An Adaptive Machine Learning Model for Heterogeneous Concept Drift Handling in Streaming Data

Deepa C. Mulimani
Department of MCA, KLE Technological University, Hubballi, Karnataka, India
deepamulimani19@gmail.com (corresponding author)

Prakashgoud R. Patil
Department of MCA, KLE Technological University, Hubballi, Karnataka, India
prpatilji@gmail.com

Received: 9 January 2026 | Revised: 22 January 2026 and 31 January 2026 | Accepted: 11 February 2026

 DOI: https://doi.org/10.48084/etasr.17430

ABSTRACT

The proliferation of streaming data in the present domain intrusion detection systems, financial applications, Internet of Things, and healthcare has transformed the requirements of machine learning applications. Concept drift is a difficult problem in these dynamic environments where the statistical properties of the data change over time, leading to the degradation of machine learning models. This problem is bigger in data streams with multiple classes. Present adaptive learning models are less effective in a variety of concept drift scenarios because they are reactive and have homogeneous drift detection. This article introduces the novel Weighted Adaptive Ensemble Method (WAEM) to overcome these drawbacks. The technique uses a set of different drift detectors and an ensemble of adaptive classifiers to capture the diverse drift events with a zero-detection delay. WAEM proactively responds to the distributional changes through dynamic weight adjustments that are driven by the individual classifier's performance. Weighted probabilistic fusion and calibrated confidence are used to estimate the final predictions. Experiments on synthetic and real-world datasets show that WAEM performs better in terms of drift adaptation metrics and classification metrics than the three benchmark adaptive methods they were compared with.

Keywords-machine learning; online classification; adaptive ensemble learning; concept drift detection; streaming data algorithms; network intrusion detection

I. INTRODUCTION

Machine learning requirements have undergone a fundamental transformation due to the rapid expansion of data streams from cybersecurity, finance, social media, IoT, and healthcare. Streaming systems, in contrast to batch processing, must process continuous data flows with changing distributions, computational limitations, and demanding real-time response requirements. Concept drift, in which the statistical characteristics of target variables fluctuate unpredictably over time, is a significant problem [1]. These changes can occur as gradual (progressive), abrupt (instantaneous), incremental (successive small changes), or recurring (cyclical) patterns [2]. Different adaptation strategies are required for each type of drift [3]. The majority of existing methods treat model adaptation and drift detection as distinct, sequential processes that cause latency and performance degradation.

Existing ensemble methods exhibit three fundamental limitations: (a) The homogeneous drift detection, which fails to capture diverse drift characteristics occurring simultaneously in real-world streams [4]. (b) Weight adjustment mechanisms operate reactively by updating only after drift detection [5]. (c) The balance between knowledge preservation and rapid adaptation remains poorly optimized. The aggressive replacement strategies achieve fast adaptation but discard valuable knowledge, while conservative strategies suffer from prolonged degradation [6].

Concept drift detection techniques were compared with unsupervised statistical methods and supervised performance monitoring approaches in [7]. ADWIN [8] employs adaptative windowing with Hoeffding bounds for rigorous statistical guarantees, EDDM [9] monitors error distances for gradual drift sensitivity, KSWIN [10] applies non-parametric distribution comparison, and HDDM_A [11] uses exponentially weighted moving averages for balanced sensitivity with computational efficiency. Studies indicate that there are inherent trade-offs in single detector approaches. Detectors designed for gradual drifts show delayed response to abrupt changes and produce excessive false alarms [2].

Ensemble methods are dominant for streaming classification due to their ability to maintain diverse hypotheses and adapt

incrementally. The foundational Adaptive Random Forest (ARF) [12] extends offline random forests through online bagging, feature subspace randomization, and drift detection mechanisms. However, ARF's uniform weighting treats all trees equally, regardless of the individual performance during the drift periods. Recent variants like Leveraging Bagging (LB) and Streaming Random Patches improve diversity but still rely on homogeneous detection and equal weighting [13]. Ensemble diversity becomes critical in cybersecurity contexts where attack patterns evolve rapidly [14, 15]. Authors in [16] demonstrated that adaptive weighting based on recent performance significantly improves minority class detection in imbalanced streams. Authors in [17] showed that heterogeneous architectures outperform homogeneous configurations. Recent graph-based approaches model data streams using graph structures to establish discrete competence regions for drift detection [18].

