electronics

MDPI

Article

CLAIRE: A Four-Layer Active Learning Framework for Enhanced IoT Intrusion Detection

Abdulmohsen Almalawi

School of Computer Science & Information Technology, King Abdulaziz University, Jeddah 22254, Saudi Arabia; balmalowy@kau.edu.sa; Tel.: +966-12-6400000

Abstract

The integration of the Internet of Things (IoT) has become essential in our daily lives. It plays a core role in operating our daily infrastructure from energy grids and water distribution systems to healthcare and household devices. However, the rapid growth of IoT connections exposes our world to various sophisticated cybersecurity threats. Responding to these potential threats, many security measures have been proposed. The IoT-based Intrusion Detection System is one of the salient components of the security layer and alerts security administrators to any suspicious behaviors. In fact, machine learning-based IDS shows promising results, especially supervised models, but such models require expensive labelling processes by domain experts. The active learning strategy reduces the annotation cost and directs experts to label a small set of carefully selected instances. This paper proposes a robust approach called Clustering-based Layered Active Instance REpresentation (CLAIRE). It involves selecting both representative and informative instances. The former is selected through three sequential clustering-based layers, while the latter is selected by the fourth layer that implements an ensemble-based uncertainty mechanism to identify the most informative instances. Comprehensive evaluation on two well-known IoT datasets, namely, N-BaIoT and CICIoT2023, demonstrates promising results in selecting a small set of instances that capture the various data distributions of the data even in imbalanced datasets. We compare the results of the proposed approach with state-of-the-art baselines that work in the same scope of traditional machine learning.

Keywords: Internet of Things (IoT); Intrusion Detection System (IDS); active learning; IoT security; clustering

Academic Editor: Myung-Sup Kim

Received: 17 October 2025
Revised: 15 November 2025
Accepted: 18 November 2025
Published: 20 November 2025

Citation: Almalawi, A. CLAIRE: A Four-Layer Active Learning Framework for Enhanced IoT Intrusion Detection. *Electronics* **2025**, *14*, 4547. https://doi.org/10.3390/electronics14224547

1. Introduction

The Internet of Things (IoT) has become an integral part of our daily lives, extending into our core infrastructure, such as energy grids, water distribution systems, and healthcare. Beyond these sectors, IoT has made its way into the home environment, where many household appliances are now connected and controlled through the Internet [1,2]. A recent study shows that IoT adoption is growing rapidly and is expected to reach about 40.6 billion connections by 2034 [3]. Figure 1 shows this expected growth. Although the integration of IoT into our daily lives offers significant convenience, it also raises serious concerns. As more IoT devices become interconnected, cybersecurity challenges are increasing rapidly. IoT systems are widely adopted in both personal and industrial applications, where they are typically embedded with electronics and sensors that communicate via public networks. This connectivity makes them highly vulnerable to a broad spectrum of cyber threats, including but not limited to Denial-of-Service (DoS), Distributed Denial-of-Service

(DDoS), and injection attacks. Notably, between 2020 and 2022, there was a dramatic 776% increase in cyberattacks with bandwidths ranging from 100 Gbps to 400 Gbps, highlighting the escalating scale and intensity of these threats [4].

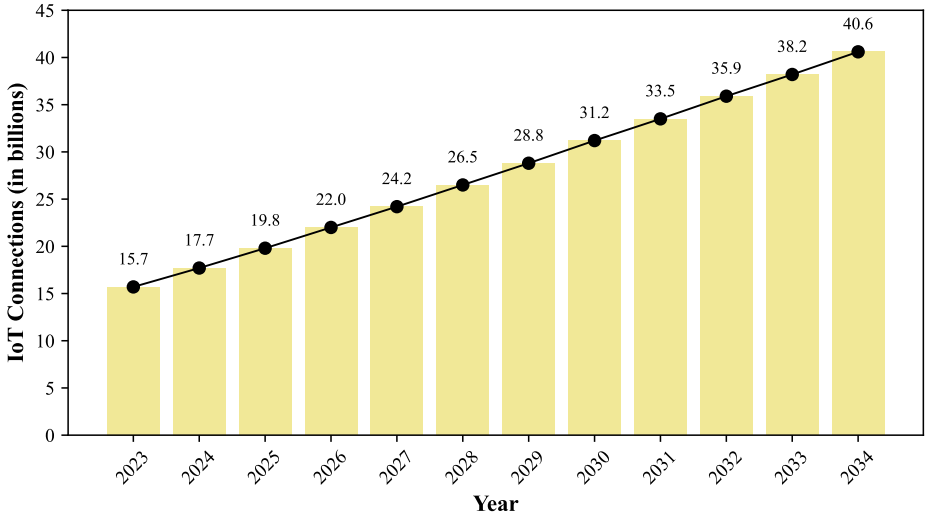

Figure 1. Global IoT connections trend from 2023 to 2034 (in billions).

To mitigate or prevent the potentially evolving risks, we first need to monitor the behaviors of the system to ensure it operates within expected normal behaviors. However, in complex and dynamic environments that consist of a large number of components, it is challenging for human operators and static rule-based techniques to identify the boundaries between normal and abnormal behaviors. In the literature, machine learning (ML)-based intrusion detection systems (IDSs) have demonstrated promising results in learning and differentiating between various behaviors. Therefore, many ML-based IoT IDSs have been proposed for cybersecurity [5–8]. Nevertheless, the deployment of effective IoT-based IDS solutions faces a number of issues: (i) efficiency and accuracy are among the open issues. In practice, supervised learning is found to be the best strategy to have robust IoT-based IDS [9], but labeled data must be available; (ii) the operation of IoT-based IDS in resource-constrained IoT networks requires small and representative data that capture most behaviors of the monitored system.

Therefore, active learning techniques are being explored as a potential solution, enabling the learning of both highly informative and representative instances from large datasets, leading to minimal, yet comprehensive, training sets that support lightweight, real-time, and accurate IoT-based IDSs. In addition, this results in minimizing the substantial effort required from domain experts to explore large datasets [10–13]. In reality, the establishment of a small and representative dataset is still a challenging issue. To efficiently learn such a dataset that captures the various behaviors present in unlabelled large datasets, we propose the CLAIRE framework that integrates clustering techniques with an active learning strategy. The novelty of CLAIRE is to integrate efficient clustering techniques with an active learning strategy to develop an active learning framework that addresses two open and challenging issues in IoT environments: data imbalance and data heterogeneity. In other words, this framework works with IoT network traffic, which is by nature heterogeneous and imbalanced. Moreover, CLAIRE introduces some key advantages over existing methods. Unlike the existing methods that either learn representative or informative instances, the proposed approach works on both simultaneously. Moreover, the existing methods start their learning process by randomly selecting instances that are used as a seed for the learning process. In fact, if the seed is drawn from a limited distribution, the learning process may fail to learn instances that reflect the full diversity of the dataset.

As illustrated in Figure 2, this framework involves four layers. The first three layers focus on learning representative instances that provide a strong initial seed for the subsequent informative learning phase. In the first layer, we propose a Various-Widths Clustering (VWC) technique [14] that is fast and efficient in partitioning unlabelled data into micro-clusters. Each resultant cluster is assumed to have instances that exhibit the same behaviors and distributions. Subsequently, this produces a large number of micro-clusters. However, the main goal of this study is to obtain a small set of representative instances that fit within realistic expert annotation budgets. Therefore, we propose a condensation clustering process based on the Expectation-Maximization (EM) algorithm to further condense the instances generated by the first layer into a more compact representation while preserving their overall representativeness. Thus far, the resultant set is still beyond the targeted budget. Thus, we propose a sampling strategy that maximizes the dispersion of the selected, truly representative instances. Up to this point, we have efficiently selected the initial and representative seeds used for the last layer, which is responsible for learning informative instances that maximize model performance. This framework provides complementary stages for IoT-based IDS, from pre-processing to monitoring. The results demonstrate promising performance of the proposed approach, achieving nearly optimal results on two well-known and widely used benchmark datasets. Interestingly, the approach demonstrates efficient outcomes on the imbalanced data when compared to state-of-the-art active learning baselines.

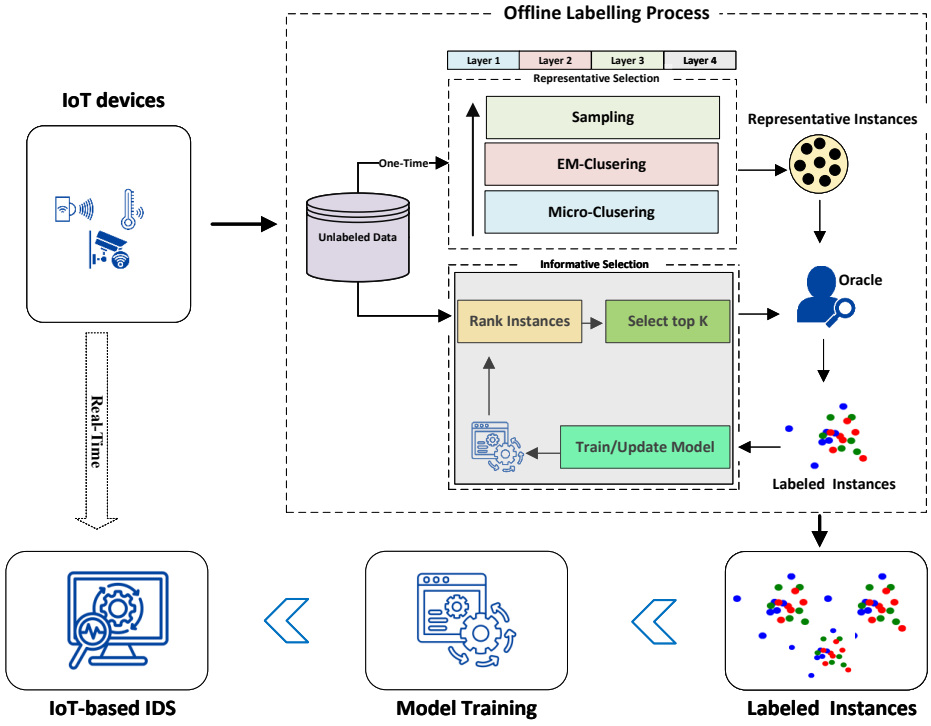

Figure 2. CLAIRE framework architecture showing the main components from unlabelled IoT traffic to IoT-based IDS.

