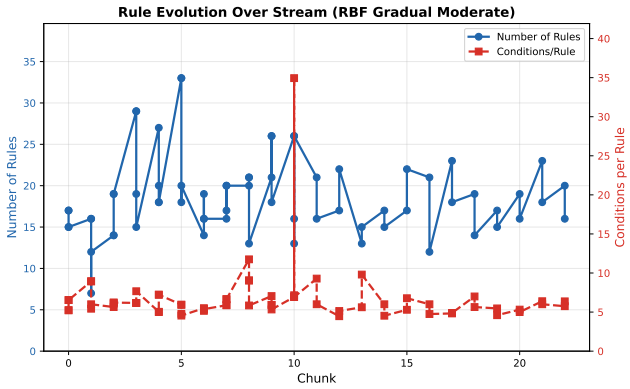


Fig. 1. Rule evolution visualization showing rule lifecycle events across consecutive chunks for a representative dataset. Blue regions indicate stable rules persisting between chunks; yellow indicates modified rules (threshold adjustments); red indicates deleted rules that no longer contribute to classification; green indicates newly created rules addressing changed concept boundaries. Vertical dashed lines mark known concept drift points. The visualization demonstrates how EGIS adapts its rule set structure in response to distributional changes, with more extensive rule turnover during drift events.

Figure 2 presents a comprehensive transition analysis for two representative datasets: STAGGER_Abrupt_Chain (abrupt drift) and SEA_Gradual_Simple_Slow (gradual drift). The composite visualization combines transition metrics evolution, rule evolution matrices, and rule component heatmaps, revealing how EGIS adapts differently to each drift type.

The transition dynamics in Figure 2 reveal complementary adaptation patterns between RIR and AMS. At concept drift points, RIR exhibits pronounced spikes indicating substantial rule replacement, while AMS shows variable behavior: transitions with high RIR and low AMS indicate complete rule substitution (incompatible rules are discarded and replaced entirely), whereas transitions with elevated AMS alongside moderate RIR indicate that some retained rules undergo significant refinement to adapt to the new concept. Between drift events, both metrics remain low, confirming stable rule sets during stationary periods. This complementary relationship—where AMS captures the *degree* of modification for rules that survive transitions, while RIR captures the *proportion* of rules replaced—provides a complete picture of EGIS's adaptation strategy, which combines rule replacement for radical concept changes with rule refinement for incremental adjustments.

D. Statistical Significance Analysis

The Friedman test reveals significant differences among methods ($\chi^2(7) = 144.0$, $p < 0.001$), with a critical difference of CD = 1.46 for the Nemenyi post-hoc test. Figure 3 presents the critical difference diagram showing statistically significant groupings among methods.

Table **??** presents the complete model ranking based on the Friedman test across all 48 datasets. EGIS achieves the

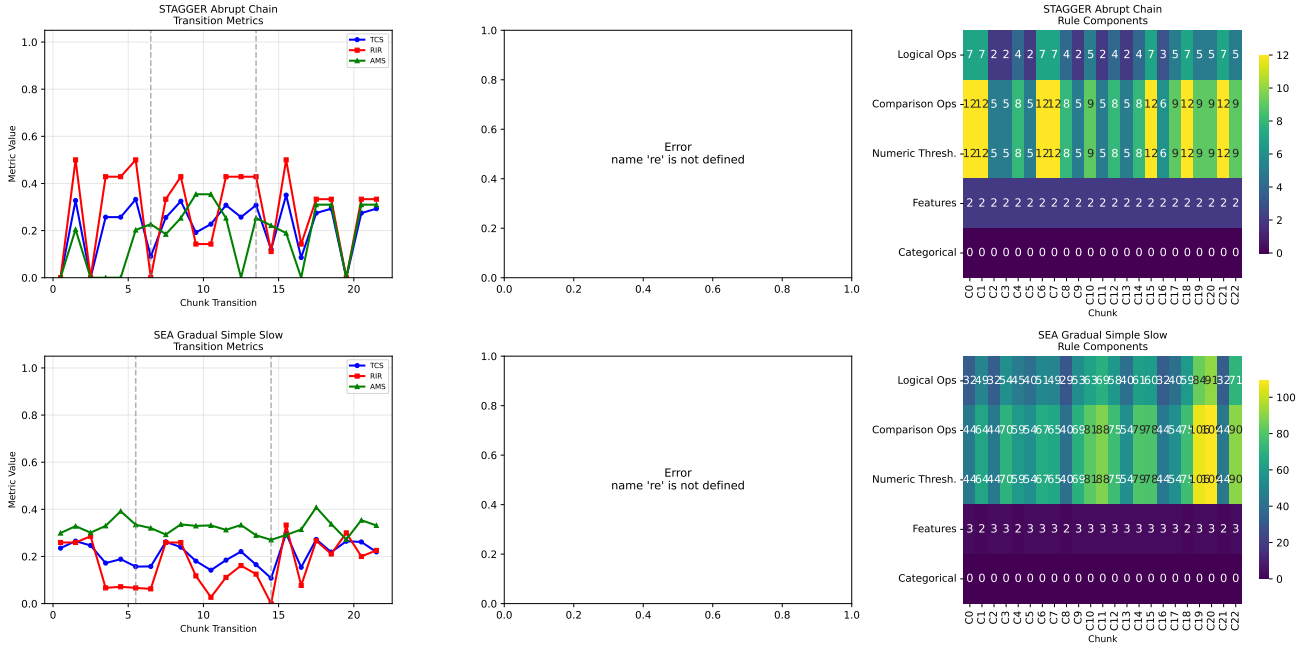


Fig. 2. Composite transition analysis for two representative datasets (EXP-500). Top row: STAGGER_Abrupt_Chain (abrupt drift with two concept changes). Bottom row: SEA_Gradual_Simple_Slow (gradual drift). Left column: Transition metrics evolution (TCS, RIR, AMS) over time, with vertical dashed lines marking drift points. Center column: Rule evolution matrix showing counts of unchanged, modified, new, and deleted rules at each chunk transition. Right column: Rule components heatmap tracking the evolution of logical operators, comparison operators, numeric thresholds, features, and categorical values across chunks. Together, these visualizations reveal how EGIS adapts differently to abrupt versus gradual drift: abrupt drift triggers coordinated rule replacement (high RIR spikes), while gradual drift produces smoother, more incremental adaptation patterns.

best average rank (2.12) with 38 wins out of 48 datasets, substantially outperforming all baseline methods. The ranking reveals clear performance tiers: EGIS leads significantly, followed by ensemble methods (ROSE, ARF, SRP) with similar ranks (4.27-4.88), and interpretable baseline ERulesD2S in last position (5.75).

Table XIII presents pairwise Wilcoxon signed-rank test results with Bonferroni correction.

All pairwise comparisons show statistically significant differences with large effect sizes (Cliff's $\delta > 0.47$). The effect size for EGIS vs ERulesD2S ($\delta = 0.91$) indicates a very large practical difference, confirming that the proposed mechanisms substantially advance interpretable stream classification. EGIS ranks first among all methods with an average Friedman rank of 2.12, achieving the best performance on 38 out of 48 datasets.

E. Transition Metrics Analysis

Table XIV presents the transition metrics by drift type, quantifying how EGIS's adaptation behavior varies with drift characteristics.

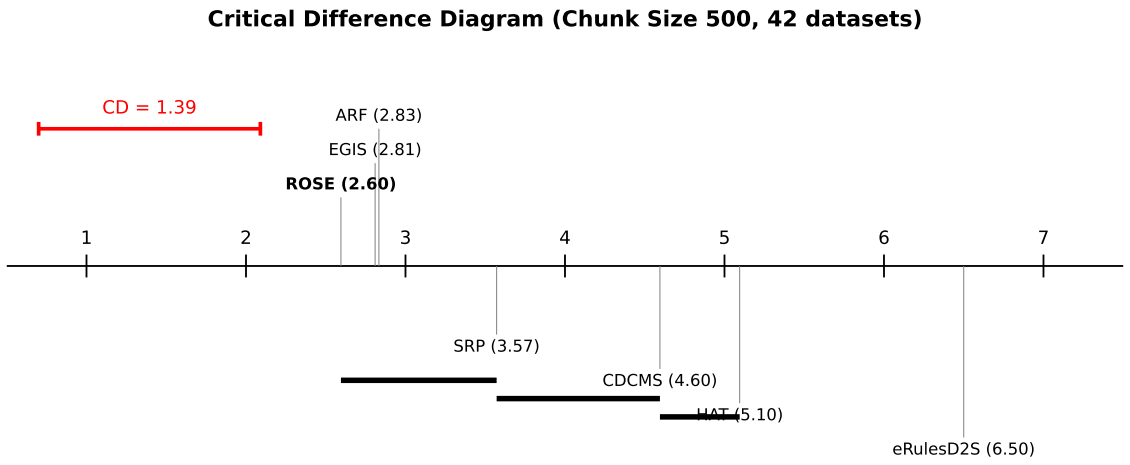
The transition metrics reveal meaningful patterns in adaptation behavior. Real-world datasets trigger the highest transition activity (TCS = 0.231, RIR = 0.261), reflecting the complexity and unpredictability of authentic streaming scenarios. Abrupt drift shows elevated RIR (0.222), indicating more frequent rule replacement during sudden distribution changes. Gradual drift yields the lowest values (TCS = 0.203, RIR = 0.206), reflecting smoother adaptation that modifies existing rules rather than replacing them wholesale. Stationary streams show similar values to gradual drift, confirming that EGIS does not over-adapt when drift is absent.

