Class-Specific Feature Weighting Query Strategy for Online Active Learning in Intrusion Detection

Daniel Mourão Martins[1(✉)], Ronaldo Ribeiro Goldschmidt[1], and Ronaldo Moreira Salles[1,2]

[1] Military Institute of Engineering, Rio de Janeiro, RJ, Brazil
{mourao.daniel,ronaldo.rgold,salles}@ime.eb.br
[2] CIICESI, ESTG, Polytechnic of Porto, Porto, Portugal
rmo@estg.ipp.pt

Abstract. Network Intrusion Detection Systems operate in dynamic environments characterized by continuously evolving network traffic and the emergence of previously unseen attack patterns. Traditional supervised learning approaches typically assume closed-set scenarios and rely on large labeled datasets, limiting their applicability in real-world streaming scenarios where labeling is costly and concept drift occurs. Active Learning offers an alternative by selectively querying informative samples for annotation. However, most existing AL query strategies rely primarily on prediction uncertainty and implicitly assume that feature importance remains globally consistent across classes. In intrusion detection tasks, this assumption is often unrealistic, since different attack types may depend on distinct subsets of features. This work proposes a Class-Specific Feature Weighting for Open-Set (CSFWOS) framework for Online Active Learning. The proposed method models feature relevance independently for each class and selects samples based on a weighted feature space. This strategy enhances the distinction between known classes and previously unseen attacks in the feature space, supporting more reliable detection across diverse attack patterns. Experimental results on benchmark datasets show improvements in incremental detection performance under low query ratios in non-stationary streaming environments compared with widely adopted active learning strategies. The findings demonstrate the potential of the approach to inspire further research.

Keywords: Active Learning · Query Strategy · Feature Weighting · Data Stream · Intrusion Detection

1 Introduction

The rapid evolution of networked systems and the increasing sophistication of cyber threats, particularly zero-day attacks, pose significant challenges to Intrusion Detection Systems (IDS). Network environments are inherently dynamic,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16768, pp. 349–367, 2027.
https://doi.org/10.1007/978-3-032-30488-9_22

where data distributions evolve over time due to new applications, configuration changes, and emerging cyber threats [3,12].

Traditional supervised learning approaches rely on the closed-set assumption, where all classes are known a priori. Although effective in static environments, these methods degrade significantly in streaming scenarios, where continuous adaptation is required and labeled data is scarce [4].

Active Learning (AL) mitigates labeling costs by selecting the most informative samples for annotation. In streaming settings, this paradigm becomes Online Active Learning (OAL), where decisions must be made sequentially under strict labeling budgets [5,19,34].

Obtaining reliable labels is a costly process that demands time and specialized knowledge, becoming a significant bottleneck. Manual annotation by cybersecurity experts is costly, time-consuming, and operationally impractical in high-throughput environments. Furthermore, due to the sensitive nature of data security in network traffic, which may contain confidential or personal information, automatic labeling mechanisms are generally undesirable or unsafe, as incorrect or unsupervised labeling may propagate errors, introduce bias, or compromise security decisions. Consequently, careful human validation remains necessary, but should be reduced.

This reliance on human intervention to label new attack samples introduces additional complexity, creating bottlenecks that delay response times and reduce system agility. Although automated annotation methods have advanced, they cannot guarantee reliable labels. Consequently, expert validation remains necessary, motivating the development of optimized strategies that reduce the need for human annotation while preserving labeling quality [20].

Current models face significant challenges in differentiating between benign variations in network traffic, evolved variants of known attacks, and genuinely novel threats. These challenges are particularly pronounced in highly dynamic network environments. Thus, concept drift is an inherent characteristic of network traffic, in which the statistical properties of the data evolve over time [31].

Not all network traffic features contribute equally to pattern identification. Identifying more significant attributes or even irrelevant attributes can increase the performance of classification algorithms. Feature Weighting (FW) addresses this issue by assigning relevance to each attribute, allowing the algorithm to focus on the most discriminating information [24].

For instance, TCP window size may be critical for detecting denial-of-service (DoS) attacks but irrelevant for port scans. In the CICIDS-2017 dataset, "Subflow Fwd Bytes" and "Total Length Fwd Packet" features are key for identifying Infiltration and Bot attacks, whereas "Bwd Packet Length Std" is strong indicator for DoS variants [14]. Thus, global feature weighting fails to capture these class-specific distinctions.

This work investigates the intersection of these research areas by leveraging recent advances that incorporate feature weighting into active learning query strategies and enhance clustering algorithms. Specifically, we propose Class-Specific Feature Weighting for Open Set (CSFWOS) framework that combines

feature-aware sample selection with improved cluster-based modeling. Our main contributions are summarized as follows:

- Class-specific feature weighting to capture heterogeneous feature relevance.
- A query strategy driven by feature-weight under labeling constraints.
- An OAL framework capable of handling concept drift in open-set scenarios.

The remainder of this paper is organized as follows. Section 2 presents the fundamental concepts of AL query strategies and FW, along with a review of related work. Section 3 describes the proposed methodology. Section 4 details the experimental setup and evaluation. Finally, Sect. 5 discusses the results and presents the conclusions of this study.

2 Background and Related Work

Active Learning has been widely studied to reduce the high cost of manual labeling in supervised tasks [13,32]. Rather than labeling entire datasets, AL iteratively selects the most informative instances through a query strategy. This approach is particularly suitable for cybersecurity, where expert analysis is required to validate alerts and identify new attack patterns [1,23]. By prioritizing informative samples, AL reduces annotation costs while maintaining competitive performance.

2.1 Active Learning in Data Streams

Traditional AL algorithms operate under batch learning assumptions, where the whole dataset is available during training. However, network traffic naturally arrives as a continuous data stream. In such environments, models must process incoming data sequentially while adapting to evolving distributions in real time under limited labeling budgets [5].

Data stream mining research highlights that evolving environments often suffer from concept drift [31]. In intrusion detection scenarios, these changes may occur due to novel attacks, evolving user behavior, or modifications in network infrastructure. Consequently, IDS models must continuously update their internal representations to remain effective.