The rest of this paper is organized as follows. Section 2 provides the related work on active learning techniques and IoT intrusion detection systems. We formulate the proposed CLAIRE approach and provide an effective solution to the active learning challenges in IoT environments in Section 3. Sections 4 and 5 present the comprehensive experimental evaluation, including dataset descriptions, baseline methods, performance metrics, and detailed analysis of results across various imbalanced scenarios. Finally, Section 6 presents the conclusion and future research directions.

2. Related Work

Active learning is a practical way to cut annotation effort and cost in machine learning. Instead of labelling an entire corpus, the learner targets examples that are likely to add the most value to the model. The goal is to approach the performance of a fully supervised system while using far fewer labeled instances. This matters in domains where labels are expensive or must be created under constraints, such as intrusion detection systems (IDS) for resource-limited settings. By narrowing the training set to the most informative samples, active learning becomes a feasible alternative to exhaustive labelling [15].

Standard formulations of active learning are usually grouped into three strategies, each differing in how candidates for labelling are chosen. Membership Query Synthesis [16] asks the model to generate synthetic examples and then obtains expert labels; while potentially informative, it is rarely adopted in practice because producing realistic samples is difficult [17]. Stream-Based Selective Sampling [18] considers items as they arrive and decides on the fly whether to request a label, which suits sensor streams and other online scenarios. Pool-Based Sampling [19], the most common in applied work, starts from a fixed pool of unlabelled data and selects the most informative items for annotation, often guided by uncertainty sampling or query-by-committee. This last approach tends to balance practicality and effectiveness in real deployments.

Tharwat and Schenck [20] describe an active learning scheme organized in two stages to address class imbalance. Starting from the exploratory stage, candidate points are selected using Balanced Exploration (BE) and three Latin Hypercube variants (LHCE-I, LHCE-II, and LHCE-III), with deliberate emphasis on minority classes. The following exploitation phase identifies uncertain regions near decision boundaries by calculating vote entropy from an ensemble of classifiers. Within those regions, candidate points are generated through Particle Swarm Optimization (PSO) and matched to the closest real, unlabelled samples for annotation. This approach balances class coverage with informativeness, resulting in realistic instances that are suitable for labelling.

Unlike traditional uncertainty sampling methods such as entropy-based or margin-based approaches, which select instances the model is most uncertain about and then evaluate performance using overall classification accuracy after annotation, Wang et al. [21] have proposed an active learning technique called Uncertainty Sampling via Maximizing Expected Average Precision (USAP), which shifts the sampling strategy from local uncertainty to global ranking impact. This is achieved by modelling the unknown true labels of all unlabelled instances as Bernoulli random variables, allowing the model to simulate possible label configurations. In other words, the model asks the following question: Which instance, if labeled, would most improve the model's ability to rank all other instances correctly? This strategy is more efficient when ranking consistency often matters more than aggregate classification accuracy.

Lu et al. [22] proposed a family of algorithms that apply the Passive-Aggressive update rule to learn from labeled instances selected through a margin-based querying strategy. This is performed by exploiting both misclassified and low-confidence correctly classified instances, using the prediction margin to guide both the label querying process and model updates. Similarly, Qin et al. [23] adapted a margin-based strategy in their sampling method. They employed a "best-versus-second-best" margin approach, which calculates the uncertainty of an instance by taking the difference between the two largest posterior probabilities. In this approach, similarity and uncertainty measures for all the unlabelled instances are calculated in the current batch.

Cacciarelli et al. [24] proposed a stream-based active learning approach that collects instances sequentially. This approach uses Mahalanobis distance to calculate the dissimilarity between unlabelled instances and those in the training dataset. This calculation indicates

the informativeness scores. Afterwards, any instance that has an informativeness score exceeding the defined threshold, which is the Upper Control Limit (UCL), is selected to be labeled. This predefined threshold is learned from historical data. The authors claim this approach can work in real-time for labelling.

In this work [25], an approach called QUIRE is proposed. This approach takes into account both informative and representative instances. For informativeness, the method selects instances about which the current model is most uncertain, while for representativeness, it identifies instances that are strongly connected or highly similar to many other unlabelled instances. In this process, the cluster structure of unlabelled instances and class assignments of labeled examples are taken into account. In essence, a representative instance is one that lies in a dense region of the unlabelled pool and shares high similarity with a large portion of the data. Although the proposed approach shows promise, it may suffer from computational complexity when applied to large datasets, and it relies on simplifying assumptions such as the use of a quadratic loss function and the manifold structure of data.

Ash et al. [26] propose an active learning approach based on a process that begins by randomly selecting a small set of unlabelled instances to be labeled. The performance of the model heavily depends on this initial labeled set. A neural network is then trained using these labeled examples. After that, the model predicts labels for all remaining unlabelled data, and for each example, a gradient vector is computed based on the model's prediction. The gradient vectors—often termed gradient embeddings—summarize the model's uncertainty; larger magnitudes typically signal greater doubt. To avoid selecting many near-duplicates, the algorithm clusters these vectors with k-means++ and then chooses representatives from different clusters. This maintains both uncertainty and diversity in the queried batch and reduces redundancy during labelling.

Flesca et al. [27] introduce an active learning framework that looks beyond informativeness or representativeness in isolation and instead prioritizes examples expected to most improve downstream performance once labeled. The procedure starts by training a deep neural network (DNN) on a small, randomly labeled seed set. A lightweight predictive model is then fit to estimate the impact of adding a new label, using gradient-based signals as features. That model scores unlabelled candidates so later iterations can target points predicted to deliver the largest benefit. A practical consideration is the seed set: if the initial labels are narrow or unrepresentative, the regressor may generalize poorly and fail to highlight the most useful instances.

Soltani et al. [28] present an approach built on a graph-based active learning strategy, where deep learning is adapted to the active learning process. A subset of the data is randomly selected for initial labelling, and a neural network model is trained on this small seed. This trained model is then used to evaluate the remaining unlabelled data and produce probability scores. These scores are fed into a graph-based method that identifies selectable segments, which are the regions between samples of different classes. From these selectable segments, the algorithm chooses the middle sample to label, assuming it carries the most informative value. The model is retrained after each iteration with the updated labeled dataset. However, this retraining step is computationally expensive and time-consuming, and the quality of the entire process heavily depends on the initial seed.

Halder et al. [29] propose a stream-based active learning framework that operates in hierarchical clustering to identify representative instances that are labeled by an annotator and used to train an ensemble classifier, then continuously evaluating new instances and requesting labels for uncertain ones. The system periodically performs clustering on newly collected instances to update the ensemble. However, its effectiveness depends on selecting

truly representative instances during clustering, as poor initial selection may lead to weak prediction performance.

Although current active learning methods have shown promise, notable gaps remain. A frequent shortcoming is the difficulty of choosing samples that are both highly informative for the decision boundary and genuinely representative of the underlying distribution, which weakens coverage and can bias learning toward a narrow slice of the data. Methods that emphasize uncertainty alone often over-sample ambiguous but redundant points, while approaches centred on representativeness may ignore rare yet decisive regions. Class imbalance adds a second, persistent complication. In many real IoT security datasets, a few majority classes dominate and minority classes appear only sparsely, sometimes with different feature scales or noise levels. Under these conditions, common selection rules struggle to discover and retain minority classes, batch selection can drift toward the majority, and performance becomes sensitive to the initial labeled set and the label budget. Together, these factors make it harder to explore the full label space and to prioritize examples that materially improve learning, limiting the ability to obtain a model that is both robust and balanced. Furthermore, their performance heavily depends on the quality of the initial labeled sets. Therefore, this study addresses these limitations by proposing an efficient active learning approach that simultaneously identifies small, highly informative, and truly representative instances from large training datasets, ultimately leading to the development of efficient and robust IoT-based IDSs.