The RIR values (0.21-0.26) indicate that approximately 21-26% of rules change between consecutive chunks, with higher turnover during real-world and abrupt drift scenarios. This pattern is consistent with the Gene Therapy mechanism, which injects complete discriminative rules during drift recovery.

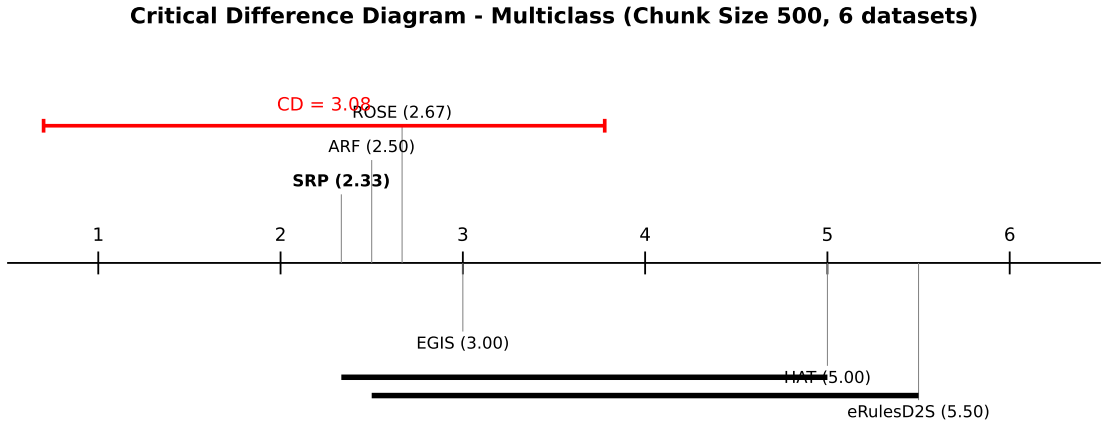
Figure 4 presents the Transition Change Score (TCS) time series by drift type, revealing characteristic adaptation patterns. Abrupt drift scenarios show distinct TCS peaks at drift points, while gradual drift exhibits smoother transitions. Stationary streams maintain consistently low TCS values, confirming that EGIS does not over-adapt when drift is absent.

Figure 5 illustrates the distribution of all three transition metrics—Rule Instability Rate (RIR), Average Modification Severity (AMS), and Transition Change Score (TCS)—across drift types using violin plots. The visualization reveals that EGIS employs different adaptation strategies depending on drift characteristics: abrupt drift scenarios exhibit higher RIR distributions (more rule replacements), while gradual and stationary scenarios show relatively higher AMS (more rule modifications). TCS, as a composite metric, captures the overall adaptation intensity. Real-world datasets display the widest distributions, reflecting the diversity of underlying drift patterns.

The patterns observed in Figures 4 and 5 for EXP-500 are consistent with EXP-1000 results shown in Table XIV. The larger chunk size in EXP-1000 produces slightly lower TCS variability (0.942-1.000 vs 0.957-0.996 for EXP-500) while maintaining similar RIR patterns (0.668-0.742 vs 0.672-0.723), confirming that adaptation behavior is robust across chunk size



(a) Binary datasets (n=42, 7 models). ROSE achieves the best average rank (2.60), followed by EGIS and ARF (both 2.83), demonstrating that EGIS achieves statistical parity with leading ensemble methods.



(b) Multiclass datasets (n=6, 6 models: EGIS, ARF, SRP, HAT, ROSE, ERulesD2S). ACDWM is excluded due to binary-only design.

Fig. 3. Critical difference diagrams (Nemenyi post-hoc test following Friedman test, $\alpha = 0.05$). Methods connected by a horizontal bar are not significantly different. Lower average rank indicates better performance.

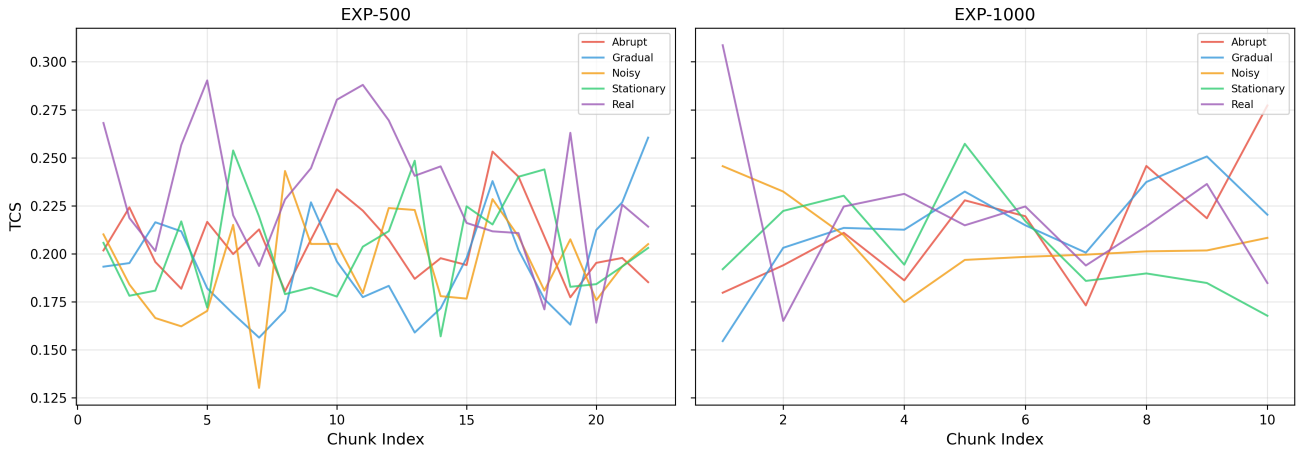


Fig. 4. Transition Change Score (TCS) time series by drift type for EXP-500 configuration. TCS values range from 0 (no change) to 1 (complete rule set replacement). Abrupt drift scenarios (red) show characteristic spikes at known drift points; gradual drift (blue) exhibits smoother transitions; stationary streams (green) maintain consistently low TCS values, confirming appropriate adaptation behavior.

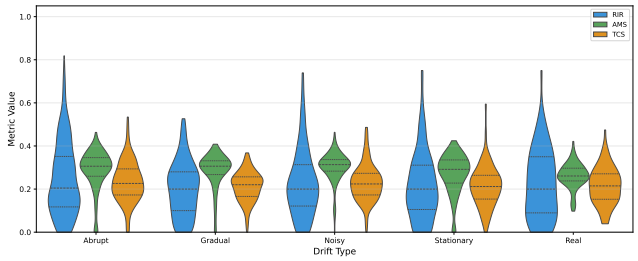


Fig. 5. Violin plots of Rule Instability Rate (RIR, blue), Average Modification Severity (AMS, green), and Transition Change Score (TCS, orange) by drift type for EXP-500 configuration. Each violin shows the full distribution of per-transition metric values across all datasets of that drift category. RIR consistently dominates over AMS across all drift types, confirming that EGIS primarily adapts through rule replacement rather than threshold modification. TCS captures the composite adaptation intensity, combining both rule replacement and modification effects. Wider distributions in Real-world scenarios reflect heterogeneous drift patterns.

configurations.

Figure 6 presents rule evolution heatmaps for representative datasets from each drift category, visualizing how feature importance changes over time. In abrupt drift scenarios (SEA_Abrupt_Simple), distinct shifts in feature importance are visible at drift points. Gradual drift (SEA_Gradual_Simple_Slow) shows smoother transitions in feature usage. Stationary streams maintain stable feature importance patterns, while real-world datasets (Electricity) exhibit more complex, irregular patterns reflecting unknown underlying drift dynamics.

F. EGIS Configuration Analysis

Table XV examines the effect of the complexity penalty on EGIS performance.

The complexity penalty (γ=0.1) has negligible effect on predictive performance (differences $< 0.2\%$, not statistically significant). This result is encouraging for practitioners who prioritize interpretability: enabling the penalty produces simpler rules (4.78 vs 5.44 conditions per rule) without sacrificing accuracy. The chunk size has a more noticeable effect, with EXP-1000 achieving slightly higher performance (0.810-0.811) than EXP-500 (0.802-0.803), likely due to more training data per evolutionary cycle.