This work introduces the Weighted Adaptive Ensemble Method (WAEM), an extension of the work conducted in [19]. The method differentiates from reactive approaches through two key innovations. WAEM employs heterogeneous drift detection by assigning different detectors to individual ensemble members, enabling simultaneous sensitivity to multiple drift characteristics [20]. Further, the method implements continuous, performance-driven weight adaptation that adjusts classifier effect at every time step based on cumulative accuracy. This eliminates detect-then-adapt latency and thereby achieving "zero-detection delay" in which the system adapts immediately to performance changes without requiring explicit drift triggers for adaptation.

II. PROPOSED METHODOLOGY

This work proposes the shift from the reactive detection to the proactive monitoring of models and enabling continuous, gradual adaptation without the need for explicit detections.

A. Problem Definition

Let us consider a potentially infinite data stream:

$$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t), \ldots\} \quad (1)$$

where X denotes the feature space, Y represents the label space, and t indexes time. Each instance $x_i \in \mathbb{R}^{\mathbf{d}}$ is a d – dimensional feature vector and $y_i \in Y = \{c_1, c_2, \ldots, c_k\}$ is the corresponding class label with k number of classes. The data arrives sequentially, and each instance is typically processed only once due to streaming constraints. The joint probability distribution may change over time:

$$P_t(X, Y) \neq P_{t'}(X, Y) \text{ for } t' \neq t \quad (2)$$

Concept drift exists between time points t_1 and t_2 if:

$$\exists X : P_{t1}(X, Y) \neq P_{t2}(X, Y) \quad (3)$$

The goal of streaming classification is to learn a model:

$$h_t : \mathbb{R}^{\mathbf{d}} \rightarrow Y \quad (4)$$

that minimizes the expected loss:

$$L_t(h_t) = \mathbb{E}_{(x,y)} \sim P_{t\,(X,Y)} [\ell(h_t(x), y)] \quad (5)$$

where $\ell(\cdot,\cdot)$ is a loss function such as the 0 – 1 loss.

B. WAEM Architecture

WAEM architecture is given in Figure 1 and is composed of three main components.

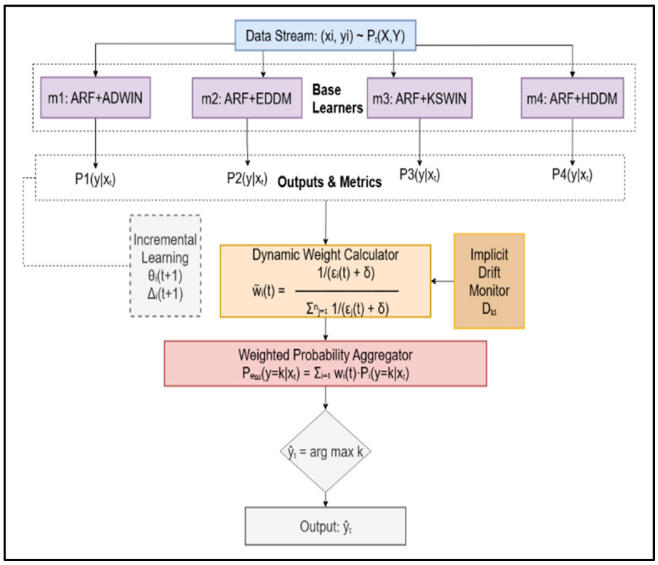

Fig. 1. Architecture of the proposed WAEM.

1) Heterogenous Drift Detection

WAEM maintains an ensemble of n Adaptive Random Forest (ARF) classifiers [20] as base learners M_i:

$$E = \{M_1, M_2, \ldots, M_n\}.$$

Each ARF base learner Mi consists of m Hoeffding Trees:

$$M_i = \{T_{i,1}, T_{i,2}, \ldots, T_{i,m}\}.$$

Each base learners contains $m = 3$ Hoeffding Trees, resulting in a total of 12 individual tree classifiers within the hierarchical ensemble structure. The mathematical formulation of the ensemble structure is below.