3. The Proposed Approach

In the proposed CLAIRE, we assume that the data generated by the IoT environment is entirely unlabelled. As illustrated in Figure 2, the offline process consists of two parts. The first part is responsible for learning the representative instances based on three layers, where the first two layers are based on clustering techniques, while the third layer employs a dispersion-maximizing sampling technique. The second part is accountable for obtaining the informative instances. CLAIRE is designed for offline labelling process at edge gateways, not for direct execution on resource-constrained IoT devices. Upon completion of the offline process, a small set of annotated (labeled) representative and informative instances is obtained, which is then used to build a lightweight intrusion detection model to monitor the IoT environment. Formally, we denote this unlabelled dataset as $\mathcal{U} = \{x_1, x_2, \ldots, x_n\}$, where each x_i is a feature vector in a d-dimensional space. Since the data are generated in a raw form and inherently unlabelled, there is no explicit label for each observation. Moreover, we have no prior knowledge regarding their distribution, potential imbalance, or the possible number of class labels (or behaviors). The proposed approach focuses on identifying informative and representative observations from $\mathcal{U}$. In the first three layers, CLAIRE selects diverse and representative observations to capture the overall distribution and variety of the data. This ensures that the chosen observations are not biased toward a specific distribution or restricted to certain regions of the feature space. In the fourth layer, the most informative observations are identified based on representative data obtained in the previous layers after labelling by experts or an oracle. A comprehensive description and formulation of these layers are provided in the following sections.

3.1. First Layer: Various-Widths Clustering

As stated, there is no prior knowledge about learning data that were produced. Therefore, we aim to partition the data into micro-clusters, each containing observations that are closely related or exhibit the same behaviors and distribution. Consequently, we adopt an efficient various-width clustering algorithm called VWC [14], which is computationally efficient in clustering data into thousands of micro-clusters while accounting for various

data distributions. VWC involves two steps. First, it partitions the data based on a global width (or radius) learned from the learning data. Next, any cluster exceeding the user-defined size threshold $s_{\max}$ is repartitioned using a refined radius derived from the cluster itself, rather than from the entire data. This process repeats until all clusters meet the predefined size requirement. Subsequently, a merging step is performed: if any cluster is entirely contained within another, it is merged back into its parent cluster. These iterative partitioning and merging procedures yield well-distributed, compact, and well-separated clusters. For more details about this algorithm, please refer to [14]. Let $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ be the set of clusters produced by VWC, and let $s_{\max}$ be the predefined maximum size for each cluster. The resulting clusters satisfy: $\bigcup_{j=1}^{K} C_j = \mathcal{U}, \quad C_i \cap C_j = \varnothing \quad \text{for } i \neq j$, and $|C_j| \leq s_{\max} \quad \text{for all } j = 1, \ldots, K$.

Figure 3 provides a two-dimensional example of how the algorithm constructs clusters. Some clusters are dense, whereas others are sparse, yet all instances within a given cluster share a similar level of sparsity. This method is particularly efficient for high-dimensional data exhibiting varying distributions. The basic idea of VWC is to group observations that share similar distributions and behaviors, thereby increasing the likelihood of label consistency within each group (or cluster). In other words, we aim to cluster the instances into groups based on their specific distributional characteristics, even if multiple micro-clusters correspond to the same behaviors (or class label). A cluster in which all instances share the same behaviors is assumed to be a pure cluster, and it is critical for selecting the most suitable representatives, as it ensures that the representative accurately reflects the underlying behaviors. The purity of a cluster is defined by the presence or absence of class labels as follows:

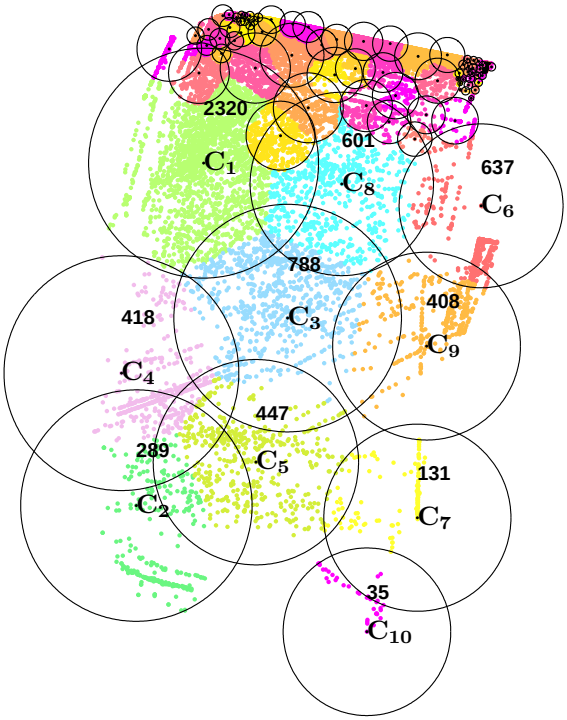

Figure 3. Output of the Various-Widths Clustering algorithm showing efficient partitioning of high-dimensional data into micro-clusters. Each cluster groups observations with similar distributions and behaviors to maximize label consistency within groups.

Definition 1 (Label-Pure Cluster). *Let $C = \{C_1, \ldots, C_K\}$ be a set of clusters, where each cluster C_k contains a subset of instances, and let $\mathcal{Y}$ be the set of possible class labels. A cluster C_k is* label-pure *if all its instances share the same label $y \in \mathcal{Y}$. The purity of a cluster C_k is defined as*

Purity$(C_k) = \frac{\max_{l \in \mathcal{Y}} |C_k \cap l|}{|C_k|}$, *where* $|C_k|$ *is the total number of instances in* C_k *and* $\max_{l \in \mathcal{Y}} |C_k \cap l|$ *is the number of instances in* C_k *from the most frequent class. The overall purity score (OPS) is calculated as* $OPS = \frac{1}{N} \sum_{k=1}^{K} |C_k| \cdot Purity(C_k)$, *where* N *is the total number of instances. This score ranges from 0 to 1, with higher values indicating purer clusters.*

Definition 2 (Low-Variance Cluster). *Let* $C = \{C_1, \ldots, C_K\}$ *be the set of clusters produced by VWC. For each numeric attribute* $j = 1, \ldots, M$, *define the mean* $\bar{x}_{k,j} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{i,j}$ *and the variance* $Var(x_{k,j}) = \frac{1}{|C_k|} \sum_{i \in C_k} \left(x_{i,j} - \bar{x}_{k,j} \right)^2$. *The variance of cluster* C_k *is then given by* $Var(C_k) = \frac{1}{M} \sum_{j=1}^{M} Var(x_{k,j})$, *and the overall variance score (OVS) for all clusters C is calculated as* $OVS = \frac{1}{N} \sum_{k=1}^{K} |C_k| \cdot Var(C_k)$, *where* N *is the total number of data points.*

Figure 4 demonstrates the relationship between the large number of partitioned clusters and the overall purity of the label and the low-variance scores. While a mathematical proof is beyond the scope of this discussion, these experimental results give an indication that when data are partitioned into very micro clusters, it is grouped into more homogeneous groups. From this empirical result, the overall variance score of the clusters can be utilized as a promising metric to set the appropriate value of the cluster size threshold of the proposed clustering method VWC to partition unlabelled data.

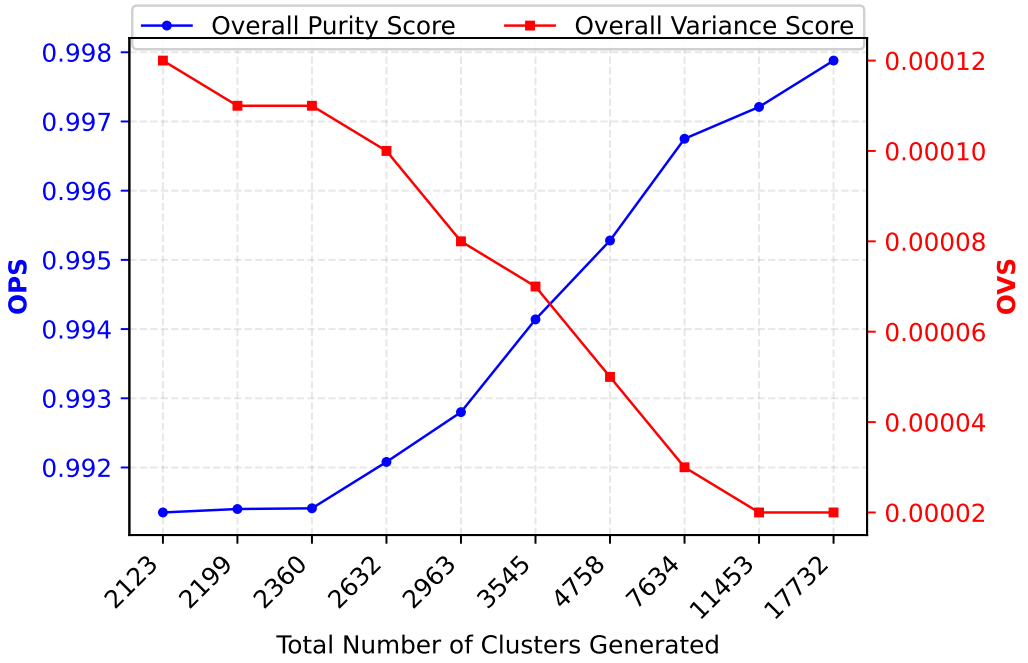

Figure 4. Cluster count versus OPS and OVS for VWC algorithm applied to CICIoT2023 dataset. Higher cluster numbers yield increased purity and decreased variance, indicating more homogeneous groupings. This also shows that VWC is able to produce well-compacted clusters.