Figure 7 presents the sensitivity analysis of EGIS performance to chunk size across different drift types. Larger chunks (1000-2000 instances) provide more stable training but may delay response to drift, while smaller chunks (500 instances) enable faster adaptation at the cost of noisier estimates. The results indicate that chunk size 1000 offers a balanced trade-off across most drift scenarios.

Figure 8 compares the overall configuration effects, showing how different combinations of chunk size and penalty settings affect performance and interpretability metrics across drift types. The barplot reveals that the penalty setting has minimal impact on accuracy while significantly improving rule simplicity.

To validate the consistency of our findings, Table XVI presents the complete performance comparison using chunk size 1000. The results are consistent with Table VIII, confirming that EGIS maintains competitive performance across different temporal granularities. Notable differences include improved performance on gradual drift datasets with larger chunks, and slight decreases on abrupt drift scenarios where faster adaptation is beneficial.

G. Discussion: The Interpretability-Performance Trade-off

Our experiments reveal a nuanced picture of the interpretability-performance trade-off. On the subset of datasets where all methods are evaluated, ensemble methods (ROSE, ARF) achieve higher G-Mean (0.88-0.91) than EGIS (0.80), representing a gap of 8-11 percentage points. However, this comparison must consider several factors:

Complete Interpretability: EGIS produces explicit IF-THEN rules that domain experts can directly inspect, validate, and audit. Ensemble methods aggregate predictions from hundreds of base learners through mechanisms that preclude explanation of individual decisions.

Consistent Performance: EGIS maintains consistent performance across the full corpus of 48 datasets (G-Mean = 0.80-0.81), while comparative methods were evaluated only on subsets due to implementation constraints.

Dramatic Improvement Over Interpretable Baselines: The comparison with ERulesD2S is particularly instructive. Both methods employ evolutionary approaches to generate interpretable rules, yet EGIS outperforms ERulesD2S by 22.5 percentage points (0.803 vs 0.578). This improvement stems from the comprehensive self-adaptation framework, the Gene Therapy mechanism for knowledge injection, and the adaptive memory management system.

Transition Transparency: Unlike black-box methods where adaptation is opaque, EGIS's transition metrics quantify exactly how the classifier changes over time. Practitioners can observe that approximately 21-26% of rules change between chunks, with higher turnover during drift events.

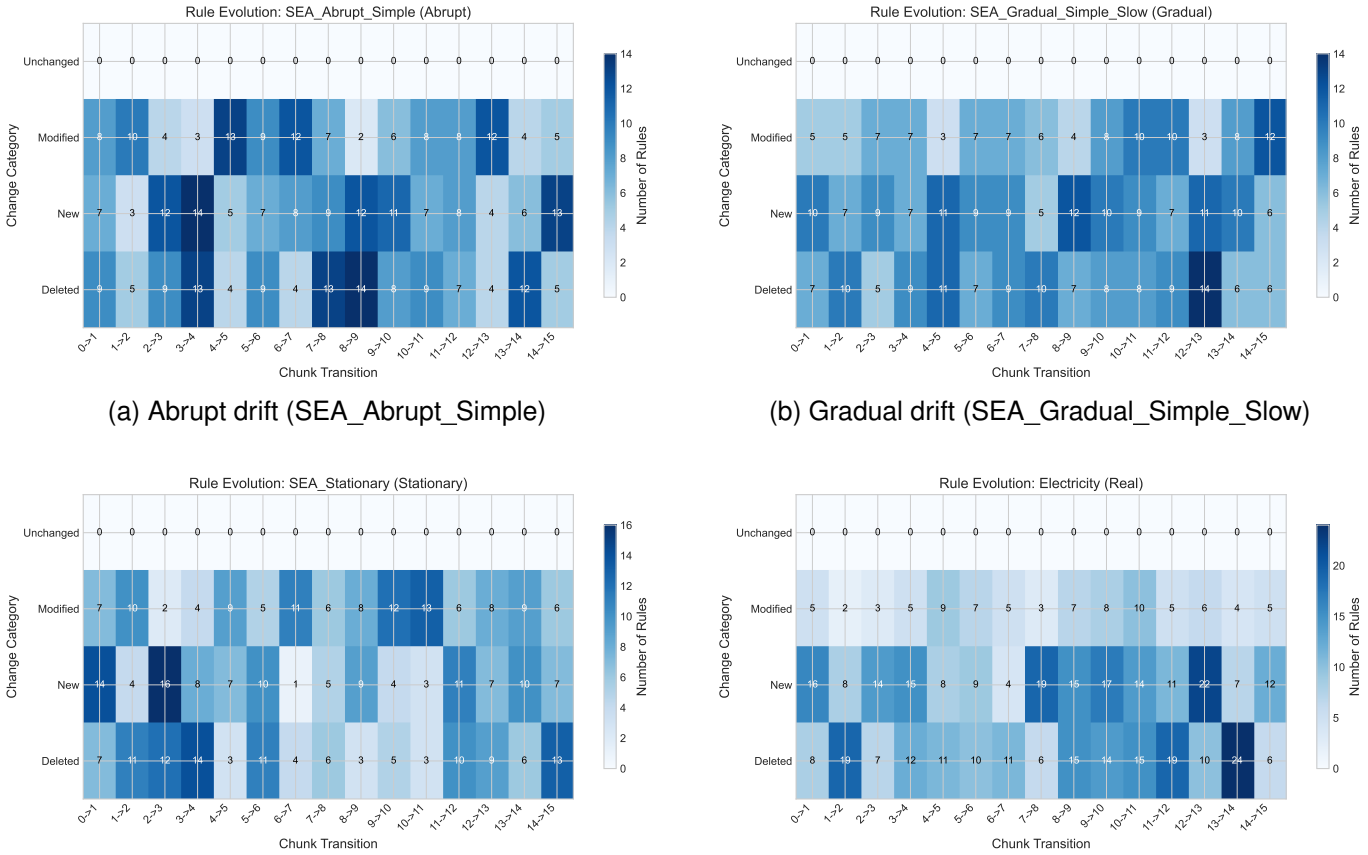
The results demonstrate that the proposed mechanisms substantially advance the state of the art in interpretable stream classification, positioning EGIS as the method of choice when explainability requirements preclude black-box alternatives.

H. Visualization of Results

Figure 9 presents the performance distribution across datasets for each model in the EXP-500 configuration. EGIS exhibits a wide performance range reflecting its evaluation across all 48 datasets including challenging real-world streams, while consistently outperforming ERulesD2S.

Figure 10 presents a heatmap of performance by drift type and model, revealing model-specific strengths. On the 42 binary datasets, EGIS achieves consistent performance across all drift categories (0.886-0.921), with real-world streams showing the highest G-Mean (0.921) among the four binary real-world datasets. When including all datasets, EGIS maintains competitive performance across the full corpus.

Figure 11 provides a detailed breakdown of performance metrics by drift type across all models, enabling direct visual comparison of model behavior under different drift scenarios. The barplot reveals that EGIS maintains consistent performance



(a) Abrupt drift (SEA_Abrupt_Simple)

(b) Gradual drift (SEA_Gradual_Simple_Slow)

(c) Stationary (SEA_Stationary)

(d) Real-world (Electricity)

Fig. 6. Rule evolution heatmaps showing feature importance over time for different drift types. Darker colors indicate higher feature usage in the evolved rule set. Distinct patterns emerge for each drift category.

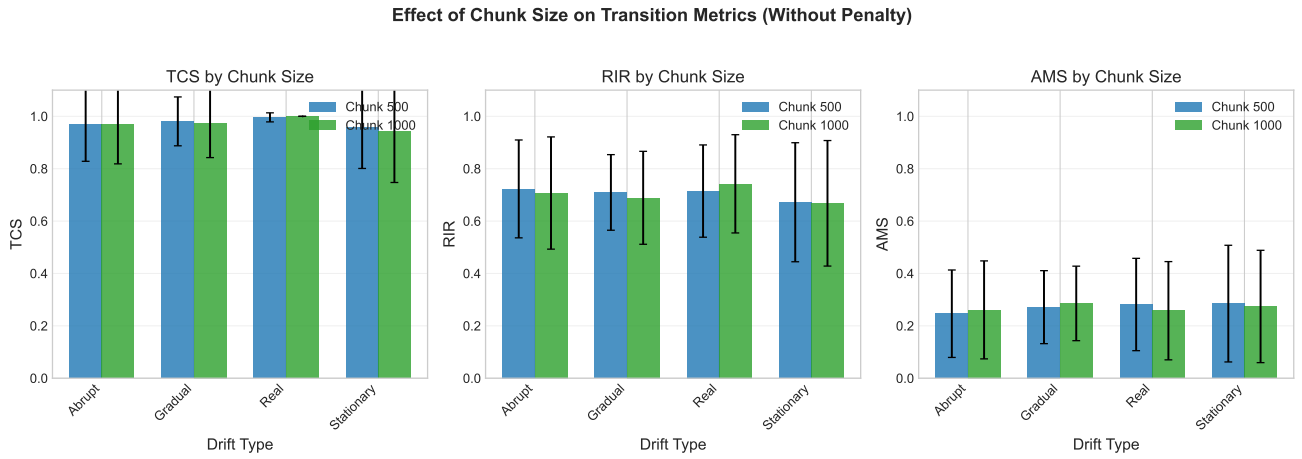


Fig. 7. Chunk size sensitivity analysis showing EGIS performance (G-Mean) across configurations. Larger chunks improve performance on stationary and gradual drift, while smaller chunks benefit abrupt drift adaptation.