2.2 Query Strategies in Active Learning

Hierarchical categorizations of AL query strategies indicate that they can be organized into several categories. Comprehensive taxonomies of these strategies are presented in [13,32].

However, despite variations in terminology and structural organization across the literature, most taxonomies converge toward a common set of conceptual groups. As illustrated in Fig. 1, query strategies encompass paradigms such as reinforcement learning, random sampling, and hybrid methods. Nevertheless, the core distinction lies in two primary categories: representativeness-based and informativeness-based.

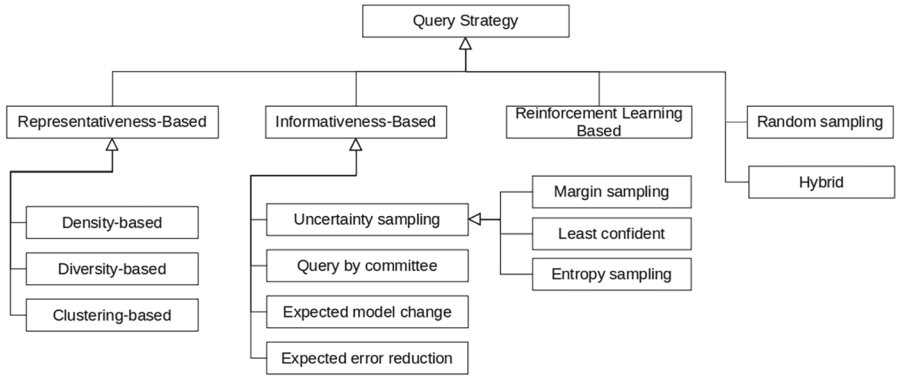

Fig. 1. A general taxonomy of query strategies for AL

- **Informativeness-Based:** these strategies select instances that maximize the expected information gain for the model or where the model exhibits low confidence in its predictions.
- **Representativeness-Based:** these strategies focus on the underlying structure of the data, selecting instances that are representative of the overall data distribution or of specific regions in the feature space.

In addition, hybrid strategies combine two or more query criteria, leveraging their complementary strengths while mitigating their individual limitations. Such approaches aim to balance exploration and exploitation [13,32], i.e. the exploration of new regions of the feature space with the exploitation of uncertainties at known boundaries. The evolution of online AL strategies shows a clear trend towards hybrid approaches.

Although these strategies have shown promising results in many domains, most existing approaches assume that feature contributions remain stable across classes. In network traffic data, however, the relevance of features may vary significantly between different types of attacks.

2.3 Feature Weighting in Active Learning

High-dimensional datasets often contain redundant or irrelevant features that may negatively affect the performance of machine learning algorithms. Feature weighting techniques have therefore been widely studied to estimate the relative importance of input variables and improve learning performance [9,10].

Global feature weighting approaches assign a single importance vector to the entire dataset, as in W-k-means [11] and IFWFCM [33]. Although effective in many scenarios, these approaches assume that feature relevance is homogeneous across the dataset.

More flexible approaches estimate feature relevance locally. Cluster-dependent weighting methods such as SSFCM-FWCW [9] and FWCW-FCM

[10] compute adaptive feature weights for each cluster, enabling subspace specialization and increased resilience to noise and outliers. Similarly, class-dependent weighting strategies have been proposed in high-dimensional classification settings, where feature importance varies significantly across classes [25].

In the context of Active Learning, feature-aware query strategies have recently gained attention. Methods such as AL-FEW [6] and SHAP-based acquisition strategies [30] incorporate feature importance information to guide instance selection. Although these approaches provide improved interpretability and structural reasoning, they typically rely on global feature importance measures, implicitly assuming that the same attributes are relevant for all classes.

2.4 Class-Specific Feature Weighting for Intrusion Detection

Different attack types often rely on distinct behavioral patterns and feature subsets. For instance, denial-of-service attacks may manifest through traffic volume anomalies, while infiltration attacks may be characterized by subtle protocol-level features. Consequently, global weighting schemes may obscure discriminative patterns.

Class-specific feature weighting addresses this limitation by estimating feature importance independently for each class. Instead of a single global vector, the model learns class-specific feature importance profiles.

This perspective is further supported by studies in text classification on Naïve Bayes classifiers, which show that class-specific feature weighting provides a more granular and realistic representation of feature relevance, since the importance of attributes naturally varies across classes [25].

Similar ideas have been explored in clustering and classification algorithms designed for high-dimensional data [9,25]. However, the integration of class-specific feature weighting into Active Learning query strategies remains limited.

Recent intrusion detection approaches have begun to incorporate domain knowledge into localized feature relevance. The DFAID framework [17], for example, incorporate feature-level information through a feature deviation score that identifies instances whose attribute values significantly deviate from the statistical distribution of the cluster where they reside. Similarly, A3PF [16] prioritizes samples by assigning higher importance to critical network attributes, but in a binary classification problem.

2.5 Comparative Analysis of Existing Strategies

The evolution from uncertainty-based to feature-aware approaches reveals a progressive shift toward structural reasoning. However, existing methods still exhibit important limitations, as summarized in Table 1.

Despite recent advances in OAL-based IDS, several limitations remain. First, most query strategies rely exclusively on prediction uncertainty and ignore structural information in feature space. Second, feature-aware approaches typically employ global feature importance, neglecting the fact that different attack classes

Table 1. Comparison of Active Learning strategies in intrusion detection

Method	Feature Scope	Query Criterion	Open-Set
Uncertainty Sampling	Global	Predictions based	Limited
Query by Committee	Global	Model disagreement	Limited
Density-Based	Global	Representativeness	Limited
Hybrid	Global	Mixed Criteria	Partial
AL-FEW [6]	Global	Feature Perturbation	No
SHAP-AL [30]	Global	Explanation based	No
Proposed	Class-Specific	Class Feature Weighting	Yes

may rely on distinct subsets of features. Finally, only a limited number of studies explicitly address open-set conditions where previously unseen attacks may appear.

These limitations motivate the development of query strategies capable of capturing class-specific variations in feature space while operating effectively in dynamic streaming environments, such as intrusion detection.