In this layer, each cluster C_k is represented by its medoid m_k, defined as the point that minimizes the sum of dissimilarities to all other points in the cluster.

$$m_k = \arg\min_{x_i \in C_k} \sum_{x_j \in C_k} d(x_i, x_j), \tag{1}$$

where $d(x_i, x_j)$ is the dissimilarity measure (e.g., Euclidean distance) between points x_i and x_j. The medoids form an intermediate dataset $D_{\text{med}} = \{m_1, \ldots, m_K\}$, which serves as the input for the second layer. Algorithm 1 presents the Various-Width Clustering process

that partitions the unlabelled dataset into micro-clusters and extracts the medoid from each cluster.

Algorithm 1: Layer 1: Various-Width Clustering (VWC)

Input: $\mathcal{U}$: unlabelled dataset, $s_{\max}$: maximum cluster size threshold
Output: D_{med}: set of cluster medoids
// Initially assuming the whole dataset as a cluster
Initialize queue $Q \leftarrow \{\mathcal{U}\}$;
Initialize final clusters $\mathcal{C} \leftarrow \emptyset$;
// Iterative partitioning until all clusters <=$s_{\max}$
while $Q \neq \emptyset$ **do**
    $C \leftarrow$ dequeue from Q;
    if $|C| \leq s_{\max}$ **then**
        // Cluster <= $s_{\max}$, keep it
        Add C to $\mathcal{C}$;
    else
        // Cluster size >$s_{\max}$, split using VWC [14]
        Learn width w from variance of C;
        Partition C using VWC with width $w \rightarrow \{C_1, C_2, \ldots, C_k\}$;
        foreach *sub-cluster* C_i **do**
            Enqueue C_i to Q // Re-process each sub-cluster
// Get medoid from each cluster (Equation (1))
Initialize $D_{\text{med}} \leftarrow \emptyset$;
foreach *cluster* $C_j \in \mathcal{C}$ **do**
    Compute medoid m_j using Equation (1);
    Add m_j to D_{med};
return D_{med};

3.2. Second Layer: Medoid Refinement

This layer must come sequentially after the first one, as depicted in Figure 2. It only works on generated medoids from all clusters. The key point of this layer is to further reduce the medoids into a more compact representation while preserving their overall representativeness. To achieve this, we adapt the Expectation-Maximization (EM) algorithm due to its capability of dealing with non-spherical cluster shapes and varying densities, which may exist when condensing thousands of medoids into a smaller space, ensuring robust and meaningful clustering results. Although EM may not be feasible on very large datasets, this issue might not be inherited in this layer since it is applied on the generated medoids that consist of a very small portion of the dataset.

Let θ represent the cluster parameters, including the mean μ_k and covariance Σ_k for each cluster k. The EM procedure iteratively optimizes θ and assigns instances to clusters based on their likelihood. The E-step computes the expected cluster assignments (Equation (2)): better compactness. The large blank space is not necessary.

$$P(z_i = k \mid x_i, \theta) = \frac{P(x_i \mid z_i = k, \theta)\, P(z_i = k)}{\sum_{j=1}^{K} P(x_i \mid z_i = j, \theta)\, P(z_i = j)}, \tag{2}$$

where z_i is the cluster assignment for an instance x_i, and K is the total number of clusters. The M-step updates the parameters θ to maximize the data likelihood (Equation (3)):

$$\theta = \arg\max_{\theta} \sum_{i=1}^{n} \sum_{k=1}^{K} P(z_i = k \mid x_i, \theta) \log\Big[P(x_i, z_i = k \mid \theta)\Big]. \tag{3}$$

After performing EM refinement, we compute the variance σ_k^2 for each cluster C_k. Consider a cluster C_k containing $|C_k|$ instances, where each instance x_i is characterized by M numeric attributes $x_{i,1}, \ldots, x_{i,M}$. For each attribute $j \in \{1, \ldots, M\}$, we define the mean and variance of the attribute within the cluster as follows:

$$\mu_{k,j} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{i,j}, \tag{4}$$

$$\sigma_{k,j}^2 = \frac{1}{|C_k|} \sum_{i \in C_k} (x_{i,j} - \mu_{k,j})^2. \tag{5}$$

Here, $\mu_{k,j}$ represents the mean of the j-th attribute for cluster C_k, and $\sigma_{k,j}^2$ denotes the variance of the j-th attribute within the cluster. The overall variance of cluster C_k, denoted as σ_k^2, is calculated by averaging the attribute-specific variances $\sigma_{k,j}^2$:

$$\sigma_k^2 = \frac{1}{M} \sum_{j=1}^{M} \sigma_{k,j}^2 \quad \text{(variance of cluster } C_k\text{)} \tag{6}$$

As illustrated in Figure 5, the resulting clusters are categorized into three groups based on their variance profiles, α, β, and γ, corresponding to high $(\sigma_k^2 \geq \tau_{\text{high}})$, moderate $(\tau_{\text{low}} \leq \sigma_k^2 < \tau_{\text{high}})$, and low $(\sigma_k^2 < \tau_{\text{low}})$ variance, respectively. The variance thresholds τ_{high} and τ_{low} can be estimated using the separation ratio (SR $= \mu_\alpha / \mu_\gamma$) as a heuristic, where μ_α and μ_γ represent the mean variance of high-variance and low-variance groups, respectively, which measures the distinction between these groups. Therefore, any values of τ_{high} and τ_{low} that show higher separation ratios are selected as appropriate thresholds that discriminate high, medium, and low clusters in their overall variance scores. For example, high-variance α clusters exhibit considerable diversity, so all their instances are retained to preserve essential variability, forming $D_\alpha = \bigcup_{k \in \alpha} C_k$. Conversely, low-variance γ clusters are highly homogeneous, so each cluster is represented by its medoid m_k, yielding $D_\gamma = \{\, m_k \mid k \in \gamma \}$. Clusters in β have moderate variance and undergo a second round of clustering, subdividing each C_k into $\{C_{k,1}, \ldots, C_{k,L_k}\}$ (e.g., via EM). We then collect the medoids of these sub-clusters to form $D_\beta = \{\, m_{k,l} \mid k \in \beta,\ l = 1, \ldots, L_k \}$.

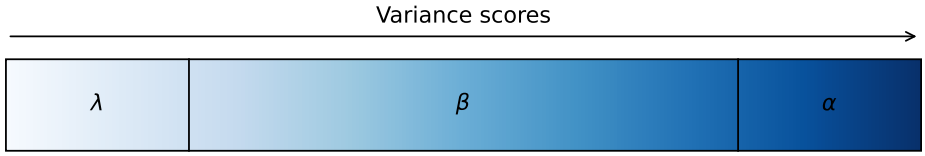

Figure 5. $\alpha, \beta, \gamma \rightarrow$ Areas of clusters with low, moderate, and high variance, respectively.

Ultimately, the outcome of this layer is more compact and condensed data, denoted as $D_{\text{final}} = D_\alpha \cup D_\beta \cup D_\gamma$. This data D_{final} forms a compact and small dataset and is assumed to preserve the characteristics and various distributions of the original data. The steps of this layer are formalized in Algorithm 2. Although this procedure provides a condensed representation, our goal is to further reduce its size while preserving its representativeness, as our main objective is to facilitate labelling with minimal expert involvement. Hence, in the next section, we will discuss the third layer, which selects the final representative instances.

Algorithm 2: Layer 2: EM-based Medoid Refinement

Input: D_{med}: medoid set from Algorithm 1,
$\tau_{\text{low}}, \tau_{\text{high}}$: variance thresholds (determined by separation ratio SR $= \mu_\alpha / \mu_\gamma$)
Output: D_{final}: refined representative set
`// Apply EM clustering to medoids (Equations (2) and (3))`
Apply EM clustering to $D_{\text{med}} \rightarrow \{C_1, \ldots, C_n\}$;
Initialize $D_\alpha \leftarrow \emptyset$, $D_\beta \leftarrow \emptyset$, $D_\gamma \leftarrow \emptyset$;
`// Categorize clusters by variance into three categories (Figure 5)`
foreach *cluster* C_k **do**
    Compute cluster variance σ_k^2 using Equation (6);
    if $\sigma_k^2 \geq \tau_{high}$ **then**
        `// High variance` α`: heterogeneous cluster, keep all instances`
        $D_\alpha \leftarrow D_\alpha \cup C_k$;
    else if $\sigma_k^2 < \tau_{low}$ **then**
        `// Low variance` γ`: homogeneous cluster, keep medoid only`
        Compute medoid m_k using Equation (1);
        $D_\gamma \leftarrow D_\gamma \cup \{m_k\}$;
    else
        `// Moderate variance` β`: subdivide and extract medoids`
        Subdivide C_k into $\{C_{k,1}, \ldots, C_{k,L_k}\}$ using EM;
        foreach *sub-cluster* $C_{k,l}$ **do**
            Compute medoid $m_{k,l}$ using Equation (1);
            $D_\beta \leftarrow D_\beta \cup \{m_{k,l}\}$;
`// Termination: All clusters categorized based on variance thresholds`
`// Combine all refined instances from the three categories`
$D_{\text{final}} \leftarrow D_\alpha \cup D_\beta \cup D_\gamma$;
return D_{final};

3.3. Third Layer: Sampling

In this section, we present the final stage of selecting representative instances from a processed dataset. Let $D_{\text{final}} = \{x_1, x_2, \ldots, x_n\}$ denote the dataset obtained from the second layer's clustering process (e.g., EM), where n is the total number of instances and each x_i is a d-dimensional vector. Our goal is to select a subset $S = \{s_1, s_2, \ldots, s_k\} \subseteq D_{\text{final}}$ of k representative instances ($k \leq n$), where each s_j corresponds to one of the original d-dimensional vectors x_i. The selection criterion maximizes the *dispersion* between instances in S, measured using the Manhattan (L_1) distance metric. This is because it directly aligns with the layer-2 clustering process that emphasizes attribute-level variance and also harmonizes with the variance-based optimization criteria used in the EM-based clustering. For each candidate instance $s_i \in D_{\text{final}} \setminus S$, its minimum distance to the current set S is computed using

$$d_{\min}(s_i, S) = \min_{s_j \in S} d_{L_1}(s_i, s_j) = \min_{s_j \in S} \sum_{m=1}^{d} |s_{i,m} - s_{j,m}| \tag{7}$$

$$s_t = \arg \max_{s_i \in D_{\text{final}} \setminus S} d_{\min}(s_i, S), \quad t = 2, \ldots, k \tag{8}$$

This maximum-minimum criterion greedily selects the candidate instances that is farthest—in terms of L_1 distance—from all instances already in S.