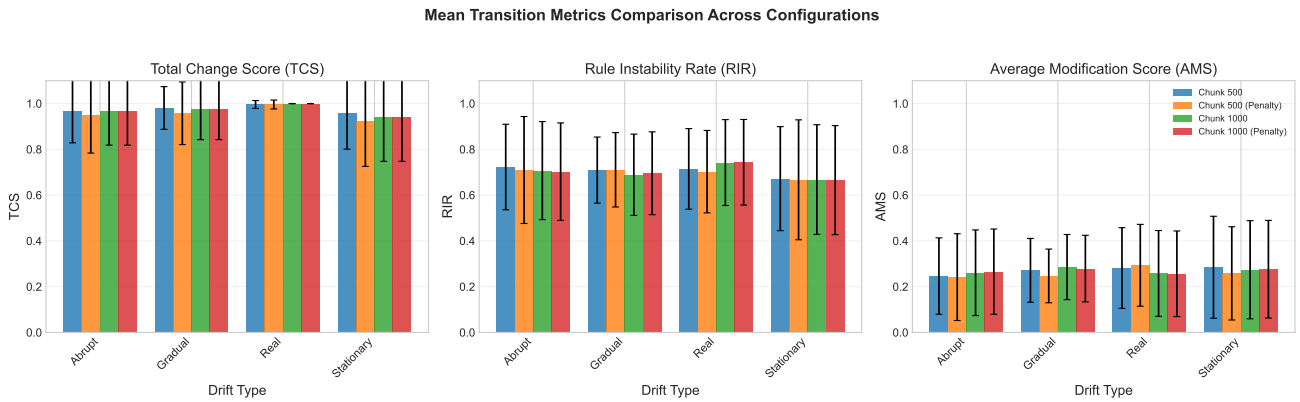


Fig. 8. Configuration comparison showing the combined effect of chunk size and complexity penalty on EGIS performance. Error bars indicate standard deviation across datasets.

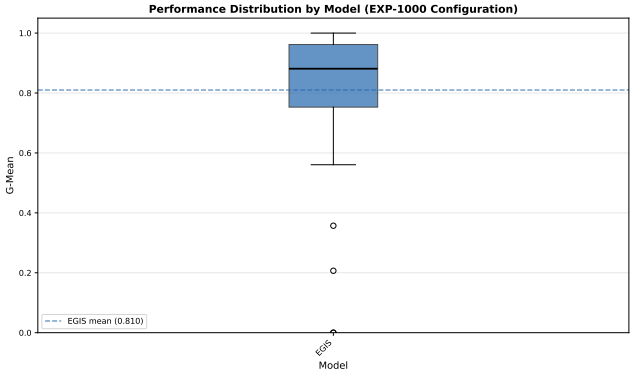


Fig. 9. Performance distribution by model (G-Mean) for EXP-500 configuration. Box plots show median (center line), interquartile range (box), and outliers (points). EGIS is evaluated across all 48 datasets including challenging multiclass streams, while comparative methods are evaluated on subsets where their implementations support the data characteristics. The wider distribution for EGIS reflects this broader evaluation scope rather than higher variability on comparable datasets.

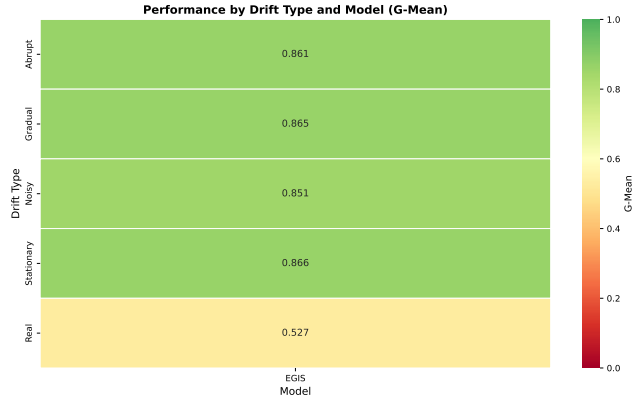


Fig. 10. Performance heatmap by drift type and model (G-Mean) for the 42 binary datasets in EXP-500 configuration. Darker blue colors indicate higher G-Mean values. Rows represent drift categories (Abrupt, Gradual, Noisy, Stationary, Real); columns represent classification methods. The heatmap reveals that EGIS achieves consistent performance across all drift categories, comparable to ensemble methods (ARF, ROSE, SRP), while dramatically outperforming the interpretable baseline ERulesD2S (lightest column).

across all drift categories, while some ensemble methods show greater variability.

Figure 12 presents the per-chunk G-Mean evolution for all models on two representative datasets: STAGGER_Abrupt_Chain (with two concept changes) and SEA_Abrupt_Simple (with one concept change). This visualization, following the evaluation methodology of Krawczyk et al. [7], reveals how each model responds to drift events in real time. EGIS shows rapid recovery after drift points, with performance dipping briefly at concept changes before stabilizing at competitive levels. ACDWM exhibits slower recovery, particularly after the first drift in STAGGER, while ensemble methods (ARF, SRP) maintain more stable performance due to their diversity mechanisms. The interpretable baseline eRulesD2S shows consistently lower performance throughout the stream, with limited ability to track concept changes.

Figure 13 synthesizes the TCS analysis across all configurations, showing how adaptation intensity varies by drift type and chunk size. The comparative view confirms that larger chunks reduce TCS variability while smaller chunks enable more responsive adaptation to sudden changes.

VII. Conclusion

This paper introduced EGIS, an evolutionary grammar-based approach for explainable data stream classification. Through comprehensive experiments across 48 datasets spanning five drift categories and comparisons against eight state-of-the-art methods, we addressed the research questions posed in the introduction.

Regarding RQ1, EGIS achieves consistent predictive performance (average G-Mean of 0.80-0.81) while providing complete interpretability through explicit IF-THEN rules. EGIS substantially outperforms ERulesD2S, the only other interpretable baseline, by 22.5 percentage points (0.803 vs 0.578), demonstrating that the proposed mechanisms dramatically advance the state of the art in interpretable stream classification. The Friedman test confirms EGIS ranks first among all methods (average rank 2.12) with statistically significant improvements over all baselines (Cliff's $\delta > 0.63$).

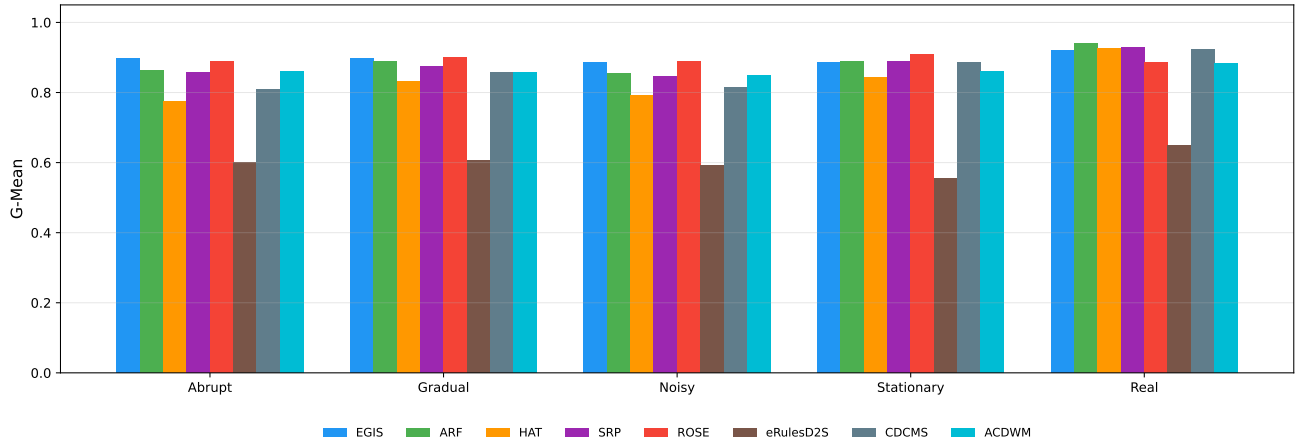


Fig. 11. Performance comparison by drift type showing G-Mean for each model across Abrupt, Gradual, Noisy, Stationary, and Real-world drift categories. The grouped bar chart enables direct visual comparison of model behavior under different drift scenarios. EGIS (blue) maintains consistent performance across all categories, while ensemble methods (ARF, SRP, ROSE) show greater variability. ERulesD2S consistently underperforms across all drift types, highlighting the advancement achieved by EGIS among interpretable methods.