3 Proposed Methodology

This section presents the central contribution of this study: the Class-Specific Feature Weighting for Open-Set (CSFWOS) framework. The proposed methodology integrates local feature weighting with a query strategy specifically designed to capture structurally informative samples, supporting the learning of previously unseen attack patterns.

3.1 Streaming Protocol

A common evaluation protocol in streaming environments is prequential (test-then-train) evaluation, which evaluates each incoming instance before incorporating it into the training process [7,19]. This protocol provides an unbiased estimate of online predictive performance and enables continuous monitoring of performance degradation caused by distributional changes. In this protocol, each incoming batch is first evaluated using the current model to estimate instantaneous generalization performance and, if selected by the query strategy, is then incorporated into the training set.

This process ensures that performance metrics reflect the current data distribution, so degradation in prequential metrics such as Micro-F1 can act as an implicit signal of concept drift, enabling continuous monitoring and incremental model adaptation.

Let $\mathcal{B}_t$ denote the batch at time step t. The stream is defined as:

$$\mathcal{D} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_T\} \tag{1}$$

In summary, for each incoming batch $\mathcal{B}_t$: (i) the model predicts labels (test phase), (ii) the query strategy selects informative samples, (iii) the oracle provides labels only for queried instances, and (iv) the model is incrementally updated (train phase).

3.2 Feature Importance Estimation

The framework is agnostic to the specific feature weighting method and supports multiple implementations, for example:

- **Impurity-based importance:** Feature importance metrics estimated from impurity-based splitting criteria in tree-based models (e.g., Gini impurity, entropy, and log-loss). Let $\mathrm{I}_j^{(c)}$ be the importance for feature j and class c from a one-vs-rest tree ensemble:

$$w_j^{(c)} = \mathrm{I}_j^{(c)} \tag{2}$$

- **SHAP-based importance:** Feature contributions computed using SHAP values. Let $\phi_j^{(c)}(\mathbf{x})$ be the SHAP value for feature j in class c and $\mathbb{E}[\cdot]$ the expectation operator representing the average value computed over the set of observed samples:

$$w_j^{(c)} = \mathbb{E}\big[|\phi_j^{(c)}(\mathbf{x})|\big] \tag{3}$$

- **ReliefF weighting:** Feature relevance estimated based on nearest-neighbor differences, based on relief algorithms:

$$w_j^{(c)} = \mathrm{ReliefF}_j^{(c)} \tag{4}$$

This flexibility allows adapting the framework to different models and computational constraints.

3.3 Class-Specific Feature Weighting

To capture heterogeneous feature relevance, we maintain a class-specific feature weighting representation [8]:

$$\mathbf{w}_t^{(c)} = \{w_{t,1}^{(c)}, w_{t,2}^{(c)}, \ldots, w_{t,d}^{(c)}\}, \quad c \in \mathcal{Y}_t \tag{5}$$

where $w_{t,j}^{(c)}$ represents the importance of feature j for class c at time t.

Unlike global feature importance, this formulation allows modeling class-specific patterns, which is particularly relevant in intrusion detection where different attack types rely on distinct subsets of features.

Let $\mathbf{x} \in \mathbb{R}^d$ denote an sample in $\mathcal{B}_t$ represented in a d-dimensional feature space, and let $\mathcal{C} = \{c_1, \ldots, c_n\}$ be the set of known classes. A classifier provides prior probabilities $P(c|\mathbf{x})$ for each class $c \in \mathcal{C}$.

A weighted representation of $\mathbf{x}$ is computed using:

$$\tilde{\mathbf{x}} = \mathbf{x} \odot \mathbf{w}(\mathbf{x})$$

where $\odot$ denotes the Hadamard (element-wise) product and $\mathbf{w}(\mathbf{x})$ can be defined according to one of the following strategies.

- **Probability-weighted feature space.** The feature weights are calculated as the expected class-specific weight vectors under the previously estimated distribution of all classes:

$$\mathbf{w}(\mathbf{x}) = \sum_{c \in \mathcal{C}} P(c|\mathbf{x}) \mathbf{w}^{(c)}$$

- **Most probable class weighting.** Weights correspond to the class with maximum estimated probability:

$$c^* = \arg\max_{c \in \mathcal{C}} P(c|\mathbf{x}), \quad \mathbf{w}(\mathbf{x}) = \mathbf{w}^{(c^*)}$$

- **Top-k class averaged weighting.** To improve robustness, weights can be computed as the mean weight vector of the k most probable classes:

$$\mathbf{w}(\mathbf{x}) = \frac{1}{k} \sum_{c \in \mathcal{C}_k(\mathbf{x})} \mathbf{w}^{(c)}$$

  where $\mathcal{C}_k(\mathbf{x})$ denotes the set of k classes with highest probability.

3.4 Query Strategy

The CSFWOS adopts a hybrid query strategy that combines representativeness and uncertainty under a query budget. Let $\mathbf{x} \in \mathbb{R}^d$ denote an input sample and $\tilde{\mathbf{x}}$ its representation in the class-specific weighted feature space. Suspicious samples are defined as those for which the classifier exhibits predictive confidence below a threshold, i.e.

$$\mathcal{S} = \{\mathbf{x} \mid \max_{c \in \mathcal{C}} P(c|\mathbf{x}) < \tau_t\}.$$

First, the suspicious samples are mapped into the weighted feature space and clustered in order to partition them into regions with similar feature relevance. From each cluster k, a representative sample $\mathbf{r}_k$ is selected.

Let QR denote the predefined query ratio. The number of samples that can be submitted to the oracle for labeling is therefore defined as:

$$|\mathcal{Q}| = QR \times |\mathcal{B}_t| \tag{6}$$

The query set $\mathcal{Q}$ is constructed in two stages. Initially, the representative samples from each cluster are included:

$$\mathcal{Q}_1 = \{\mathbf{r}_k \mid k = 1, \ldots, K\}.$$

If $|\mathcal{Q}_1| < |\mathcal{Q}|$, the remaining query positions are filled using an uncertainty-based ranking. Let $U(\mathbf{x})$ denote the uncertainty score of the classifier. The remaining samples are selected according to

$$\mathcal{Q}_2 = \mathrm{Top}_{|\mathcal{Q}|-|\mathcal{Q}_1|}\left(U(\mathbf{x})\right), \quad \mathbf{x} \in \mathcal{S} \setminus \mathcal{Q}_1.$$

Finally, the query decision function is defined as

$$Q(\mathbf{x}_t) = \begin{cases} 1, & \mathbf{x}_t \in \mathcal{Q}_1 \cup \mathcal{Q}_2 \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

This strategy ensures that the selected samples maximize structural representativeness in the weighted feature space while prioritizing highly uncertain instances when completing the query budget.