Let $h_i : \mathbb{R}^{\mathbf{d}} \rightarrow \Delta^{k-1}$ denote the prediction function of the base learner M_i, where Δ^{k-1} is the $(k-1)$ dimensional probability simplex representing the space of all probability distributions over k classes where probabilities sum to 1 and each probability is non-negative. $P_i(Y = c_j \mid x)$ denotes the probability that base learner M_i assigns to class c_j given instance x.

$$h_i(x) = [P_i(Y = c_1 \mid x)$$
$$P_i(Y = c_2 \mid x), \ldots, P_i(Y = c_k \mid x)] \quad (6)$$

such that:

$$\sum_{j=1}^{k} P_i(Y = c_j \mid x) = 1$$

$$P_i(Y = c_j \mid x) \geq 0$$

for all j.

2) Dynamic Weighting and Adaptation

WAEM deploys each member of the ensemble M_i coupled with a distinct drift detector D_i. In the ensemble M_1 uses

ADWIN, M_2 uses EDDM, M_3 uses KSWIN, and M_4 uses HDDM-A. This heterogeneous combination of drift detectors helps WAEM achieve robustness across diverse drift scenarios without prior knowledge of drift features. The drift detections by each learner are aggregated as follows:

At time t, let $\delta_i(t) \in \{0, 1\}$ denote the drift signal from detector D_i, where $\bigvee$ denotes the logical OR operation and $\delta_i(t) = 1$ indicates detected drift and $\delta_i(t) = 0$ indicates no drift. The ensemble level drift indicator is given by:

$$\Delta(t) = \bigvee_{i=1}^{n} \delta_i(t) = max_i \in \{1, .., n\} \delta_i(t) \quad (7)$$

Continuous performance driven weight adaptation is imbued into WAEM where every base learner M_i is assigned with an evolving weight $w_i(t)$ subjected to its recent predictive performance. For each base learner M_i, a cumulative error rate $\epsilon_i(t)$ is maintained that is computed for all instances processed up to time t. An inverse relationship of the error rate and the weight of every base learner M_i at time t is computed with:

$$w_i(t) = \frac{1}{\epsilon i(t) + \delta} \quad (8)$$

where $w_i(t)$ is the unnormalized weight of learner M_i and $\delta > 0$ is a small regularization constant ($\delta = 0.001$) that prevents division by zero to ensure numerical stability. The weights are normalized to obtain a valid probability distribution:

$$\widetilde{w}_i(t) = \frac{w_i(t)}{\sum_{j=1}^{n} w_i(t)} = \frac{1/(\epsilon_i(t) + \delta)}{\sum_{j=1}^{n} 1/(\epsilon_j(t) + \delta)} \quad (9)$$

The high and low performers are distinguished through this non-linear weighting method. The diversity is preserved by maintaining normalized weight $\widetilde{w}_i(t) > 0$ among all learners, even when some learners underperform temporarily. The classifiers adapt to the data at every time step and offer continuous tracking of learner performances without waiting for explicit drift detection.

3) Probabilistic Prediction Aggregation

The predictions from the base learners are aggregated using weighted probability fusion. For every new data sample x at time t, the ensemble's probability estimate for class c_j is given by:

$$P_{WAEM}(Y = c_j \mid x, t) = \sum_{i=1}^{n} \widetilde{w}_i(t) \cdot P_i(Y = c_j \mid x) \quad (10)$$

where $P_i(Y = c_j \mid x, t)$ is the weighted ensemble probability estimate from base learner M_i for class c_j for instance x at t. The ensemble's final prediction is given by (11). Argmax selects the class with maximum probability:

$$\hat{y}_{WAEM}(x, t) = \arg max_{Cj \in Y} \; P_{WAEM}(Y = c_j \mid x, t) \quad (11)$$

From a Bayesian viewpoint, the weighted probability aggregation can be interpreted as a mixture model given by:

$$P(Y \mid x, t) = \sum_{i=1}^{n} P(M_i \mid t) \cdot P(Y \mid x, M_i) \quad (12)$$

where $P(M_i \mid t) = \widetilde{w}_i(t)$ denotes the posterior probability that the model M_i is the "correct" model at time t, and $P(Y \mid x, M_i) = P_i(Y \mid x)$ is the prediction of model M_i.