We implement this selection procedure as follows. First, randomly pick $s_1 \in D_{\text{final}}$ and initialize $S = \{s_1\}$. Next, for $t = 2, \ldots, k$, for each $s_i \in D_{\text{final}} \setminus S$ we update the set S by setting $S \leftarrow S \cup \{s_t\}$, and this process continues until k instances have been selected.

Having determined a subset S of k representative instances through the previous layers, we now establish these instances as the initial "seed" for the subsequent active learning stage. This seed, defined by the user-chosen parameter k, where a larger k improves representation in the feature space by increasing diversity, improving the robustness of the initial active learning model. By labelling this representative seed with the help of domain experts (or an oracle), the active learning procedure gains a strong foundation to further identify "informative" instances, i.e., those whose labels are most valuable for refining the model. In this way, the final dataset emerges as both representative (owing to the clustering-driven selection) and informative. In the following section, we will discuss how we learn the informative instances for which the model is least confident. Algorithm 3 describes how the process of dispersion-based sampling process works step by step.

Algorithm 3: Layer 3: Dispersion-Based Sampling

Input: D_{final}: refined set from Algorithm 2, k: sample size
Output: $\mathcal{L}$: initial labeled seed set
`// Initialize with random starting instance`
Randomly select $s_1 \in D_{\text{final}}$;
Initialize $S \leftarrow \{s_1\}$;
`// Greedy selection to maximize dispersion using Manhattan (`L_1`) distance`
for $t = 2$ **to** k **do**
  foreach $s_i \in D_{\text{final}} \setminus S$ **do**
    `// Compute minimum distance to current selected set using Equation (7)`
    Compute $d_{\min}(s_i, S)$ using Equation (7);
  `// Select instance farthest from all currently selected instances using Equation (8)`
  $s_t \leftarrow$ select using Equation (8);
  $S \leftarrow S \cup \{s_t\}$;
`// Termination:` k `maximally dispersed instances selected`
`// Query oracle for labels to create initial seed for Layer 4`
Query oracle for labels of S, producing $\mathcal{L} \leftarrow S$;
return $\mathcal{L}$;

3.4. Fourth Layer: Informative Learning

Building on the initial seed S derived from the first three layers to ensure representativeness, we now shift our focus to the fourth layer, which pursues *informative* learning. In this layer, the labeled seed S, determined through the previous three layers and annotated by domain experts (or an oracle), serves as the initial labeled set for informative learning.

We propose an ensemble learning technique through which an uncertainty score for each unlabeled data instance is calculated. First, we train an ensemble of M classifiers. Let $\mathcal{L}_t$ denote the current labeled set at iteration t, and let $\mathcal{U}$ be the unlabeled pool. For every unlabeled instance $x \in \mathcal{U}$, the ensemble's average class probability distribution is used to compute an entropy measure of uncertainty as follows:

$$H_t(x) = -\sum_{k=1}^{K}\Big(\tfrac{1}{M}\sum_{m=1}^{M} P_m^{(t)}(k \mid x)\Big) \times \log\Big(\tfrac{1}{M}\sum_{m=1}^{M} P_m^{(t)}(k \mid x)\Big) \quad (9)$$

where K represents the total number of classes, and $P_m^{(t)}(k \mid x)$ is the probability score that is assigned to class k by the m-th classifier at iteration t. Higher values of $H_t(x)$ imply greater predictive uncertainty in the ensemble's consensus. We sort the instances based on their entropy scores and top instances are selected as informative ones. Then they are annotated by an expert and added to $\mathcal{L}_t$. Through this iterative process, the model focuses on learning the informative instances allocated in challenging regions of the feature space. The procedure terminates once a predefined labelling budget is exhausted. The result is a final labeled dataset that has representative instances (from Layers 1–3) and informative ones (from this layer 4). This final dataset $\mathcal{L}_t$ is assumed to be a small and compact dataset that preserves the characteristics and various distributions of the original dataset. Moreover, it is expected to be reliable for producing accurate and lightweight classification models for IoT environments. The following Algorithm 4 describes in detail how the ensemble-based uncertainty active learning process is performed in the proposed approach.

Algorithm 4: Layer 4: Ensemble-Based Uncertainty Sampling

Input: $\mathcal{L}$: initial seed from Algorithm 3, $\mathcal{U}$: unlabeled dataset, B: labelling budget, M: ensemble size, b: batch size
Output: $\mathcal{L}$: final labeled dataset
`// Iterative batch-based uncertainty sampling to identify informative instances`
while $|\mathcal{L}| < B$ *and* $\mathcal{U} \setminus \mathcal{L} \neq \emptyset$ **do**
  `// Train ensemble of` M `classifiers on current labeled set`
  Train ensemble $\{h_1, \ldots, h_M\}$ on $\mathcal{L}$;
  `// Compute entropy-based uncertainty for each unlabeled instance`
  foreach $x \in \mathcal{U} \setminus \mathcal{L}$ **do**
    Compute average probability $P(k|x)$ across ensemble for all classes k;
    Compute entropy $H(x)$ using Equation (9);
  `// Select batch of instances with highest uncertainty (top-`b` maximum entropy)`
  Select top b instances: $\mathcal{B} \leftarrow \{x_1^*, \ldots, x_b^*\}$ with highest $H(x)$;
  `// Query oracle for batch labels and add to training set`
  Query oracle for labels of all $x \in \mathcal{B}$;
  $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{B}$;
`// Termination: Budget exhausted (`$|\mathcal{L}| \geq B$`) or no unlabeled instances remain`
return $\mathcal{L}$;

4. Experimental Evaluation

4.1. Dataset Description

We evaluated CLAIRE with two well-known IoT datasets: N-BaIoT [30] and CICIoT2023 [31]. The authors of these datasets made them available for research and testing purposes to detect intrusions in IoT systems. We used each dataset in its original form, without changing any of the features. N-BaIoT has network traffic from nine industrial IoT devices that were infected by the Mirai and BASHLITE botnets, whereas CICIoT2023 contains traffic from 105 IoT devices set up in a lab to emulate real-world conditions. Originally,

both datasets, N-BaIoT and CICIoT2023, contain millions of instances, each with a different number of features and class labels. From each dataset, we sampled training and testing subsets with no overlap to prevent overfitting. Furthermore, we created three variants (V1, V2, V3) for each dataset with different class imbalance ratios in the training data, while maintaining the same independent testing set for consistent evaluation. From N-BaIoT, we created three variants with different distributions of class labels, namely, NB-V1, NB-V2, and NB-V3. NB-V1 is sampled while maintaining the original class ratios, and for NB-V2 and NB-V3, we modified their class distributions to enable comprehensive evaluations on imbalanced data, not specifically with a certain attack type. Table 1 shows the configuration of this dataset. Similarly, we follow the same approach for CICIoT2023, yielding CIC-V1, CIC-V2, and CIC-V3, with distributions described in Table 2. This evaluation demonstrates the performance of the proposed CLAIRE across a range of class-imbalanced data with different attack types. In this evaluation, we use non-overlapping testing datasets that were not exposed during training. This assesses how well the proposed CLAIRE generalizes to new, unseen cases.

Table 1. Class distribution across N-BaIoT variants and testing dataset.

Class Label	Data Sets			
	NB-V1	**NB-V2**	**NB-V3**	**Test Set**
normal	29.53%	50.79%	0.26%	29.53%
tcp	22.61%	38.89%	0.26%	22.61%
combo	18.50%	0.19%	43.49%	18.50%
junk	17.35%	0.19%	40.79%	17.35%
udp	6.35%	0.19%	14.93%	6.35%
scan	5.66%	9.74%	0.26%	5.66%
Total Instances	893,633	519,551	380,082	382,986
All datasets contain 115 features				

Table 2. Class distribution across CICIoT2023 variants and testing dataset.