Algorithm 1 summarizes the proposed framework. Initially, predicted class probabilities are estimated and suspicious samples are identified based on model confidence (lines 1–3). For each incoming batch, a class-specific feature importance vector is computed, and a weighted representation of each sample is derived accordingly (lines 5–7). Subsequently, the query budget is determined and clustering is performed in the weighted feature space (lines 8–9), from which representative samples are selected (line 10). The algorithm then, when necessary, augments the query set with additional instances selected based on uncertainty scores, in accordance with the predefined query ratio (lines 11–18). Finally, the selected samples are queried for labeling, the model is updated, and the process proceeds to the next batch (lines 19–22).

4 Experimental Evaluation

4.1 Experimental Setup

To evaluate the effectiveness of the proposed query strategy, we conduct experiments in a streaming scenario under non-stationary and open-set conditions.

Datasets. The evaluation is performed using widely adopted intrusion detection benchmarks datasets: NF-UNSW-NB15-V2, CIC-IDS2017 [17,22].

These datasets are widely used in the literature due to their diversity of attack types, class imbalance, and realistic traffic characteristics [21,29]. This evaluation includes the NetFlow-based version of the UNSW-NB15 dataset, which consist of network flow records extracted from raw packet captures and transformed into standardized NetFlow features [27].

In the NF-UNSW-NB15-V2 dataset, the "MIN_TTL" and "MAX_TTL" features were excluded from the analysis because they exhibited unrealistically high predictive performance [26].

To simulate real-world streaming conditions, the data stream is constructed such that the initial training set includes only benign traffic and a single known attack class. The remaining attack classes are then introduced progressively over

Algorithm 1. CSFWOS Query Strategy

Require: Stream of batches $\mathcal{D} = \{\mathcal{B}_1, \ldots, \mathcal{B}_T\}$, query ratio QR , classifier model f

for each batch $\mathcal{B}_t$ **do**
  Predict class probabilities $P(c|\mathbf{x})$ for all $\mathbf{x} \in \mathcal{B}_t$
  Identify suspicious samples $\mathcal{S} = \{\mathbf{x} \in \mathcal{B}_t \mid \max_c P(c|\mathbf{x}) < \tau\}$
  for each $\mathbf{x} \in \mathcal{S}$ **do**
    Compute class-specific weighted vector $\mathbf{w}(\mathbf{x})$
    Compute weighted representations $\tilde{\mathbf{x}} = \mathbf{x} \odot \mathbf{w}(\mathbf{x})$
  end for
  Compute query budget cardinality $|\mathcal{Q}| = \lfloor QR \times |\mathcal{B}_t| \rfloor$
  Cluster $\tilde{\mathbf{x}}$ into k clusters, where k is proportional to $|\mathcal{Q}|$
  Select representative sample $\mathbf{r}_k$ from each cluster
  Initialize query set $\mathcal{Q}_1 = \{\mathbf{r}_1, \ldots, \mathbf{r}_K\}$
  if $|\mathcal{Q}_1| < |\mathcal{Q}|$ **then**
    Compute uncertainty score $U(\mathbf{x})$ for $\mathbf{x} \in \mathcal{S} \setminus \mathcal{Q}_1$
    Select remaining samples $\mathcal{Q}_2 = \mathrm{Top}_{|\mathcal{Q}|-|\mathcal{Q}_1|}(U(\mathbf{x}))$
    $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2$
  else
    $\mathcal{Q} = \mathcal{Q}_1$
  end if
  Query oracle for labels of $\mathbf{x} \in \mathcal{Q}$
  Update classifier f using new labels
  Update class-specific feature weighting $w_j^{(c)}$ using new labels
end for

time. Specifically, each dataset is partitioned into larger segments, with each segment incorporating a new attack class not present in the previous ones. Once introduced, samples from a given attack class continue to appear in subsequent segments until the available instances are exhausted or the stream terminates. These macro-level segments are subsequently concatenated and repartitioned, while preserving their order, into smaller batches according to the predefined batch size. This procedure ensures the gradual emergence of novel attacks while maintaining a consistent batch structure for streaming evaluation.

Initial Configuration. The proposed framework supports multiple design alternatives and parameter configurations. In this study, a single empirically representative configuration was adopted, based on commonly used parameter settings reported in the literature.

This choice balances statistical stability, computational efficiency, and experimental reproducibility while enabling a clear evaluation of the proposed query strategy under realistic labeling constraints. Consequently, the evaluation focuses on assessing the effectiveness of the proposed framework. The selected parameters are summarized in Table 2.

Labeling efficiency is evaluated using query ratios $1\%, 5\%, 10\%, 20\%$, enabling analysis under different levels of human supervision. To facilitate consistent comparison across experiments, the confidence threshold is kept fixed in the experi-

Table 2. Configuration parameters

Parameter	Value
Initial seed size	10,000 samples
Seed composition	benign + 1 known attack
Batch size ($\|B_t\|$)	1,000 samples
Number of batches	58(NF-UNSW-NB15-V2), 34(CIC-IDS2017)
Confidence threshold (τ)	0.95
Query ratio (QR)	1%, 5%, 10%, 20%
Clustering model	Mini-batch K-means
Cluster scaling factor k	0.5
Cluster representative sample $\mathbf{r}_k$	Nearest point to centroid
Uncertainty criteria ($U(\mathbf{x})$)	Margin Sampling
Classifier model f	Random Forest
Feature importance estimation	Impurity-based importance: Gini impurity
Class-specific feature weighting	Probability-weighted feature space
Evaluation protocol	Prequential (test-then-train) evaluation

ments. The classifier and the feature-importance estimation model are independent to avoid bias.