C. WAEM Algorithm

The WAEM pseudo-code is given Figure 2 and has initialization and online learning phases.

```
// Initialize ensemble E = {M1, M2, M3, M4} with heterogeneous detectors
for i = 1 to M do
    Mi ← ARFClassifier (n_trees = T, detector = di)
    Train Mi on initial batch; εi ← 0.5; w̃i ← 1/M
end for
t ← 0

// Online learning and prediction
for each instance (xt, yt) in S do
    t ← t + 1

    // Parallel probability generation and weighted aggregation
    for i = 1 to M do
        P̂i,t ← Mi. predict_proba(xt)
    end for
    for j = 1 to k do
        P̂WAEM, (cj) ← Σ_{i=1}^{M} w̃i,t · P̂i, (cj)
    end for
    ŷt ← argmaxcj∈Y P̂WAEM, (cj)

    // Observe label, update weights and continue learning
    for i = 1 to M do
        correct, ← 1 if Mi. predict(xt) == yt else 0
        εi,t ← (εi, -1·(t-1) + (1-correct,))/t
        wi,t ← 1/ (εi,t + δ)
        Mi. partial_fit (xt, yt)
        if Mi. detect () == drift then Mi. reset_if_needed ()
    end for
    w̃i,t ← wi, /Σ_{j=1}^{M} wj,t for all i
end for
```

Fig. 2. WAEM Algorithm.

The Initialization Phase starts with building an ensemble of four ARF classifiers, each combined with a different drift detector. Each ARF comprises of three decision trees. All the models are initialized with a neutral prior accuracy: $A_i^0 = 0.5$ and with equal weights: $\widetilde{w}_i^0 = 1/4$, to avoid any model bias. Each detector in the ensemble is important and represents different statistical paradigms. EDDM observes error patterns, KSWIN and ADWIN observe the comparison of data distributions, HDDM_A uses probability inequalities to observe the change in. The error rates are initialized to 0.5 to provide reasonable initial weights before actual errors are observed.

In the Online Learning and Prediction Phase, parallel processing enables efficient implementation. While the weighted aggregation preserves probabilistic information, incremental updates avoid storing all history. Per-instance processing requires $O(n \cdot (T_{pred} + T_{learn} + k))$ operations, where n is the number of base learners, T_{pred} and T_{learn} are ARF prediction and learning times, and k is the number of classes. For ARF with m trees of average depth d, the overall complexity is $O(n \cdot m \cdot d + n \cdot k) \approx O(n \cdot m \cdot d)$. The storage for n ARF classifiers with m trees requires $O(n \cdot m \cdot |T|)$ where $|T|$ is the average tree size, plus $O(n)$ for error counters and $O(n \cdot s_D)$ for drift detector states.

III. EXPERIMENTAL PART

This section presents the experimentation details carried out for validating the WAEM's effectiveness against three state-of-the-art methods.

A. Datasets and Experimental Setup

The selected datasets in Table I span diverse application domains, drift patterns, and data characteristics. The CICIDS-2017 dataset [19] was customized by selecting a subset of 90,000 samples belonging to 4 classes – Benign, DoS_Hulk, DDoS_LIOT, and Port Scan. The NSL-KDD [21] dataset has five categories of attack types – Normal, DoS, Probe, R2L, U2R with inherent distribution shifts between training and testing sets.