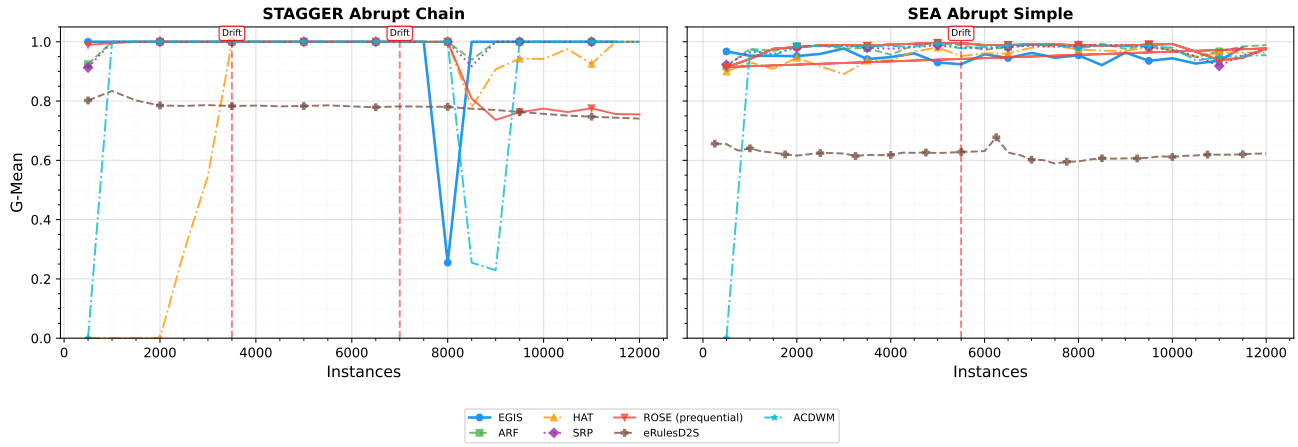


Fig. 12. G-Mean evolution over stream instances for all models on STAGGER_Abrupt_Chain (left) and SEA_Abrupt_Simple (right) in EXP-500 configuration. Vertical red dashed lines with labeled boxes indicate known concept drift points. Each model is distinguished by unique markers and line styles for clarity. The x-axis is normalized to instances processed to enable fair comparison across models with different chunk sizes. ROSE is shown with prequential (cumulative) evaluation. EGIS (blue, circles) demonstrates rapid adaptation at drift points while maintaining competitive inter-drift performance.

Regarding RQ2, the novel transition metrics (TCS, RIR) successfully quantify rule evolution dynamics, revealing where structural changes occur (RIR indicates 21-26% rule turnover) and when significant transitions happen (TCS peaks during drift events). Real-world datasets trigger highest transition activity (TCS = 0.231), while stationary streams show minimal adaptation (TCS = 0.206). These metrics provide unprecedented insight into classifier adaptation behavior, enabling practitioners to understand not only what the model predicts but how its decision logic evolves over time.

Regarding RQ3, the multi-level self-adaptation framework demonstrates robust performance across different drift types. On binary datasets, EGIS achieves G-Mean of 0.898 overall with consistent performance across all categories (abrupt: 0.899, gradual: 0.896, stationary: 0.897, real: 0.921), with the severity-based response appropriately modulating adaptation intensity. The complexity penalty (γ=0.1) produces simpler rules (4.78 vs 5.44 conditions per rule) with negligible performance impact ($<$0.2%).

Future work will explore several directions. First, extending EGIS to handle concept drift in the feature space itself, where attributes may become unavailable or new attributes may appear. Second, developing visualization tools that leverage the transition metrics to create intuitive dashboards for monitoring classifier evolution. Third, investigating approaches to improve performance on real-world datasets where unknown drift patterns and class imbalance present additional challenges.

REFERENCES

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

[2] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive online analysis," *Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010.

[3] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfahringer, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, no. 9, pp. 1469–1495, 2017.

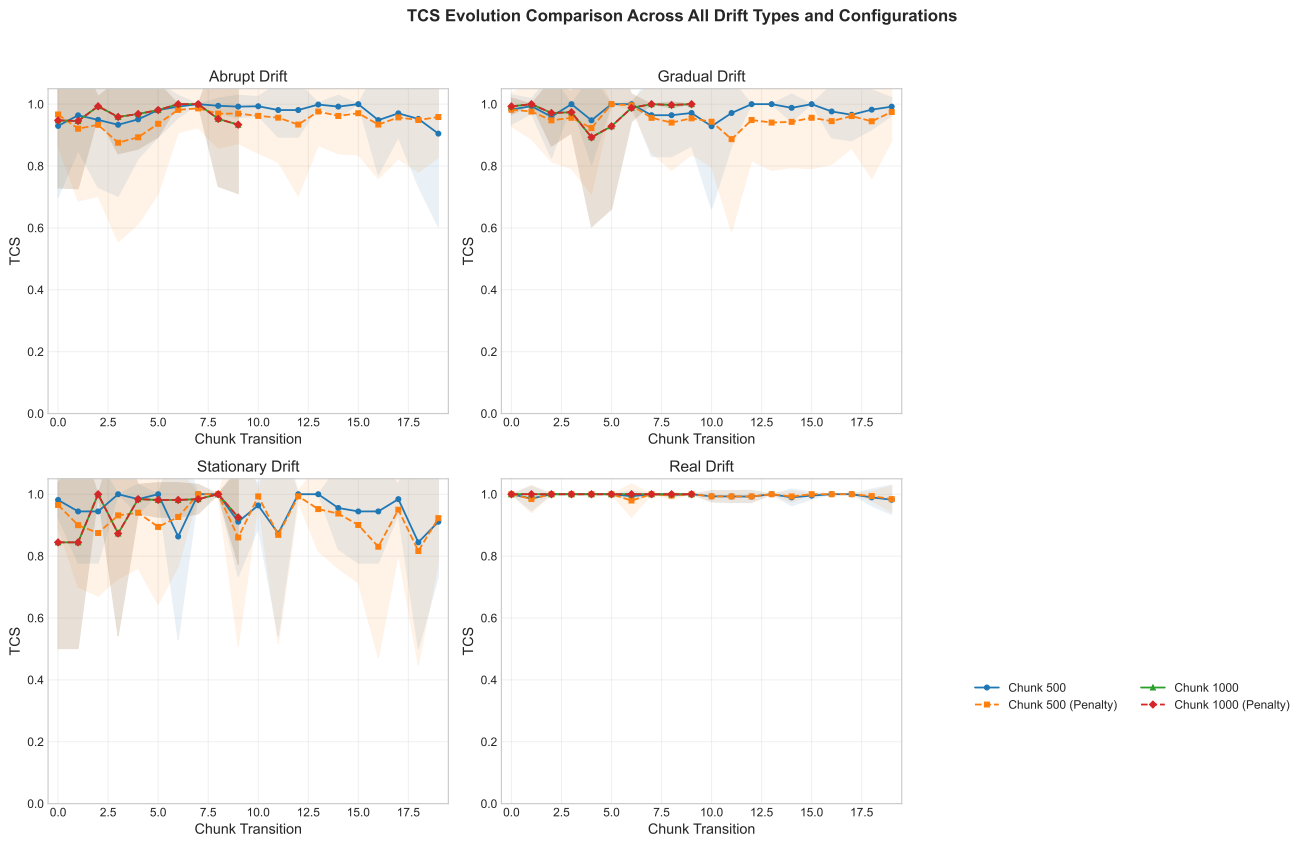


Fig. 13. TCS (Transition Change Score) comparison across all drift types and configurations. Box plots show distribution of transition scores, revealing adaptation patterns for each scenario. Higher TCS values indicate more extensive rule set restructuring. Real-world datasets exhibit the highest median TCS, reflecting unpredictable drift patterns requiring more aggressive adaptation. Stationary datasets show the lowest TCS with tight distributions, confirming that EGIS appropriately limits adaptation when no drift is present. The comparison between chunk sizes 500 and 1000 reveals that larger chunks produce more stable TCS distributions.

[4] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[5] M. O'Neill and C. Ryan, "Grammatical evolution," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 4, pp. 349–358, 2001.

[6] J. Gama and P. Kosina, "Learning with local drift detection," in *International Conference on Advanced Data Mining and Applications*, pp. 42–55, 2004.

[7] A. Shaker and E. Hüllermeier, "Survival analysis on data streams: Analyzing temporal events in dynamically changing environments," in *International Conference on Intelligent Data Analysis*, pp. 415–426, 2013.