To evaluate zero-day detection capability, an open-set simulation protocol is adopted in which only one attack class is initially known, while the remaining attack classes are introduced in subsequent batches [20]. This setup reproduces real environments where new attacks emerge after deployment.

4.2 Baselines

The proposed method is compared against:

- Random Sampling (RS)
- Uncertainty Sampling: Margin Sampling (US)
- Hybrid Approach: Margin Sampling + Clustering-based (Hybrid)
- No Active Learning: all samples labeled (Query All)

These methods represent standard baselines in active learning and open-set recognition [2,28].

4.3 Results and Analysis

Tables 3 and 4 present a comparison between the baseline methods and the proposed CSFWOS approach across the evaluated datasets. The reported results correspond to the average performance over 10 independent runs for each configuration, using different random seeds to ensure robustness. CSFWOS adopts the

Table 3. Performance on **NF-UNSW-NB15-V2** under different query ratios.

Method	QR	Macro F1	Macro Recall	Balanced Acc.	AULC
RS	1%	0.562 ± 0.005	0.589 ± 0.004	0.660 ± 0.004	0.733 ± 0.003
	5%	0.660 ± 0.003	0.678 ± 0.002	0.767 ± 0.002	0.818 ± 0.001
	10%	0.688 ± 0.002	0.704 ± 0.002	0.796 ± 0.001	0.841 ± 0.001
	20%	0.707 ± 0.001	0.723 ± 0.001	0.817 ± 0.001	0.857 ± 0.001
US	1%	0.542 ± 0.011	0.575 ± 0.008	0.638 ± 0.006	0.713 ± 0.005
	5%	0.646 ± 0.004	0.662 ± 0.004	0.747 ± 0.003	0.802 ± 0.003
	10%	0.691 ± 0.003	0.706 ± 0.003	0.798 ± 0.003	0.842 ± 0.002
	20%	0.715 ± 0.002	0.730 ± 0.002	0.822 ± 0.000	0.861 ± 0.000
Hybrid	1%	0.540 ± 0.011	0.570 ± 0.009	0.639 ± 0.008	0.715 ± 0.007
	5%	0.660 ± 0.002	0.675 ± 0.002	0.764 ± 0.003	0.816 ± 0.002
	10%	0.692 ± 0.002	0.708 ± 0.002	0.799 ± 0.002	0.843 ± 0.001
	20%	0.713 ± 0.002	0.728 ± 0.002	0.822 ± 0.001	0.860 ± 0.001
CSFWOS	1%	0.532 ± 0.014	0.563 ± 0.012	0.632 ± 0.012	0.710 ± 0.011
	5%	0.660 ± 0.003	0.677 ± 0.003	0.764 ± 0.003	0.816 ± 0.002
	10%	**0.695 ± 0.002**	**0.711 ± 0.002**	**0.803 ± 0.001**	**0.846 ± 0.001**
	20%	0.715 ± 0.002	0.730 ± 0.002	0.822 ± 0.001	0.861 ± 0.001
Query All	100%	0.723 ± 0.002	0.736 ± 0.002	0.831 ± 0.000	0.867 ± 0.000

Table 4. Performance on **CIC-IDS2017** under different query ratios.

Method	QR	Macro F1	Macro Recall	Balanced Acc.	AULC
RS	1%	0.713 ± 0.029	0.753 ± 0.018	0.812 ± 0.019	0.808 ± 0.023
	5%	0.771 ± 0.012	0.791 ± 0.009	0.857 ± 0.010	0.862 ± 0.011
	10%	0.775 ± 0.010	0.793 ± 0.008	0.862 ± 0.007	0.868 ± 0.011
	20%	0.772 ± 0.012	0.789 ± 0.009	0.862 ± 0.007	0.868 ± 0.011
US	1%	0.712 ± 0.029	0.748 ± 0.017	0.809 ± 0.020	0.804 ± 0.027
	5%	0.745 ± 0.029	0.772 ± 0.019	0.836 ± 0.018	0.838 ± 0.023
	10%	0.748 ± 0.020	0.775 ± 0.015	0.841 ± 0.010	0.841 ± 0.014
	20%	0.768 ± 0.014	0.787 ± 0.011	0.859 ± 0.010	0.865 ± 0.013
Hybrid	1%	0.721 ± 0.022	0.758 ± 0.014	0.820 ± 0.015	0.816 ± 0.022
	5%	0.771 ± 0.023	0.790 ± 0.017	0.860 ± 0.015	0.866 ± 0.021
	10%	0.779 ± 0.021	0.797 ± 0.016	0.865 ± 0.016	0.872 ± 0.021
	20%	0.781 ± 0.008	0.796 ± 0.007	0.866 ± 0.006	0.875 ± 0.008
CSFWOS	1%	**0.731 ± 0.027**	**0.763 ± 0.017**	**0.824 ± 0.017**	**0.823 ± 0.023**
	5%	0.771 ± 0.024	0.790 ± 0.019	0.857 ± 0.016	0.865 ± 0.020
	10%	**0.783 ± 0.020**	0.798 ± 0.015	**0.868 ± 0.012**	**0.877 ± 0.016**
	20%	0.782 ± 0.015	0.797 ± 0.012	0.867 ± 0.008	0.876 ± 0.011
Query All	100%	0.799 ± 0.005	0.809 ± 0.005	0.920 ± 0.000	0.939 ± 0.000

same configuration as the Hybrid method, differing only in the incorporation of class-specific feature weighting. Therefore, any observed performance differences between these two methods can be directly attributed to the effect of introducing class-weighted feature importance.

The results summarized in the tables show that increasing the query ratio consistently improves performance. However, within the same dataset and query ratio, the performance of all strategies is similar, with minor variations observed.

Performance differences across datasets can be attributed to variations in their underlying characteristics, including feature sets and their distributions, class overlap, attack types, and the quality of data generation and extraction. Consequently, the evaluation metrics exhibit distinct value ranges across datasets. These variations highlight the challenges of evaluating query strategies across multiple datasets, even within the same application domain.