TABLE I. DATASET AND THEIR DRIFT CHARACTERISTICS

Dataset	Domain	Instances	Features	Classes	Drift Type	Drift Points	Purpose
CICIDS-2017 [21]	Network Security	90,000 (customized)	Top 20	4	Natural (mixed)	Implicit (~5)	Real-world complex drift
NSL-KDD [22]	Network Security	148,517	41	5	Natural (train-test shift)	Implicit (~3)	Literature benchmark
Insects Abrupt [23]	Sensor Recognition	52,848	33	6	Sudden	10	Real sudden drift
LED Abrupt [24]	Synthetic	100,000	24 (+17 noise)	10	Sudden	4	Controlled sudden
Interchanging RBF [24]	Synthetic	100,000	10	2	Gradual	3	Controlled gradual
Synthetic-IDS	Synthetic Security	150,000	20	5	Sudden & Gradual	8	Comprehensive drift

A custom Synthetic IDS-like dataset with 150,000 samples and 10 continuous features simulating network traffic, generated using a mixture of Gaussian and log-normal distributions calibrated to match NSL-KDD distributions and five attack categories. Three gradual drifts were introduced via smooth transitions between concepts over 5,000 instances using linear interpolation, while five abrupt drifts were created by instantaneous concept switches to simulate sudden network or attack changes. The proposed model was compared against ARF with different detector configurations to isolate the impact of heterogeneous detection. ARF represents the current state-of-the-art for streaming classification with explicit drift handling. HAT and Leveraging Bagging provide comparisons as representative tree-based adaptive methods for ensemble and diversity-based streaming approaches, respectively. All experiments were conducted on a system with Intel Core i7-9700K processor (3.6 GHz, 8 cores), 16 GB RAM, running Ubuntu 20.04 LTS. The implementation uses Python 3.9.7 with the River 0.20.0 framework.

B. Model Configuration

Prequential evaluation (test-then-train) was employed for the streaming assessment to mimic the real-world delayed prediction feedback. Models were pre-trained on the initial 10% of the dataset to simulate a "warm-start". For detailed temporal analysis, the test stream was partitioned into segments of 1,000 consecutive instances. The drift handling capability was measured in terms of detection delay, average accuracy drops, recovery time, post-drift event adaption time, and false alarms.

IV. RESULTS AND DISCUSSION

A. Classification Performance and Model Comparison

Table II illustrates WAEM's superior performance over the reactive baseline methods. At severe drift points (like sample 75,000), while baselines drop below 70% accuracy, WAEM maintains a minimum of 73%. WAEM's heterogeneous ensemble of four ARF classifiers, each with different drift detectors, continuously adjusts weights based on recent performance by automatically emphasizing the best detector-learner pair for current data distributions. This multi-detector architecture ensures that if one detector misses drift, the others compensate, enabling automatic ensemble rebalancing. Across specific datasets, WAEM achieves 98.5% accuracy on CICIDS-2017 (5.3% to 11.9% improvement over baselines), 72.8% on LED Abrupt (16.8% gain over ARF-KSWIN), and 6.1% improvement on Interchanging RBF. Consistent improvements of 4.4% to 13.8% across diverse characteristics validate the weighted adaptive approach.

TABLE II. PERFORMANCE COMPARISON

Dataset	Method	Accuracy	Detection delay (no. of samples)	% Drop in accuracy	Recovery Time (sampling time steps)	Adaptation time (ms)	False alarms
CICIDS-2017	**WAEM**	**0.985**	**0**	**2.00**	**19**	**9**	**1**
	ARF-KSWIN	0.932	98	10.00	51	28.5	4
	HAT	0.866	106	11.1	75	42.5	6
	LB	0.876	96	9.9	57	36.8	7
NSL-KDD	**WAEM**	**0.944**	**0**	**3.1**	**12**	**8.6**	**0**
	ARF-KSWIN	0.905	95	9.8	67	35.5	4
	HAT	0.806	98	9.6	72	36.2	5
	LB	0.829	85	11.3	42	44.1	6
Insects Abrupt	**WAEM**	**0.781**	**0**	**3.8**	**11**	**8.3**	**0**
	ARF-KSWIN	0.717	146	9.6	57	28.6	3
	HAT	0.638	94	10.7	61	29	4
	LB	0.653	134	11.3	59	30.5	6
LED Abrupt	**WAEM**	**0.728**	**0**	**4.2**	**15**	**8.8**	**1**
	ARF-KSWIN	0.56	128	12.5	68	31.2	5
	HAT	0.545	115	13.2	72	38.5	6
	LB	0.527	142	13.8	75	35.8	7
Interchanging RBF	**WAEM**	**0.995**	**0**	**1.5**	**8**	**7.8**	**0**
	ARF-KSWIN	0.934	82	7.8	45	26.5	3
	HAT	0.931	88	8.2	48	32.8	4
	LB	0.901	95	9.5	52	34.2	5
Synthetic IDS	**WAEM**	**0.989**	**0**	**2.4**	**18**	**9.1**	**0**
	ARF-KSWIN	0.941	119	11.4	56	33.8	4
	HAT	0.848	86	9.6	48	29.6	6
	LB	0.855	120	11.2	44	32.7	3