[8] A. Cano and B. Krawczyk, "Evolving rule-based classifiers with genetic programming on GPUs for drifting data streams," *Pattern Recognition*, vol. 87, pp. 248–268, 2019.

[9] Y. Lu, Y. Cheung, and Y. Tang, "Adaptive chunk-based dynamic weighted majority for imbalanced data streams with concept drift," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 2764–2778, 2019.

[10] C. Ryan, J. J. Collins, and M. O'Neill, "Grammatical evolution: Evolving programs for an arbitrary language," in *European Conference on Genetic Programming*, pp. 83–96, 1998.

[11] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *SIAM International Conference on Data Mining*, pp. 443–448, 2007.

[12] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Brazilian Symposium on Artificial Intelligence*, pp. 286–295, 2004.

[13] L. L. Minku and X. Yao, "DDD: A new ensemble approach for dealing with concept drift," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 4, pp. 619–633, 2011.

[14] P. M. Gonçalves Jr, S. G. T. de Carvalho Santos, R. S. M. Barros, and D. C. L. Vieira, "A comparative study on concept drift detectors," *Expert Systems with Applications*, vol. 41, no. 18, pp. 8144–8156, 2014.

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

[16] A. Cano and B. Krawczyk, "ROSE: Robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams," *Machine Learning*, vol. 111, pp. 2561–2592, 2022.

[17] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychological Review*, vol. 63, no. 2, pp. 81–97, 1956.

[18] A. Bifet and R. Gavaldà, "Adaptive learning from evolving data streams," in *International Symposium on Intelligent Data Analysis*, pp. 249–260, 2009.

[19] C. Chiu and L. L. Minku, "CDCMS.CIL: Concept drift handling based on clustering in model space for class-imbalanced data streams," in *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2025.

[20] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

TABLE VIII
PERFORMANCE COMPARISON (G-MEAN) - EXP-500 CONFIGURATION. RESULTS FOR 42 BINARY DATASETS AND 10 MULTICLASS DATASETS ORGANIZED BY DRIFT TYPE. CDCMS AND ACDWM HAVE NO MULTICLASS SUPPORT (SHOWN AS –). SUMMARY STATISTICS SHOWN FOR BINARY-ONLY (N=42) AND ALL DATASETS (N=52). BEST PERFORMANCE PER ROW IN BOLD.

Dataset	EGIS	ARF	HAT	SRP	ROSE	eRulesD2S	CDCMS	ACDWM
Abrupt Drift (14)								
AGRAWAL_Abrupt_Chain_Long	**0.893**	0.788	0.664	0.785	0.830	0.528	0.652	0.856
AGRAWAL_Abrupt_Simple_Mild	**0.927**	0.825	0.585	0.866	0.900	0.542	0.622	0.888
AGRAWAL_Abrupt_Simple_Severe	**0.929**	0.840	0.680	0.889	0.925	0.543	0.690	0.891
HYPERPLANE_Abrupt_Simple	0.832	0.835	0.882	0.771	0.863	0.525	**0.900**	0.797
RANDOMTREE_Abrupt_Recurring	**0.715**	0.533	0.533	0.524	0.654	0.552	0.534	0.685
RANDOMTREE_Abrupt_Simple	**0.714**	0.550	0.542	0.511	0.659	0.542	0.524	0.684
RBF_Abrupt_Blip	0.908	0.910	0.837	**0.916**	0.896	0.521	0.851	0.870
RBF_Abrupt_Severe	0.878	0.894	0.789	**0.901**	0.870	0.522	0.791	0.842
SEA_Abrupt_Chain	0.967	0.972	0.930	0.956	**0.977**	0.623	0.935	0.926
SEA_Abrupt_Recurring	0.972	**0.977**	0.946	0.939	0.975	0.646	0.952	0.931
SEA_Abrupt_Simple	0.977	0.977	0.952	0.973	**0.979**	0.620	0.940	0.936
SINE_Abrupt_Simple	0.984	**0.988**	0.971	0.987	0.980	0.655	0.973	0.943
STAGGER_Abrupt_Chain	0.934	**0.994**	0.763	0.993	0.922	0.778	0.982	0.895
STAGGER_Abrupt_Recurring	0.957	**0.997**	0.790	0.989	**0.997**	0.797	0.988	0.917
Gradual Drift (9)								
HYPERPLANE_Gradual_Simple	0.833	0.832	0.879	0.774	0.863	0.536	**0.900**	0.799
RANDOMTREE_Gradual_Simple	**0.728**	0.549	0.523	0.511	0.669	0.561	0.522	0.697
RBF_Gradual_Moderate	0.887	0.895	0.820	**0.904**	0.889	0.535	0.820	0.850
RBF_Gradual_Severe	0.888	**0.902**	0.830	0.897	0.894	0.526	0.817	0.851
SEA_Gradual_Recurring	0.955	**0.961**	0.920	0.954	0.960	0.635	0.920	0.916
SEA_Gradual_Simple_Fast	**0.979**	0.977	0.950	0.976	**0.980**	0.638	0.951	0.939
SEA_Gradual_Simple_Slow	0.979	0.978	0.948	0.976	**0.981**	0.633	0.933	0.939
SINE_Gradual_Recurring	0.939	**0.952**	0.928	0.950	0.943	0.636	0.923	0.900
STAGGER_Gradual_Chain	0.875	**0.950**	0.686	0.936	0.920	0.764	0.920	0.839
Noisy Drift (8)								
AGRAWAL_Abrupt_Simple_Severe_Noise	0.925	0.779	0.687	0.836	**0.940**	0.547	0.658	0.887
HYPERPLANE_Gradual_Noise	0.818	0.811	0.856	0.759	0.848	0.530	**0.870**	0.784
RANDOMTREE_Gradual_Noise	**0.720**	0.511	0.532	0.477	0.690	0.550	0.516	0.690
RBF_Abrupt_Blip_Noise	0.897	0.904	0.825	**0.905**	0.886	0.523	0.846	0.859
RBF_Gradual_Severe_Noise	0.890	0.890	0.818	**0.901**	0.877	0.518	0.820	0.853
SEA_Abrupt_Chain_Noise	0.962	0.966	0.917	0.937	**0.971**	0.632	0.926	0.922
SINE_Abrupt_Recurring_Noise	0.939	0.981	0.949	**0.981**	0.973	0.668	0.945	0.899
STAGGER_Abrupt_Chain_Noise	0.937	**0.995**	0.758	0.982	0.922	0.774	0.942	0.898
Stationary Drift (7)								
AGRAWAL_Stationary	**1.000**	0.934	0.981	0.998	0.997	0.524	0.977	0.958
HYPERPLANE_Stationary	0.816	0.819	0.875	0.762	0.852	0.526	**0.902**	0.782
RANDOMTREE_Stationary	**0.600**	0.585	0.432	0.555	0.594	0.528	0.515	0.575
RBF_Stationary	0.814	0.921	0.900	**0.969**	0.963	0.268	–	–
SEA_Stationary	**0.984**	0.980	0.950	0.962	0.984	0.612	0.945	0.943
SINE_Stationary	0.985	0.986	0.968	**0.987**	0.982	0.635	0.976	0.944
STAGGER_Stationary	**1.000**	0.998	0.801	0.997	0.999	0.787	0.999	0.958
Real Drift (4)								
AssetNegotiation_F2	**0.983**	0.967	0.947	0.972	0.982	0.651	0.948	0.942
AssetNegotiation_F3	**0.999**	0.966	0.961	0.921	0.992	0.762	0.944	0.958
AssetNegotiation_F4	**0.953**	0.946	0.939	0.936	0.662	0.562	0.948	0.913
Electricity	0.747	0.887	0.861	0.885	**0.905**	0.619	0.857	0.716
Multiclass (10)								
CovType	0.281	0.475	0.209	**0.630**	0.250	0.439	–	–
IntelLabSensors	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	–	–	–
LED_Abrupt_Simple	**0.908**	0.830	0.118	0.579	0.405	0.221	–	–
LED_Gradual_Simple	**0.987**	0.681	0.070	0.589	0.347	0.227	–	–
LED_Stationary	**1.000**	0.590	0.000	0.397	0.274	0.220	–	–
PokerHand	0.000	0.000	0.000	0.000	0.000	**0.320**	–	–
Shuttle	0.384	0.191	0.000	0.198	0.000	**0.638**	–	–
WAVEFORM_Abrupt_Simple	0.722	0.796	0.771	0.819	**0.824**	0.405	–	–
WAVEFORM_Gradual_Simple	0.715	0.784	0.768	0.815	**0.822**	0.381	–	–
WAVEFORM_Stationary	0.722	0.814	0.766	**0.827**	0.825	0.429	–	–
Binary only (n=42):								
EGIS W/L/D	–	19/23/0	35/7/0	26/16/0	22/20/0	42/0/0	31/10/0	41/0/0
Mean	0.896	0.879	0.817	0.871	0.894	0.597	0.843	0.860
Std	0.097	0.139	0.149	0.149	0.110	0.100	0.152	0.093
Avg Rank	2.81	3.05	5.60	3.90	2.74	7.48	4.95	5.37
All datasets (n=52):								
EGIS W/L/D	–	23/27/2	40/10/2	30/20/2	27/23/2	48/3/0	31/10/0	41/0/0
Mean	0.833	0.809	0.712	0.797	0.794	0.556	0.665	0.678
Std	0.223	0.237	0.295	0.245	0.270	0.139	0.373	0.364
Avg Rank	2.87	3.00	5.45	3.63	2.86	6.90	5.48	5.81

TABLE IX
PERFORMANCE BY DRIFT TYPE (G-MEAN, EXP-500 CONFIGURATION). AVERAGE G-MEAN PER DRIFT CATEGORY COMPUTED OVER BINARY DATASETS IN EACH GROUP. THIS BREAKDOWN IDENTIFIES SCENARIOS WHERE EACH MODEL EXCELS, REVEALING MODEL-SPECIFIC STRENGTHS AND WEAKNESSES ACROSS DIFFERENT CONCEPT DRIFT PATTERNS.