On the NF-UNSW-NB15-V2 dataset (Table 3), the proposed CSFWOS approach achieves the best performance at QR = 10%. Although the gains over competing methods are moderate, this result indicates that the proposed class-specific feature weighting mechanism can effectively identify informative samples. The RS method was more effective at QR = 1%, while in the other experiments there were similar results between some methods.

In contrast, on the CIC-IDS2017 dataset (Table 4), CSFWOS demonstrates more consistent gains across different query ratios. Notably, it achieves the best results at QR = 1% and QR = 10%, outperforming the baselines in most metrics, including AULC. This behavior may be attributed to clearer class-specific patterns or more stable feature distributions in this dataset.

From a comparative perspective, the Hybrid method generally performs competitively. However, CSFWOS consistently achieves slight improvements in several scenarios, despite their nearly identical configurations.

Finally, it is worth noting that the gap between all active learning strategies and the fully supervised setting (Query All) remains significant, especially in the CIC-IDS2017 dataset. This reinforces the inherent trade-off between labeling cost and performance.

Per-Class Analysis. To better understand the behavior of the proposed strategy across different attack categories, we analyze the evolution of the F1-score for each class during the streaming process. Figure 2 illustrates the temporal evolution of per-class performance for selected classes from one of the experimental scenarios using the NF-UNSW-NB15-V2 dataset with a query ratio of 10%.

Several observations can be derived from this analysis. Attack classes that appear later in the stream, such as *Generic*, initially present low detection performance but gradually improve as informative samples are selected by the query strategy and incorporated into the training process. This behavior reflects the ability of the framework to adapt to emerging patterns in the data distribution.

Second, minority classes such as *Backdoor* show more variability in their performance due to limited sample availability. Finally, classes such as *Fuzzers*, *Reconnaissance*, and *Shellcode* reach high performance relatively quickly, indi-

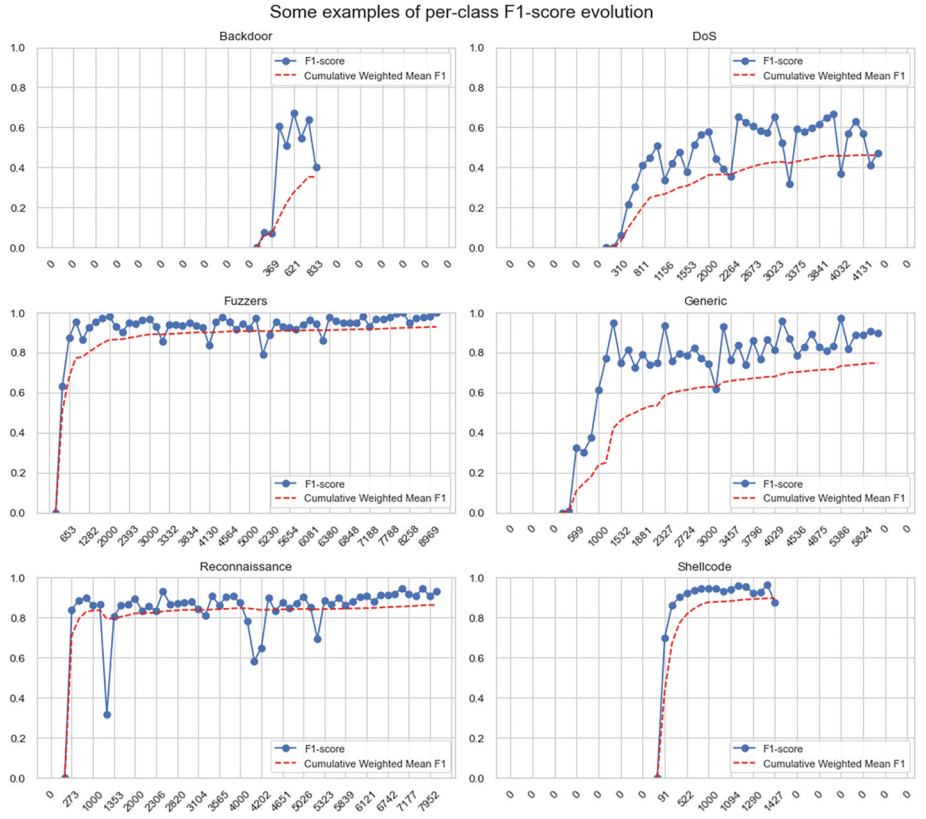

Fig. 2. Per-class F1-score evolution across the streaming batches. The blue curve represents the instantaneous F1-score for each class, while the red dashed curve indicates the cumulative weighted mean F1-score over time. (Color figure online)

cating that the proposed strategy effectively captures the relevant structural patterns of these attack categories in the feature space.

Overall, the curves show heterogeneous behavior across attack types: some classes exhibit rapid performance gains, while others improve more gradually, and in certain cases high F1-scores are not achieved.

Similar patterns were observed across all experimental configurations.

AULC Analysis. To evaluate the labeling efficiency, we analyze the evolution of classification performance throughout the streaming process across all experiments. As an illustrative example, Fig. 3 presents the Micro-F1 score as a function of the cumulative number of processed samples in one of these scenarios.

The dotted vertical lines in Fig. 3 indicate the points at which new attack classes are introduced into the stream. At these points, a noticeable drop in the curve is observed, reflecting the temporary degradation in performance caused

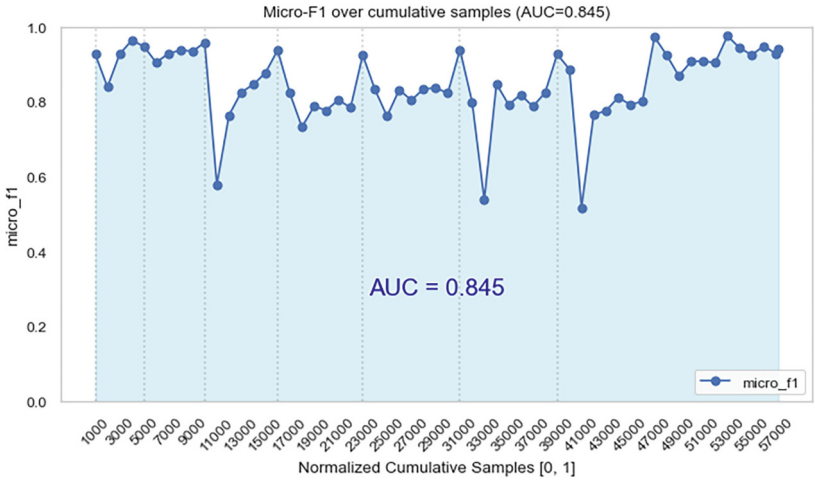

Fig. 3. Micro-F1 score over the normalized cumulative number of samples. The shaded region represents the Area Under the Learning Curve (AULC).

by the appearance of previously unseen patterns. However, the recovery rate is relatively fast as related samples are labeled. This behavior highlights the ability of the query strategy to quickly identify, label, and incorporate informative samples related to new classes into the training set.