Class Label	Data Sets			
	CIC-V1	**CIC-V2**	**CIC-V3**	**Test Set**
DDoS	69.64%	73.92%	2.02%	69.64%
DoS	16.66%	17.68%	2.02%	16.66%
Mirai	5.24%	5.56%	2.02%	5.24%
Recon	3.16%	0.19%	35.11%	3.16%
BenignTraffic	2.18%	0.19%	24.24%	2.18%
Spoofing	1.87%	1.98%	20.76%	1.87%
Web-based	0.99%	0.19%	10.95%	0.99%
Brute Force	0.26%	0.28%	2.88%	0.26%
Total Instances	549,639	517,837	49,504	235,559
All datasets contain 46 features				

4.2. The Baseline Methods

As the scope of this study focuses only on traditional machine learning techniques for active learning, we compare the proposed approach with the following established active learning techniques aligned with this paradigm: (i) MARGIN sampling method [32], which selects instances closest to the decision boundary by measuring the difference between the two largest posterior probabilities; (ii) the USAP [21] approach, which maximizes expected average precision by modeling unknown labels as Bernoulli random variables; (iii) Balanced Exploration (BE) and Latin-Hypercube Sampling variants (LHCE-I, LHCE-II, and

LHCE-III) [20], where BE employs a two-phase framework using ensemble classifiers with vote entropy to prioritize minority class instances, while the LHCE variants focus on instances near decision boundaries using ensemble-based uncertainty quantification for class imbalance (among the LHCE variants, we specifically chose LHCEIII as it demonstrated the best performance in handling class imbalance scenarios); and (iv) RAND sampling, which serves as a fundamental baseline by randomly selecting instances without any intelligent strategy.

4.3. Experimental Settings

In this evaluation, we configured the baseline methods with their default values. With respect to the proposed approach, we set the following configurations: The key parameter in the first layer is the cluster size threshold $s_{\max}$, which is set to 1000. This choice was guided by the OVS analysis, as shown in Figure 4. Through the experiment, we found that slight variations around this value had minimal impact on OVS, indicating that this parameter is not highly sensitive. In the second layer, the variance thresholds τ_{low} and τ_{high} are determined using the separation ratio (SR) heuristic to categorize clusters into low, moderate, and high variance groups. The third layer is based on a sampling process that selects representative instances by maximizing the dispersion between those instances, and the user can set the desired number of instances. The last layer is based on an ensemble approach that utilizes two classifiers, namely J48 and Random Forest, where the final uncertainty decision is based on the average prediction entropy computed across the ensemble members. For classification training and testing, four classifiers are used: J48 [33], Naive Bayes [34], k-Nearest Neighbors [35], and RandomForest [36]. Their classification results are averaged across all four classifiers to provide a comprehensive performance assessment.

5. Results and Discussion

In this section, we compare CLAIRE with five active learning baselines using two complementary evaluations. The former examines label coverage by checking the capability of an active learning method in exploring all existing behaviors (e.g., normal or any type of attack instances). The latter evaluates generalization on a held-out test set using three standard measures: macro-Precision, macro-Recall, and macro-F-measure to evaluate how the actively learned instances capture the characteristics and the various distributions of the original data. In other words, it evaluates the accuracy of the classification model in classifying the unseen testing dataset when trained on this learned data. In this evaluation, we only demonstrate the macro-averaging, which gives every class equal weight regardless of frequency, ensuring that class labels with large portions will not dominate the classification accuracy results. For detailed mathematical definitions of these standard metrics, commonly applied in intrusion detection studies, readers may refer to [37].

5.1. Class Discovery

In fact, raw IoT data is unlabeled data, which may have various types of behaviors, namely normal or abnormal. And the abnormal can come with many types of attack. Therefore, to build robust supervised IoT-based IDS, the model must be trained on all behaviors. As previously discussed, the active learning approach is considered one of the solid techniques to select the most representative and informative instances that represent the whole behaviors of the systems. The fundamental challenge is to discover or select all attack types. Existing active learning methods rely on random initialization of the first initial seed, and this can only draw instances from the major or dominant behaviors (attack types). Thus, minority classes, which may represent new or zero-day attacks, may be

missed in the discovery process. In this evaluation, we compare CLAIRE with the baselines on class label coverage (CCR). The metric CCR serves as the principal indicator and is used to determine whether the actively selected set spans the full label space. In active learning, coverage is critical because the learner should encounter at least one instance from every class to establish reliable decision boundaries. When coverage is high, the model gains broad access to training evidence and tends to be more robust across classes. If any label is absent from the selected set, the model cannot learn that class due to the lack of training examples.

In this context, ICL refers to the count of unique labels observed among the selected instances during active learning, whereas TCL denotes the total number of unique labels present in the source dataset as a whole. CCR treats all labels equally and focuses on coverage rather than per-class accuracy. The metric is defined as follows:

$$\mathrm{CCR} = \frac{\mathrm{ICL}}{\mathrm{TCL}}$$

By definition, $0 \leq \mathrm{CCR} \leq 1$. A value of 1 means complete coverage, with at least one instance from every class. A CCR below 1 implies missing class labels, which makes the selected subset less representative of the complete dataset.

As shown in Table 3, the CIC variants, DDoS and DoS dominate CIC-V1 and CIC-V2 at 86% and 92%, while minority categories such as Web-based and Brute Force attacks each account for less than 1% of the data. Even under this pronounced skew, CLAIRE achieves a CCR of 1.00 for every labelling budget in both CIC-V1 and CIC-V2 with only one instance of 0.88 at 30 labeled instances in CIC-V2. In contrast, baseline methods struggle significantly: BE achieves 1.00 only at higher budgets starting from 200 instances in CIC-V1 and 250 instances in CIC-V2, LHCEIII shows poor performance with maximum scores of 0.75 in CIC-V1 and 0.88 in CIC-V2, MARGIN reaches 1.00 inconsistently, and USAP demonstrates high volatility with scores fluctuating between 0.38 and 1.00. Similarly, on the CIC-V3 dataset, the proposed CLAIRE approach demonstrates strong performance across all budgets except for 30 and 50 instances. However, it still shows satisfactory results of 0.88 for both budgets. On the other hand, the baseline models behave less predictably. As can be seen, RAND achieves a CCR of 1.00 when sufficient labeled instances are available. MARGIN performs well overall, although it exhibits fluctuating results. LHCEIII also reaches 1.00 for higher budgets, except at 300 instances where it shows 0.88. In contrast, USAP and BE demonstrate lower performance compared to all other methods.

In a similar manner, Table 4 shows the evaluation on NB dataset variants. As can be seen, results further demonstrating that CLAIRE is still promising on all variants NB-V1 with relatively balanced distribution, NB-V2 where Normal and TCP attacks dominate at 89% combined while Combo, Junk, and UDP attacks represent only 0.19% each, and NB-V3 where Combo and Junk attacks constitute 84% while other classes become minorities. CLAIRE achieves perfect CCR scores of 1.00 across all NB variants and labelling budgets without exception. Baseline methods show varying degrees of struggle: while LHCEIII and some others achieve 1.00 in the balanced NB-V1 scenario, they fail significantly in imbalanced scenarios, with BE, MARGIN, RAND, and USAP showing poor coverage in NB-V2 and NB-V3, rarely exceeding 0.83 and often dropping as low as 0.50.

Table 3. CCR Performance Comparison on *CIC* Dataset Variants (Averaged Across Classifiers).

Approach	CIC-V1							CIC-V2							CIC-V3						
	Labeled Instances							Labeled Instances							Labeled Instances						
	30	50	100	150	200	250	300	30	50	100	150	200	250	300	30	50	100	150	200	250	300
BE	0.50	0.63	0.75	0.88	**1.00**	**1.00**	**1.00**	0.50	0.50	0.75	0.88	0.88	**1.00**	**1.00**	0.75	0.75	0.75	0.75	0.75	0.75	0.88
LHCEIII	0.50	0.50	0.50	0.50	0.50	0.75	0.75	0.25	0.25	0.50	0.50	0.50	0.50	0.88	0.75	0.75	0.88	**1.00**	**1.00**	**1.00**	0.88
MARGIN	0.50	0.88	0.75	0.75	**1.00**	**1.00**	**1.00**	0.50	0.50	0.50	0.50	0.50	0.75	0.63	**1.00**	**1.00**	0.63	0.88	**1.00**	**1.00**	**1.00**
RAND	0.75	0.75	0.63	0.88	**1.00**	**1.00**	0.88	0.38	0.50	0.50	0.63	0.63	0.75	0.75	0.75	0.75	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
USAP	0.75	0.38	**1.00**	0.50	0.38	**1.00**	**1.00**	0.50	0.38	**1.00**	0.88	**1.00**	0.88	**1.00**	0.88	0.88	0.63	0.88	0.88	0.88	0.75
CLAIRE	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.88**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.88	0.88	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Bold values indicate perfect class coverage (CCR = 1.00), meaning all class labels were discovered.

Table 4. CCR Performance Comparison on *NB* Dataset Variants (Averaged Across Classifiers).

Approach	NB-V1							NB-V2							NB-V3						
	Labeled Instances							Labeled Instances							Labeled Instances						
	30	50	100	150	200	250	300	30	50	100	150	200	250	300	30	50	100	150	200	250	300
BE	0.67	0.83	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.50	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.83	0.83	**1.00**
LHCEIII	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.50	0.50	0.50	0.67	0.50	0.50	0.50	0.67	0.67	0.67	0.67	**1.00**	**1.00**	**1.00**
MARGIN	0.67	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.50	0.50	0.50	0.50	0.67	0.83	0.83	0.67	0.67	0.67	0.67	0.67	**1.00**	**1.00**
RAND	0.83	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.50	0.67	0.50	0.67	0.83	0.50	0.67	0.50	0.50	0.67	0.67	0.67	0.50	**1.00**
USAP	**1.00**	0.83	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.50	0.67	0.83	0.83	0.67	0.67	0.83	0.67	0.50	0.50	0.83	0.67	0.50	0.50
CLAIRE	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Bold values indicate perfect class coverage (CCR = 1.00), meaning all class labels were discovered.