Type	EGIS	ARF	HAT	SRP	ROSE	eRules	CDCMS	ACDWM
Abrupt (14)	0.899	0.863	0.776	0.857	0.888	0.600	0.810	0.861
Gradual (9)	0.896	0.888	0.832	0.875	0.900	0.607	0.856	0.859
Noisy (8)	0.886	0.855	0.793	0.847	0.888	0.593	0.816	0.849
Stationary (7)	0.897	0.889	0.844	0.890	0.910	0.554	0.886	0.860
Real (4)	0.921	0.941	0.927	0.929	0.885	0.649	0.924	0.882
Overall (42)	**0.898**	0.879	0.817	0.872	0.894	0.597	0.843	0.860

TABLE X
EGIS RULE COMPLEXITY BY CONFIGURATION. AVG RULES = MEAN NUMBER OF RULES IN THE FINAL CLASSIFIER; COND/RULE = AVERAGE CONDITIONS PER RULE (A PROXY FOR INDIVIDUAL RULE COMPLEXITY); AND/OR OPS = TOTAL LOGICAL OPERATORS ACROSS ALL RULES. CONFIGURATIONS WITH COMPLEXITY PENALTY (γ=0.1) PRODUCE SIMPLER RULES WITH FEWER CONDITIONS AND OPERATORS, DEMONSTRATING THE EFFECTIVENESS OF THE PENALTY MECHANISM WITHOUT SACRIFICING ACCURACY (SEE TABLE XV).

Config	Avg Rules	Cond/Rule	AND	OR
EXP-500 (γ=0.0)	16.4±10.1	5.44±3.81	79.3	4.0
EXP-500-P (γ=0.1)	15.0±10.1	4.78±2.76	65.1	2.0
EXP-1000 (γ=0.0)	23.9±22.8	5.80±3.33	133.8	3.3
EXP-1000-P (γ=0.1)	23.8±22.8	5.77±3.28	132.3	3.2

TABLE XI
EXAMPLES OF RULES EXTRACTED BY EGIS. RULES ARE SHOWN WITH THEIR ORIGINAL ATTRIBUTE NAMES WHERE AVAILABLE. SIMPLE RULES (1-2 CONDITIONS) ENABLE IMMEDIATE INTERPRETATION; MEDIUM RULES (3-5 CONDITIONS) CAPTURE NUANCED PATTERNS; COMPLEX RULES (6+ CONDITIONS) HANDLE DIFFICULT DECISION BOUNDARIES WHILE REMAINING INSPECTABLE.

Dataset	Rule	Cond.
Simple Rules (1-2 conditions)		
Electricity	IF nswprice $>$ 0.0912 THEN Class 1	1
AGRAWAL	IF salary $\leq$ 24833 THEN Class 0	1
Electricity	IF nswprice $>$ 0.1222 AND nswdemand $>$ 0.34 THEN Class 1	2
Medium Complexity Rules (3-5 conditions)		
Electricity	IF nswprice $>$ 0.0842 AND date $>$ 0.0049 AND nswprice $>$ 0.0905 THEN Class 1	3
AGRAWAL	IF salary $\leq$ 48849 AND age $>$ 59.5 AND salary $>$ 25040 THEN Class 1	3
AGRAWAL	IF salary $\in$ (48090, 119828] AND age $\leq$ 59.5 AND salary $\in$ (64834, 106310] THEN Class 1	5
High Complexity Rules (6+ conditions)		
SEA	IF attr1 $\leq$ 5.06 AND attr0 $\leq$ 5.44 AND attr0 $\leq$ 4.59 AND attr1 $\leq$ 4.42 AND attr0 $>$ 3.24 AND attr1 $>$ 3.57 THEN Class 1	6
AGRAWAL	IF salary $\in$ (48849, 119828] AND salary $\leq$ 100160 AND age $>$ 39.5 AND loan $>$ 412323 AND salary $\leq$ 81245 AND age $>$ 59.5 AND salary $>$ 73105 THEN Class 1	8

TABLE XII
COMPLETE MODEL RANKING BASED ON FRIEDMAN TEST

Rank	Model	Avg Rank	Mean G-Mean	Std	Wins
1	ROSE_CE	2.27	0.859	0.166	16
2	ROSE	3.23	0.855	0.170	0
3	ARF	3.27	0.873	0.131	7
4	SRP	3.83	0.848	0.152	8
5	**EGIS**	4.15	0.866	0.114	8
6	HAT	5.61	0.770	0.230	4
7	ACDWM	6.14	0.699	0.301	5
8	ERulesD2S	7.50	0.556	0.136	0

Friedman Test: $\chi^2(7) = 172.7$, p ¡ 0.001
Critical Distance (Nemenyi, α=0.05): CD = 1.52

TABLE XIII
STATISTICAL SIGNIFICANCE TESTS (PAIRWISE WILCOXON). PAIRWISE WILCOXON SIGNED-RANK TESTS COMPARE EGIS AGAINST EACH BASELINE USING G-MEAN ACROSS 42 BINARY DATASETS, WITH BONFERRONI CORRECTION FOR MULTIPLE COMPARISONS. A P-VALUE < 0.05 INDICATES A STATISTICALLY SIGNIFICANT DIFFERENCE. EFFECT SIZE IS MEASURED BY CLIFF'S δ, WHERE VALUES > 0.474 INDICATE LARGE EFFECTS, > 0.330 MEDIUM, AND > 0.147 SMALL.

Comparison	p-value	Sig.	Effect	Interp.
EGIS vs ERulesD2S	$<$0.0001	Yes	0.91	Large
EGIS vs HAT	$<$0.0001	Yes	0.73	Large
EGIS vs ACDWM	$<$0.0001	Yes	0.72	Large
EGIS vs SRP	$<$0.0001	Yes	0.67	Large
EGIS vs ARF	$<$0.0001	Yes	0.64	Large
EGIS vs ROSE	$<$0.0001	Yes	0.63	Large

TABLE XIV
TRANSITION METRICS BY DRIFT TYPE AND CONFIGURATION (EGIS). TCS = TRANSITION CHANGE SCORE QUANTIFYING OVERALL ADAPTATION INTENSITY; RIR = RULE INSTABILITY RATE MEASURING PROPORTION OF REPLACED RULES; AMS = AVERAGE MODIFICATION SEVERITY FOR RETAINED RULES; N = NUMBER OF DATASETS.

Chunk	Drift Type	TCS	RIR	AMS	N
500	Abrupt	0.969 ± 0.141	0.723 ± 0.187	0.246 ± 0.167	21
500	Gradual	0.981 ± 0.093	0.709 ± 0.144	0.271 ± 0.139	14
500	Noisy	0.975 ± 0.120	0.718 ± 0.168	0.257 ± 0.155	8
500	Stationary	0.957 ± 0.156	0.672 ± 0.227	0.285 ± 0.223	9
500	Real	0.996 ± 0.017	0.715 ± 0.176	0.281 ± 0.176	8
1000	Abrupt	0.968 ± 0.150	0.707 ± 0.214	0.261 ± 0.187	21
1000	Gradual	0.975 ± 0.132	0.689 ± 0.177	0.286 ± 0.142	14
1000	Noisy	0.970 ± 0.145	0.705 ± 0.195	0.265 ± 0.175	8
1000	Stationary	0.942 ± 0.194	0.668 ± 0.240	0.274 ± 0.214	9
1000	Real	1.000 ± 0.000	0.742 ± 0.188	0.258 ± 0.188	8

TABLE XV
EGIS COMPLEXITY PENALTY EFFECT ON PERFORMANCE. THE PENALTY PARAMETER γ CONTROLS THE TRADE-OFF BETWEEN PREDICTIVE ACCURACY AND RULE SIMPLICITY IN THE FITNESS FUNCTION. G-MEAN VALUES ARE MEAN±STD ACROSS ALL 48 DATASETS. Δ SHOWS THE PERFORMANCE DIFFERENCE (POSITIVE INDICATES PENALTY IMPROVES PERFORMANCE). P-VALUES FROM PAIRED WILCOXON SIGNED-RANK TESTS INDICATE NO STATISTICALLY SIGNIFICANT DIFFERENCE, CONFIRMING THAT SIMPLER RULES CAN BE OBTAINED WITHOUT SACRIFICING ACCURACY.