Overall, the AULC values presented in the Tables 3 and 4 indicates that the proposed method not only attains higher final performance but also exhibits greater learning efficiency throughout the stream.

5 Discussion and Conclusion

The results of this study reinforce that no universal Active Learning strategy is suitable for all deployment scenarios. The effectiveness of a query strategy depends on multiple factors, including computational constraints, labeling resources, and the characteristics of the specific data domain. In the intrusion detection domain, these constraints are further exacerbated by non-stationarity, class imbalance, and the presence of previously unseen attack types.

Features that exhibit low importance for known classes may become highly informative when detecting novel classes, particularly when their values lie outside the support of known distributions. This highlights a fundamental limitation of classical feature importance methods: in open-set scenarios, the most relevant features are not necessarily those that maximize discrimination among known classes, but those that signal deviations from established patterns.

Recent research explores complementary strategies to address these challenges. For instance, pseudo-labeling expands the labeled set without additional oracle cost [16], while adaptive query budgets driven by drift detection methods, such as ADWIN [15], concentrate labeling efforts during distributional changes.

Our proposal is modular and model-agnostic, allowing the Active Learning framework to incorporate different learning models [6,18]. The results show improved sample selection and greater robustness to evolving data distributions, effectively identifying both known classes and novel or rare events.

Regarding computational cost, the additional steps are limited to class-specific feature weighting and the construction of weighted feature representations. The resulting computational overhead is related to the choice of feature importance estimation methods and feature weighting strategies.

For instance, in our experiments, combining impurity-based importance estimation with a probability-weighted feature space yields a computational complexity on the order of $\mathcal{O}(ndC)$, where n is the number of samples, d is the number of features, and C is the number of classes. These operations therefore scale linearly with these variables and are applied only to the current batch, resulting in a bounded per-iteration cost.

Since the framework reuses the clustering and uncertainty components of hybrid strategies, the overall computational complexity remains in the same asymptotic class. Moreover, these operations can be parallelized or computed on a subset of samples, further reducing practical overhead. Consequently, under the adopted configuration, the additional overhead remains controlled, achieving a favorable balance between improved sample selection and efficient resource utilization.

5.1 Future Work

Several directions remain open for future research. First, integrating full open-set and incremental capabilities represents a promising direction for future work, along with exploring optimization strategies for selecting alternative initial configurations and other datasets.

Second, extending the framework algorithm to explore hybrid strategies by evaluating additional configuration combinations that integrate structural feature analysis with representation-based and information-based approaches may further enhance its ability to detect and adapt to emerging threats.

Thirdly, improving robustness to adversarial attacks is essential, particularly in security-sensitive applications where attackers may attempt to manipulate feature distributions.

Finally, to maximize the impact of each query, it is necessary to use label propagation. Our research aim to leverage label propagation in the future.

Acknowledgments. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 001. It was also supported in part by the Brazilian Funding Authority for Studies and Projects (Finep) [Cyber Semantics Project - Contract No. 0.1.22.0335.00/Ref. Finep No. 0172/22] and the Systems Development Center of the Brazilian Army.

References

1. Abdulganiyu, O.H., Ait Tchakoucht, T., Saheed, Y.K.: A systematic literature review for network intrusion detection system (IDS). Int. J. Inf. Secur. **22**(5), 1125–1162 (2023). https://doi.org/10.1007/s10207-023-00682-2
2. Aggarwal, C.C.: Data Classification: Algorithms and Applications. CRC Press (2014). https://doi.org/10.1201/b17320
3. Ahmad, R., Alsmadi, I., Alhamdani, W., Tawalbeh, L.: Zero-day attack detection: a systematic literature review. Artif. Intell. Rev. **56**(10), 10733–10811 (2023). https://doi.org/10.1007/s10462-023-10437-z
4. Alotaibi, F., Maffeis, S.: Rasd: semantic shift detection and adaptation for network intrusion detection. In: Pitropakis, N., Katsikas, S., Furnell, S., Markantonakis, K. (eds.) SEC 2024. IFIP Advances in Information and Communication Technology, vol. 710, pp. 16–30. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-65175-5_2
5. Cacciarelli, D., Kulahci, M.: Active learning for data streams: a survey. Mach. Learn. **113**(1), 185–239 (2023). https://doi.org/10.1007/s10994-023-06454-2
6. Elokri, C., Ouaderhman, T., Chamlal, H.: Al-few: an enhanced approach for optimized query examples through feature weighting in active learning. Expert Syst. Appl. **266**, 126045 (2025). https://doi.org/10.1016/j.eswa.2024.126045
7. Gama, J., Sebastião, R., Rodrigues, P.P.: On evaluating stream learning algorithms. Mach. Learn. **90**(3), 317–346 (2012). https://doi.org/10.1007/s10994-012-5320-9
8. Golzari Oskouei, A., et al.: Feature-weighted fuzzy clustering methods: an experimental review. Neurocomputing **619**, 129176 (2025). https://doi.org/10.1016/j.neucom.2024.129176
9. Golzari Oskouei, A., Samadi, N., Tanha, J.: Feature-weight and cluster-weight learning in fuzzy C-means method for semi-supervised clusteringimage 1. Appl. Soft Comput. **161**, 111712 (2024). https://doi.org/10.1016/j.asoc.2024.111712
10. Hashemzadeh, M., Golzari Oskouei, A., Farajzadeh, N.: New fuzzy C-means clustering method based on feature-weight and cluster-weight learning. Appl. Soft Comput. **78**, 324–345 (2019). https://doi.org/10.1016/j.asoc.2019.02.038
11. Huang, J., Ng, M., Rong, H., Li, Z.: Automated variable weighting in k-means type clustering. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 657–668 (2005). https://doi.org/10.1109/TPAMI.2005.95
12. Issa, M.M., Aljanabi, M., Muhialdeen, H.M.: Systematic literature review on intrusion detection systems: research trends, algorithms, methods, datasets, and limitations. J. Intell. Syst. **33**(20230248) (2024). https://doi.org/10.1515/jisys-2023-0248
13. Kumar, P., Gupta, A.: Active learning query strategies for classification, regression, and clustering: a survey. J. Comput. Sci. Technol. **35**(4), 913–945 (2020). https://doi.org/10.1007/s11390-020-9487-4
14. Kurniabudi, Stiawan, D., Darmawijoyo, Bin Idris, M.Y., Bamhdi, A.M., Budiarto, R.: CICIDS-2017 dataset feature analysis with information gain for anomaly detection. IEEE Access **8**, 132911–132921 (2020). https://doi.org/10.1109/access.2020.3009843
15. Li, A., Han, M., Mu, D., Gao, Z., Liu, S.: Online active learning method for multiclass imbalanced data stream. Knowl. Inf. Syst. **66**(4), 2355–2391 (2024). https://doi.org/10.1007/s10115-023-02027-w