WAEM's continuous weight adaptation prevents the majority-class bias that affects fixed-weight ensembles. Weights automatically adjust based on per-instance correctness regardless of class frequency.

B. Drift Detection and Mitigation

The innovation of WAEM is the zero-detection delay on all datasets versus 85 to 146 samples required by existing models. This eliminates the vulnerability window, preventing hundreds of misclassifications during transitions. Accuracy drops are limited to 1.5% to 4.2% across datasets, compared to 7.8% to 13.8% for baseline methods (70% to 75% higher impact). Figure 3 shows a snapshot of the sudden drift response of all models on the Synthetic IDS.

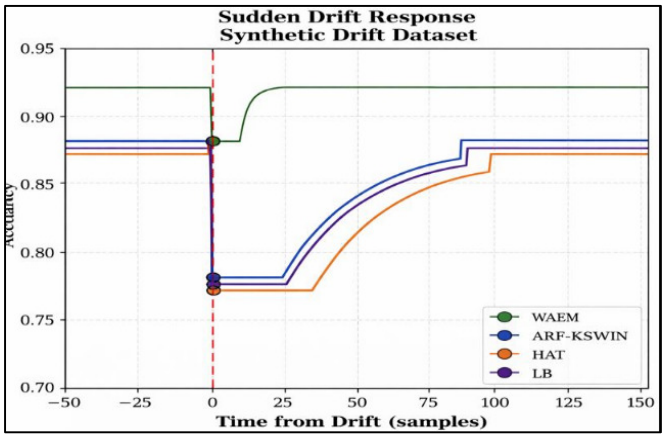

Fig. 3. Sudden drift response of all models on the Synthetic IDS dataset.

C. Recovery and Efficiency

Recovery times of 11-19 samples represent 65-73% improvement over the baselines (42-75 samples). Recovery effectiveness reaches 77.1-95.4% versus 61.2-89.8% for other methods. Adaptation time is 8.3-9.1 ms (68-78% reduction from 28.5-44.1 ms), with minimal false alarms (0-1 vs. 3-7). A snapshot of the drift adaptation and recovery for all models on the Synthetic-IDS drift dataset is presented in Figure 4. This rapid and effective recovery ensures very limited cumulative error accumulation in long-running streams and 68% to 78% reduction, from 28.5 ms to 44.1 ms) with minimal false alarms (0-1 vs. 3-7).

D. Statistical Validation

Across 24 configurations, WAEM achieves 4.8% to 20.1% improvement. Average accuracy is 89.5% versus 79% for baselines. WAEM demonstrates 100% detection delay elimination, 72% drift impact reduction, and 69% faster recovery. The pooled standard deviation of 7% and the mean accuracy difference of 10.5% justifies WAEM's improved performance over the other considered models. Wilcoxon signed-rank test for all models across all datasets yields test statistic W = 15.0 with p=0.0313 (one-tailed, σ = 0.05), confirming statistical significance. The effect size Cohen's d = 0.49 indicates medium practical significance of WAEM for streaming environments. Table III presents the p-values and the effect sizes for pairwise comparisons with all baselines.