Overall, CLAIRE maintains strong and consistent coverage across datasets despite challenging class distributions. Although these CCR results demonstrate excellent class label coverage, this metric alone does not fully validate the practical effectiveness of the active learning approach. High coverage ensures that all classes are represented in the selected instances, but it does not guarantee that these instances are the most representative and informative for model training. The quality and representativeness of the selected instances are equally important for achieving good classification performance. Therefore, the following section evaluates the effectiveness of the selected instances.

5.2. Classification Performance

Although the results of CCR indicate that class labels are well covered, coverage by itself does not ensure that the selected instances are sufficiently representative and informative for effective training. To build a more rounded view of performance, the classification capability of CLAIRE was examined using three widely adopted metrics, namely macro-Precision, macro-Recall, and macro-F-measure. These measures were computed using macro-averaging, meaning that each class was evaluated individually and then averaged so that both frequent and infrequent classes exert the same influence on the final score.

In fact, the literature describes a wide range of attack types, each with different characteristics and severity levels. However, in this evaluation, we adopt a simplified approach where we treat all attack types equally with the same impact factor. This is because we aim to evaluate whether the proposed approach is capable of learning diverse behaviors in the unlabelled dataset. In other words, the goal is not to learn specific behaviors or attack types, but rather to capture the full range of behavioural diversity. For this purpose, we evaluate the classification performance of the proposed CLAIRE approach against the baselines using four well-known classifiers: J48, Naive Bayes, k-Nearest Neighbors, and Random Forest. Each classifier is trained on the labeled instances generated by each method. This evaluation investigates the quality of the labeled instances generated by each competing method to assess their representativeness and informativeness in capturing the characteristics of the original dataset. To ensure fairness and generalization of the trained models, we test them on unseen data.

For simplicity and due to space limitations, we report performance as averages across the four classifiers rather than listing separate results for each model. This provides an overall assessment of how well each set of labeled instances captured useful information that enables various models with different learning perspectives to generalize for classifying unseen data. Additionally, we include the standard deviation alongside the averaged performance to provide readers with insight into the stability and variability of each compared method.

Tables 5–7 show classification results for the three CIC variants. As shown in Table 5, CLAIRE demonstrates promising and stable results on most labelling budgets, and it shows significant results as the budget size increases. As presented, CLAIRE achieves the highest Macro-F, Macro-Recall, and Macro-Precision results for most budget sizes, except on 50 instances where MARGIN shows better and competitive performance. In contrast, baseline methods show fluctuated results. Among the competitors, the MARGIN method performs closest, with scores from 0.69 to 0.89, yet it trails by about two to three percentage points at higher labelling budgets. The USAP method produces occasional gains at certain points, for example 0.72 at 100 labeled instances and 0.77 at 250, but lacks overall stability. Meanwhile, BE and RAND remain below 0.73, and LHCEIII consistently records lower values, staying at or under 0.55. These results indicate that CLAIRE not only achieves stronger performance overall but also learns minority classes more effectively than competing methods. Interestingly, the proposed approach demonstrates relatively lower

standard deviations, meaning it is able to learn representative and informative instances that capture most characteristics and properties of the learning data.

Table 5. Performance Comparison on *CIC-V1* Dataset (Averaged Across Classifiers).

Metric	Method	Labeled Instances						
		30	**50**	**100**	**150**	**200**	**250**	**300**
Macro-F	BE	0.34 ± 0.04	0.39 ± 0.11	0.52 ± 0.04	0.57 ± 0.04	0.63 ± 0.04	0.71 ± 0.04	0.73 ± 0.05
	LHCEIII	0.37 ± 0.05	0.36 ± 0.06	0.39 ± 0.04	0.39 ± 0.04	0.39 ± 0.04	0.51 ± 0.06	0.55 ± 0.07
	MARGIN	0.37 ± 0.08	**0.69** ± 0.04	0.57 ± 0.04	0.58 ± 0.02	0.80 ± 0.04	0.76 ± 0.04	0.87 ± 0.01
	RAND	0.50 ± 0.11	0.55 ± 0.09	0.46 ± 0.04	0.67 ± 0.08	0.62 ± 0.06	0.68 ± 0.08	0.72 ± 0.03
	USAP	0.45 ± 0.03	0.31 ± 0.02	0.72 ± 0.07	0.43 ± 0.03	0.32 ± 0.02	0.77 ± 0.07	0.69 ± 0.11
	CLAIRE	**0.53** ± 0.15	0.52 ± 0.23	**0.80** ± 0.09	**0.82** ± 0.06	**0.81** ± 0.13	**0.88** ± 0.04	**0.89** ± 0.05
Macro-Recall	BE	0.36 ± 0.07	0.41 ± 0.10	0.54 ± 0.02	0.60 ± 0.03	0.64 ± 0.04	0.73 ± 0.04	0.73 ± 0.04
	LHCEIII	0.39 ± 0.07	0.38 ± 0.08	0.42 ± 0.04	0.42 ± 0.04	0.42 ± 0.04	0.55 ± 0.08	0.55 ± 0.08
	MARGIN	0.44 ± 0.07	**0.74** ± 0.03	0.59 ± 0.04	0.61 ± 0.03	0.81 ± 0.04	0.76 ± 0.04	0.86 ± 0.02
	RAND	0.52 ± 0.09	0.55 ± 0.06	0.49 ± 0.02	0.69 ± 0.07	0.65 ± 0.03	0.70 ± 0.07	0.73 ± 0.02
	USAP	0.50 ± 0.05	0.36 ± 0.01	0.74 ± 0.08	0.48 ± 0.01	0.36 ± 0.01	0.80 ± 0.06	0.67 ± 0.09
	CLAIRE	**0.58** ± 0.12	0.60 ± 0.17	**0.83** ± 0.05	**0.83** ± 0.06	**0.83** ± 0.10	**0.89** ± 0.02	**0.90** ± 0.03
Macro-Prec.	BE	0.34 ± 0.04	0.48 ± 0.13	0.56 ± 0.05	0.64 ± 0.04	0.73 ± 0.10	0.77 ± 0.06	0.78 ± 0.05
	LHCEIII	0.39 ± 0.06	0.38 ± 0.06	0.38 ± 0.06	0.38 ± 0.05	0.38 ± 0.06	0.60 ± 0.09	0.61 ± 0.06
	MARGIN	0.36 ± 0.08	0.70 ± 0.03	0.56 ± 0.03	0.65 ± 0.01	0.81 ± 0.05	0.84 ± 0.10	**0.89** ± 0.04
	RAND	0.53 ± 0.15	0.58 ± 0.13	0.45 ± 0.05	0.69 ± 0.08	0.71 ± 0.07	0.77 ± 0.12	0.74 ± 0.04
	USAP	0.52 ± 0.06	0.30 ± 0.02	0.77 ± 0.04	0.41 ± 0.02	0.31 ± 0.02	0.79 ± 0.08	0.83 ± 0.06
	CLAIRE	**0.66** ± 0.24	**0.72** ± 0.22	**0.81** ± 0.09	**0.86** ± 0.07	**0.86** ± 0.10	**0.90** ± 0.05	0.89 ± 0.05

± standard deviation across classifiers. **Bold values** highlight the highest metric score achieved by a competing method for each labeling budget size.

Table 6. Performance Comparison on *CIC-V2* Dataset (Averaged Across Classifiers).