16. Li, B., Wang, Y., Cheng, L.: Adaptive and augmented active anomaly detection on dynamic network traffic streams. Front. Inf. Technol. Electron. Eng. **25**(3), 446–460 (2024). https://doi.org/10.1631/FITEE.2300244
17. Li, B., Wang, Y.J., Xu, K.L., Cheng, L., Qin, Z.Q.: DFAID: density-aware and feature-deviated active intrusion detection over network traffic streams. Comput. Secur. **118** (2022). https://doi.org/10.1016/j.cose.2022.102719
18. Li, N., Qi, Y., Li, C., Zhao, Z.: Active learning for data quality control: a survey. ACM J. Data Inf. Qual. **16**(2) (2024). https://doi.org/10.1145/3663369
19. Malialis, K., Panayiotou, C.G., Polycarpou, M.M.: Nonstationary data stream classification with online active learning and Siamese neural networks. Neurocomputing **512**, 235–252 (2022). https://doi.org/10.1016/j.neucom.2022.09.065
20. Martins, D.M., Salles, R.M., Goldschmidt, R.R.: Adaptive online learning approaches for detecting unknown attacks in network traffic: a systematic literature review. Secur. Priv. **8**(5), e70084 (2025). https://doi.org/10.1002/spy2.70084
21. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems. In: 2015 Military Communication and Information System Conference (MilCIS), pp. 1–6 (2015). https://doi.org/10.1109/MilCIS.2015.7348942
22. Nixon, C., Sedky, M., Champion, J., Hassan, M.: SALAD: a split active learning based unsupervised network data stream anomaly detection method using autoencoders. Expert Syst. with Appl. **248**(123439) (2024). https://doi.org/10.1016/j.eswa.2024.123439
23. Nixon, C., et al.: Reviews in online data stream and active learning for cyber intrusion detection-a systematic literature review. In: Abdennadher, N., Benkhelifa, E., Lauret, J.M., Jararweh, Y. (eds.) 2021 6th International Conference on Fog and Mobile Edge Computing, FMEC 2021, pp. 126–131. Staffordshire University, Stoke On Trent (2021). https://doi.org/10.1109/FMEC54266.2021.9732566
24. Niño-Adan, I., Manjarres, D., Landa-Torres, I., Portillo, E.: Feature weighting methods: a review. Expert Syst. Appl. **184**, 115424 (2021). https://doi.org/10.1016/j.eswa.2021.115424
25. Ruan, S., Li, H., Li, C., Song, K.: Class-specific deep feature weighting for naïve bayes text classifiers. IEEE Access **8**, 20151–20159 (2020). https://doi.org/10.1109/ACCESS.2020.2968984
26. Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M.: NetFlow datasets for machine learning-based network intrusion detection systems. In: Deze, Z., Huang, H., Hou, R., Rho, S., Chilamkurti, N. (eds.) BDTA/WiCON -2020. LNICST, vol. 371, pp. 117–135. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72802-1_9
27. Sarhan, M., Layeghy, S., Portmann, M.: Towards a standard feature set for network intrusion detection system datasets. Mob. Netw. Appl.. **27**(1), 357–370 (2021). https://doi.org/10.1007/s11036-021-01843-0
28. Scheirer, W.J., Rocha, A., Sapkota, A., Boult, T.E.: Toward open set recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(7), 1757–1772 (2013). https://doi.org/10.1109/TPAMI.2012.256
29. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information System Security and Privacy (ICISSP), pp. 108–116 (2018). https://doi.org/10.5220/0006639801080116
30. Shim, J., Kang, S.: Why does this query need to be labeled?: enhancing active learning through explanation-based interventions in query selection. Expert Syst. with Appl. **290**, 128443 (2025). https://doi.org/10.1016/j.eswa.2025.128443

31. Suárez-Cetrulo, A.L., Quintana, D., Cervantes, A.: A survey on machine learning for recurring concept drifting data streams. Expert Syst. Appl. **213**, 118934 (2023). https://doi.org/10.1016/j.eswa.2022.118934
32. Tharwat, A., Schenck, W.: A survey on active learning: state-of-the-art, practical challenges and research directions. Mathematics **11**(4), 820 (2023). https://doi.org/10.3390/math11040820
33. Xing, H.J., Ha, M.H.: Further improvements in feature-weighted fuzzy C-means. Inf. Sci. **267**, 1–15 (2014). https://doi.org/10.1016/j.ins.2014.01.033
34. Zhang, Y., et al.: Online adaptive asymmetric active learning with limited budgets. IEEE Trans. Knowl. Data Eng. **33**(6), 2680–2692 (2021). https://doi.org/10.1109/TKDE.2019.2955078