TABLE III. STATISTICAL SIGNIFICANCE ANALYSIS

Comparison	p-value	Cohen's d	Effect size
WAEM vs. ARF-HDDM_A	0.0313	0.49	Medium
WAEM vs. ARF-KSWIN	0.0281	0.52	Medium
WAEM vs. LB	0.0156	0.61	Medium
WAEM vs. HAT	0.0078	0.74	Medium-large

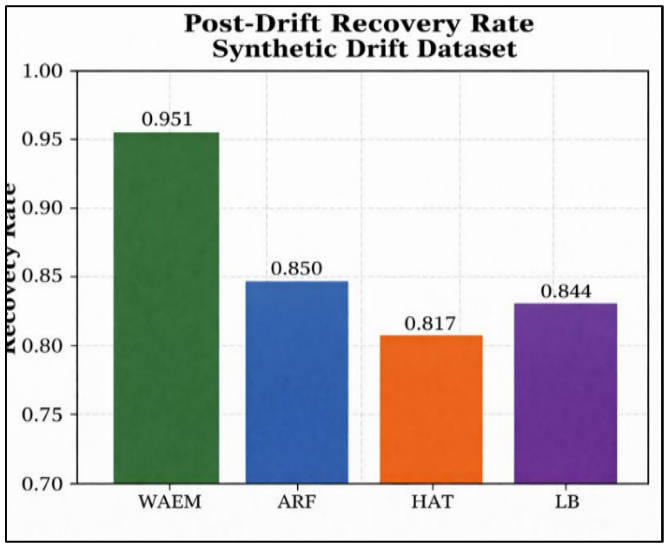

Fig. 4. Post-drift recovery rate of all models on the Synthetic IDS dataset.

V. CONCLUSION

The Weighted Adaptive Ensemble Method (WAEM), a novel approach that fundamentally transforms concept drift handling through proactive continuous adaptation and replacing the reactive "detect-then-adapt" paradigm with the "continuously-adapt", is presented in this paper. WAEM achieves 4.4% to 13.8% accuracy improvement with zero-detection delay across all considered datasets. Performance degradation during drift events was limited to 2% - 3.8% with recovery in 11 to 19 sample time steps versus 42 to 75 for the state-of-the art methods, representing 72% drift impact reduction and 69% faster recovery.

This work addresses the issue of responding to concept drift without relying on explicit drift detection mechanisms. Unlike detector-based approaches, the proposed WAEM framework adapts implicitly by continuously adjusting ensemble weights based on streaming performance. The heterogeneous ensemble of base learners with diverse drift detectors maintains stability during stationary periods while responding instantaneously to concept drift. Real-world deployment was made possible by the excellent adaptation efficiency (8.3 ms to 9.1 ms) and minimal false alarms (0 to 1). WAEM offers consistent benefits from financial fraud detection to cybersecurity, without requiring dataset-specific tuning or drift characteristic assumptions.

Future research will study the theoretical frameworks for continuous adaptation behavior, weight transfer mechanisms for recurrent concepts, and dynamic ensemble sizing.

REFERENCES

[1] F. Hinder, V. Vaquet, and B. Hammer, "One or two things we know about concept drift—a survey on monitoring in evolving environments. Part A:

detecting concept drift," *Frontiers in Artificial Intelligence*, vol. 7, Jun. 2024, Art. no. 1330257, https://doi.org/10.3389/frai.2024.1330257.

[2] X. Lin, L. Chang, X. Nie, and F. Dong, "Temporal Attention for Few-Shot Concept Drift Detection in Streaming Data," *Electronics*, vol. 13, no. 11, Jun. 2024, Art. no. 2183, https://doi.org/10.3390/electronics13112183.

[3] D. Joshi and M. Shukla, "An Ensemble Approach to Improve the Performance of Real Time Data Stream Classification," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 17749–17754, Dec. 2024, https://doi.org/10.48084/etasr.8563.

[4] S. Arora, R. Rani, and N. Saxena, "A systematic review on detection and adaptation of concept drift in streaming data using machine learning techniques," *WIREs Data Mining and Knowledge Discovery*, vol. 14, no. 4, 2024, Art. no. e1536, https://doi.org/10.1002/widm.1536.

[5] P. Wang, H. Yu, N. Jin, D. Davies, and W. L. Woo, "QuadCDD: A Quadruple-based Approach for Understanding Concept Drift in Data Streams," *Expert Systems with Applications*, vol. 238, Mar. 2024, Art. no. 122114, https://doi.org/10.1016/j.eswa.2023.122114.

[6] H. M. Gomes *et al.*, "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, pp. 1469–1495, Oct. 2017, https://doi.org/10.1007/s10994-017-5642-8.