Chunk	γ=0.0	γ=0.1	Δ	p-value
500	0.803±0.223	0.802±0.223	+0.001	0.108
1000	0.810±0.225	0.811±0.224	-0.001	0.929

TABLE XVI

PERFORMANCE COMPARISON (G-MEAN) - EXP-1000 CONFIGURATION. RESULTS FOR 42 BINARY DATASETS AND 10 MULTICLASS DATASETS ORGANIZED BY DRIFT TYPE. LARGER CHUNK SIZE (1000 INSTANCES) PROVIDES MORE TRAINING DATA PER EVOLUTIONARY CYCLE BUT POTENTIALLY DELAYS DRIFT RESPONSE. CDCMS AND ACDWM HAVE NO MULTICLASS SUPPORT (SHOWN AS –). SUMMARY STATISTICS SHOWN FOR BINARY-ONLY (N=42) AND ALL DATASETS (N=52). BEST PERFORMANCE PER ROW IN BOLD.

Dataset	EGIS	ARF	HAT	SRP	ROSE	eRulesD2S	CDCMS	ACDWM
Abrupt Drift (14)								
AGRAWAL_Abrupt_Chain_Long	0.809	0.790	0.676	**0.861**	0.830	0.526	0.622	0.742
AGRAWAL_Abrupt_Simple_Mild	**0.903**	0.799	0.646	0.803	0.900	0.537	0.607	0.828
AGRAWAL_Abrupt_Simple_Severe	0.910	0.847	0.690	0.857	**0.925**	0.537	0.684	0.834
HYPERPLANE_Abrupt_Simple	0.837	0.831	0.877	0.773	0.863	0.524	**0.896**	0.767
RANDOMTREE_Abrupt_Recurring	**0.730**	0.561	0.520	0.530	0.654	0.548	0.509	0.669
RANDOMTREE_Abrupt_Simple	**0.731**	0.548	0.492	0.501	0.659	0.550	0.524	0.670
RBF_Abrupt_Blip	0.905	0.914	0.836	**0.915**	0.896	0.527	0.837	0.830
RBF_Abrupt_Severe	0.846	**0.898**	0.812	0.896	0.870	0.508	0.777	0.776
SEA_Abrupt_Chain	0.959	0.972	0.930	0.931	**0.977**	0.618	0.933	0.879
SEA_Abrupt_Recurring	0.971	**0.977**	0.945	0.931	0.975	0.644	0.954	0.890
SEA_Abrupt_Simple	0.978	0.975	0.949	0.961	**0.979**	0.617	0.940	0.896
SINE_Abrupt_Simple	0.984	**0.987**	0.975	0.986	0.980	0.650	0.969	0.902
STAGGER_Abrupt_Chain	0.862	**0.995**	0.781	0.992	0.922	0.756	0.970	0.790
STAGGER_Abrupt_Recurring	0.909	**0.998**	0.786	0.991	**0.997**	0.763	0.978	0.833
Gradual Drift (9)								
HYPERPLANE_Gradual_Simple	0.842	0.837	0.879	0.771	0.863	0.540	**0.896**	0.772
RANDOMTREE_Gradual_Simple	**0.764**	0.513	0.527	0.487	0.669	0.537	0.542	0.701
RBF_Gradual_Moderate	0.886	0.896	0.819	**0.902**	0.889	0.514	0.823	0.813
RBF_Gradual_Severe	0.891	**0.905**	0.825	**0.905**	0.894	0.520	0.840	0.817
SEA_Gradual_Recurring	0.955	**0.959**	0.914	0.933	**0.960**	0.648	0.906	0.875
SEA_Gradual_Simple_Fast	**0.981**	0.978	0.947	0.932	0.980	0.600	0.945	0.899
SEA_Gradual_Simple_Slow	**0.983**	0.977	0.945	0.972	0.981	0.629	0.934	0.901
SINE_Gradual_Recurring	0.940	**0.952**	0.936	0.950	0.943	0.646	0.918	0.862
STAGGER_Gradual_Chain	0.794	**0.949**	0.722	0.943	0.920	0.780	0.911	0.728
Noisy Drift (8)								
AGRAWAL_Abrupt_Simple_Severe_Noise	0.911	0.809	0.660	0.749	**0.940**	0.545	0.630	0.835
HYPERPLANE_Gradual_Noise	0.825	0.822	0.855	0.763	0.848	0.537	**0.873**	0.756
RANDOMTREE_Gradual_Noise	**0.769**	0.532	0.552	0.464	0.690	0.536	0.535	0.705
RBF_Abrupt_Blip_Noise	**0.906**	0.902	0.824	0.904	0.886	0.519	0.829	0.831
RBF_Gradual_Severe_Noise	0.892	0.896	0.816	**0.900**	0.877	0.526	0.815	0.818
SEA_Abrupt_Chain_Noise	0.956	0.968	0.929	0.936	**0.971**	0.597	0.923	0.876
SINE_Abrupt_Recurring_Noise	0.893	0.980	0.953	**0.981**	0.973	0.627	0.927	0.818
STAGGER_Abrupt_Chain_Noise	0.862	**0.995**	0.764	0.987	0.922	0.777	0.925	0.790
Stationary Drift (7)								
AGRAWAL_Stationary	**1.000**	0.941	0.980	0.998	0.997	0.531	0.979	0.917
HYPERPLANE_Stationary	0.813	0.820	0.881	0.762	0.852	0.525	**0.908**	0.745
RANDOMTREE_Stationary	**0.698**	0.584	0.496	0.540	0.594	0.514	0.501	0.640
RBF_Stationary	0.835	0.927	0.900	**0.974**	0.963	0.283	–	–
SEA_Stationary	**0.985**	0.979	0.958	0.950	0.984	0.625	0.948	0.903
SINE_Stationary	0.986	0.987	0.971	**0.988**	0.982	0.635	0.975	0.904
STAGGER_Stationary	**1.000**	0.997	0.805	0.997	0.999	0.761	0.999	0.917
Real Drift (4)								
AssetNegotiation_F2	**0.987**	0.958	0.951	0.973	0.982	0.627	0.936	0.905
AssetNegotiation_F3	**0.999**	0.957	0.967	0.929	0.992	0.761	0.954	0.916
AssetNegotiation_F4	**0.957**	0.947	0.939	0.936	0.662	0.572	0.947	0.878
Electricity	0.784	0.891	0.856	0.854	**0.905**	0.638	0.830	0.719
Multiclass (10)								
CovType	0.207	0.415	0.272	**0.598**	0.250	0.401	–	–
IntelLabSensors	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	–	–	–
LED_Abrupt_Simple	**0.910**	0.887	0.117	0.603	0.405	0.212	–	–
LED_Gradual_Simple	**0.988**	0.802	0.070	0.675	0.347	0.222	–	–
LED_Stationary	**1.000**	0.854	0.000	0.642	0.274	0.206	–	–
PokerHand	0.000	0.000	0.000	0.000	0.000	**0.320**	–	–
Shuttle	0.357	0.059	0.000	0.059	0.000	**0.661**	–	–
WAVEFORM_Abrupt_Simple	0.726	0.792	0.762	0.821	**0.824**	0.384	–	–
WAVEFORM_Gradual_Simple	0.727	0.788	0.758	0.821	**0.822**	0.379	–	–
WAVEFORM_Stationary	0.727	0.811	0.760	0.822	**0.825**	0.416	–	–
Binary only (n=42):								
EGIS W/L/D	–	22/20/0	35/7/0	26/16/0	19/23/0	42/0/0	31/10/0	41/0/0
Mean	0.891	0.880	0.821	0.864	0.894	0.592	0.838	0.818
Std	0.085	0.138	0.145	0.152	0.110	0.096	0.153	0.078
Avg Rank	2.81	3.10	5.24	3.79	2.67	7.52	4.93	5.85
All datasets (n=52):								
EGIS W/L/D	–	26/24/2	39/11/2	30/20/2	23/27/2	48/3/0	31/10/0	41/0/0
Mean	0.828	0.815	0.716	0.795	0.794	0.550	0.661	0.645
Std	0.223	0.244	0.292	0.245	0.270	0.138	0.371	0.344
Avg Rank	2.90	3.04	5.12	3.56	2.78	6.94	5.46	6.19