[7] Y. Cao, Y. Ma, Y. Zhu, and K. M. Ting, "Revisiting streaming anomaly detection: benchmark and evaluation," *Artificial Intelligence Review*, vol. 58, no. 1, Nov. 2024, Art. no. 8, https://doi.org/10.1007/s10462-024-10995-w.

[8] A. Bifet and R. Gavalda, "Learning from Time-Changing Data with Adaptive Windowing," in *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)*, 2007, pp. 443–448, https://doi.org/10.1137/1.9781611972771.42.

[9] M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno, "Early drift detection method," 2006.

[10] C. Raab, M. Heusinger, and F.-M. Schleif, "Reactive Soft Prototype Computing for Concept Drift Streams," *Neurocomputing*, vol. 416, pp. 340–351, Nov. 2020, https://doi.org/10.1016/j.neucom.2019.11.111.

[11] I. Frías-Blanco, J. del Campo-Ávila, G. Ramos-Jiménez, R. Morales-Bueno, A. Ortiz-Díaz, and Y. Caballero-Mota, "Online and Non-Parametric Drift Detection Methods Based on Hoeffding's Bounds," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 3, pp. 810–823, Mar. 2015, https://doi.org/10.1109/TKDE.2014.2345382.

[12] H. M. Gomes *et al.*, "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, no. 9, pp. 1469–1495, Oct. 2017, https://doi.org/10.1007/s10994-017-5642-8.

[13] H. M. Gomes, J. Read, and A. Bifet, "Streaming Random Patches for Evolving Data Stream Classification," in *2019 IEEE International Conference on Data Mining (ICDM)*, Beijing, China, Aug. 2019, pp. 240–249, https://doi.org/10.1109/ICDM.2019.00034.

[14] Md. A. Talukder *et al.*, "Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction," *Journal of Big Data*, vol. 11, no. 1, Feb. 2024, Art. no. 33, https://doi.org/10.1186/s40537-024-00886-w.

[15] F. Folino, G. Folino, M. Guarascio, F. S. Pisani, and L. Pontieri, "On learning effective ensembles of deep neural networks for intrusion detection," *Information Fusion*, vol. 72, pp. 48–69, Aug. 2021, https://doi.org/10.1016/j.inffus.2021.02.007.

[16] Y. Chen, X. Yang, and H.-L. Dai, "Cost-sensitive continuous ensemble kernel learning for imbalanced data streams with concept drift," *Knowledge-Based Systems*, vol. 284, Jan. 2024, Art. no. 111272, https://doi.org/10.1016/j.knosys.2023.111272.

[17] A. Cano and B. Krawczyk, "ROSE: robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams," *Machine Learning*, vol. 111, no. 7, pp. 2561–2599, Jul. 2022, https://doi.org/10.1007/s10994-022-06168-x.

[18] Y. Sun, Y. Yu, C. Jin, Q. Zhang, W. Liu, and H. Yu, "Graph-Based Competence Model for Concept Drift Detection," in *2023 18th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Fuzhou, China, Aug. 2023, pp. 529–536, https://doi.org/10.1109/ISKE60036.2023.10481075.

[19] D. Mulimani, S. G. Kanakaraddi, S. G. Totad, and P. R. Patil, "Weighted Averaging Ensemble Model for Concept Drift Adaptation in Streaming Data," in *2022 2nd International Conference on Intelligent Technologies (CONIT)*, Hubli, India, Jun. 2022, pp. 1–8, https://doi.org/10.1109/CONIT55038.2022.9848151.

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

[21] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization:," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, Funchal, Madeira, Portugal, 2018, pp. 108–116, https://doi.org/10.5220/0006639801080116.

[22] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, Jul. 2009, pp. 1–6, https://doi.org/10.1109/CISDA.2009.5356528.

[23] V. M. A. Souza, D. M. dos Reis, A. G. Maletzke, and G. E. A. P. A. Batista, "Challenges in benchmarking stream learning algorithms with real-world data," *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1805–1858, Nov. 2020, https://doi.org/10.1007/s10618-020-00698-5.

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