Metric	Method	Labeled Instances						
		30	**50**	**100**	**150**	**200**	**250**	**300**
Macro-F	BE	0.32 ± 0.03	0.36 ± 0.05	0.54 ± 0.02	0.63 ± 0.04	0.62 ± 0.03	0.67 ± 0.05	0.68 ± 0.04
	LHCEIII	0.21 ± 0.05	0.22 ± 0.03	0.43 ± 0.05	0.43 ± 0.05	0.43 ± 0.05	0.43 ± 0.05	0.58 ± 0.03
	MARGIN	**0.47** ± 0.00	0.47 ± 0.01	0.48 ± 0.00	0.47 ± 0.00	0.46 ± 0.02	0.63 ± 0.03	0.46 ± 0.02
	RAND	0.30 ± 0.04	0.44 ± 0.06	0.47 ± 0.00	0.55 ± 0.01	0.48 ± 0.02	0.50 ± 0.07	0.54 ± 0.06
	USAP	0.40 ± 0.08	0.36 ± 0.00	0.68 ± 0.09	0.52 ± 0.09	0.81 ± 0.06	0.59 ± 0.02	0.65 ± 0.05
	CLAIRE	0.45 ± 0.11	**0.69** ± 0.09	**0.75** ± 0.05	**0.79** ± 0.10	**0.84** ± 0.06	**0.86** ± 0.06	**0.88** ± 0.07
Macro-Recall	BE	0.38 ± 0.03	0.39 ± 0.02	0.55 ± 0.02	0.66 ± 0.03	0.67 ± 0.04	0.70 ± 0.05	0.69 ± 0.05
	LHCEIII	0.22 ± 0.05	0.23 ± 0.04	0.42 ± 0.04	0.43 ± 0.04	0.43 ± 0.04	0.42 ± 0.03	0.60 ± 0.04
	MARGIN	0.48 ± 0.01	0.48 ± 0.01	0.49 ± 0.01	0.48 ± 0.01	0.47 ± 0.03	0.63 ± 0.04	0.46 ± 0.05
	RAND	0.31 ± 0.04	0.45 ± 0.05	0.48 ± 0.02	0.57 ± 0.02	0.49 ± 0.01	0.51 ± 0.07	0.56 ± 0.08
	USAP	0.38 ± 0.07	0.37 ± 0.00	0.65 ± 0.07	0.57 ± 0.09	0.85 ± 0.02	0.63 ± 0.04	0.66 ± 0.03
	CLAIRE	**0.48** ± 0.11	**0.70** ± 0.08	**0.78** ± 0.04	**0.88** ± 0.03	**0.87** ± 0.04	**0.87** ± 0.05	**0.89** ± 0.05
Macro-Prec.	BE	0.31 ± 0.03	0.37 ± 0.03	0.65 ± 0.12	0.68 ± 0.09	0.66 ± 0.09	0.76 ± 0.12	0.72 ± 0.07
	LHCEIII	0.22 ± 0.01	0.22 ± 0.01	0.45 ± 0.07	0.45 ± 0.06	0.45 ± 0.07	0.45 ± 0.07	0.66 ± 0.06
	MARGIN	0.47 ± 0.01	0.47 ± 0.01	0.47 ± 0.01	0.47 ± 0.01	0.46 ± 0.02	0.64 ± 0.02	0.57 ± 0.05
	RAND	0.31 ± 0.05	0.43 ± 0.07	0.47 ± 0.01	0.54 ± 0.01	0.51 ± 0.04	0.58 ± 0.09	0.58 ± 0.09
	USAP	0.43 ± 0.09	0.36 ± 0.01	**0.87** ± 0.04	0.61 ± 0.12	0.81 ± 0.08	0.66 ± 0.03	0.77 ± 0.11
	CLAIRE	**0.58** ± 0.18	**0.78** ± 0.07	0.80 ± 0.11	**0.79** ± 0.10	**0.83** ± 0.07	**0.86** ± 0.07	**0.88** ± 0.09

± standard deviation across classifiers. **Bold values** highlight the highest metric score achieved by a competing method for each labeling budget size.

Table 7. Performance Comparison on *CIC-V3* Dataset (Averaged Across Classifiers).

Metric	Method	Labeled Instances						
		30	**50**	**100**	**150**	**200**	**250**	**300**
Macro-F	BE	0.54 ± 0.10	0.55 ± 0.10	0.55 ± 0.08	0.55 ± 0.09	0.55 ± 0.10	0.58 ± 0.07	0.59 ± 0.10
	LHCEIII	0.57 ± 0.06	0.58 ± 0.05	0.69 ± 0.07	0.65 ± 0.09	0.76 ± 0.07	0.75 ± 0.03	0.75 ± 0.05
	MARGIN	0.57 ± 0.07	**0.74** ± 0.03	0.46 ± 0.04	0.66 ± 0.07	0.84 ± 0.03	0.88 ± 0.04	0.88 ± 0.03
	RAND	0.50 ± 0.03	0.59 ± 0.02	0.79 ± 0.11	0.82 ± 0.07	0.84 ± 0.07	0.83 ± 0.04	0.86 ± 0.04
	USAP	**0.63** ± 0.03	0.50 ± 0.09	0.38 ± 0.03	0.58 ± 0.10	0.72 ± 0.02	0.74 ± 0.02	0.49 ± 0.02
	CLAIRE	0.44 ± 0.08	0.69 ± 0.07	**0.84** ± 0.07	**0.88** ± 0.03	**0.88** ± 0.04	**0.90** ± 0.05	**0.91** ± 0.04
Macro-Recall	BE	0.56 ± 0.07	0.57 ± 0.07	0.57 ± 0.07	0.57 ± 0.07	0.57 ± 0.07	0.60 ± 0.04	0.60 ± 0.08
	LHCEIII	0.65 ± 0.01	0.66 ± 0.01	0.73 ± 0.04	0.74 ± 0.04	0.79 ± 0.05	0.77 ± 0.01	0.78 ± 0.03
	MARGIN	0.64 ± 0.10	**0.75** ± 0.03	0.47 ± 0.03	0.66 ± 0.06	0.82 ± 0.04	0.88 ± 0.04	0.87 ± 0.05
	RAND	0.54 ± 0.02	0.60 ± 0.01	0.78 ± 0.10	0.81 ± 0.05	0.84 ± 0.07	0.83 ± 0.06	0.85 ± 0.04
	USAP	**0.67** ± 0.03	0.57 ± 0.05	0.40 ± 0.03	0.65 ± 0.08	0.71 ± 0.02	0.74 ± 0.02	0.52 ± 0.03
	CLAIRE	0.48 ± 0.08	0.72 ± 0.05	**0.86** ± 0.05	**0.90** ± 0.03	**0.88** ± 0.04	**0.91** ± 0.02	**0.93** ± 0.01
Macro-Prec.	BE	0.55 ± 0.12	0.56 ± 0.13	0.58 ± 0.12	0.58 ± 0.13	0.57 ± 0.12	0.58 ± 0.10	0.66 ± 0.16
	LHCEIII	0.56 ± 0.06	0.56 ± 0.06	0.70 ± 0.07	0.70 ± 0.16	0.76 ± 0.09	0.84 ± 0.10	0.74 ± 0.06
	MARGIN	0.64 ± 0.12	**0.83** ± 0.06	0.47 ± 0.05	0.74 ± 0.06	**0.92** ± 0.01	0.91 ± 0.04	**0.93** ± 0.02
	RAND	0.53 ± 0.05	0.61 ± 0.04	0.83 ± 0.12	0.85 ± 0.10	0.87 ± 0.08	0.88 ± 0.02	0.89 ± 0.03
	USAP	**0.71** ± 0.05	0.60 ± 0.07	0.44 ± 0.07	0.58 ± 0.10	0.75 ± 0.02	0.76 ± 0.03	0.56 ± 0.03
	CLAIRE	0.55 ± 0.15	0.71 ± 0.07	**0.86** ± 0.05	**0.87** ± 0.03	0.90 ± 0.04	**0.92** ± 0.05	0.92 ± 0.05

± standard deviation across classifiers. **Bold values** highlight the highest metric score achieved by a competing method for each labeling budget size.

Table 6 shows the results for CIC-V2, a far more imbalanced configuration in which DDoS and DoS together represent approximately 92/100 of the samples. The remaining four categories appear only in very small proportions, each below 0.3/100. Across labelling budgets, CLAIRE is the most consistent performer among the compared methods. As the labelling budget increases, CLAIRE's scores climb in a steady way. The macro F-measure rises from 0.69 to 0.88, and macro Recall moves from 0.48 to 0.89. Macro Precision improves along the same path and reaches 0.88. USAP looks strong early, posting a macro F-score of 0.81 with 200 labeled samples, but its precision falls as the budget grows. As can be seen, BE, RAND, and MARGIN demonstrate low results, not exceeding 0.70. Notably, LHCEIII demonstrates the lowest results, not exceeding 0.60 across all budgets. Taken together, these results illustrate how class imbalance can limit learning effectiveness, particularly when rare classes must be learned under tight labelling budgets. Despite these challenges, CLAIRE maintains recall and precision in reasonable balance, even in cases where minority classes have only a few labels. With increased labeled data, its performance stabilizes further, indicating stronger and more reliable generalization compared to competing approaches. Furthermore, CLAIRE demonstrates relatively lower standard deviations across most budget sizes, revealing that the learned labeled instances are informative and contain clear patterns for building generalized and efficient classification models.

As detailed in Table 2, CIC-V3 shows a different class distribution. 91% of the samples came from Benign traffic and from three attack types, namely, reconnaissance, spoofing, and web-based. However, attack types such as DDoS, DoS, and Mirai appear much less often, while brute-force activity shows a modest increase. Again, even with this shift, CLAIRE still performs best among the methods. Table 7 demonstrates that macro F-measure grows steadily from 0.44 with 30 labeled instances to 0.91 at 300. It also reaches peak scores of 0.93 for macro Precision and 0.90 for macro Recall. MARGIN performs well in the early stages. With only 50 labeled examples, it already reaches a macro F-score of 0.74, and at 200 labels it achieves the highest macro Precision of 0.92. However, its recall still falls short of what CLAIRE achieves. RAND shows a broadly similar pattern but tends to stay a bit behind MARGIN at most labelling budgets. USAP behaves much less predictably, with its macro

F-score moving anywhere between 0.38 and 0.74, which shows that it reacts quite strongly to changes in the dataset. BE is almost the opposite: its performance stays fairly steady at around 0.59 across budgets, although it never reaches the levels achieved by CLAIRE or MARGIN.

Similarly, Tables 8–10 present the results for the NB dataset variants, where each one has different distributions of the traffic types, as summarized in Table 1. In the NB-V1, traffic types are moderately distributed. This gives each model balanced exposure to learn from all classes. As shown in Table 8, RAND in this type of distribution shows promising results, with macro F-measure scores increasing from 0.89 to 0.92. MARGIN reaches macro F-measure values between 0.47 and 0.91 and keeps its highest macro Precision of 0.92 with 150 labeled instances. CLAIRE demonstrates competitive results, improving gradually from 0.59 to 0.89 as the labelling budget increases. USAP is effective at low labelling budgets but shows limited improvement at higher labelling budgets, showing that these results are not stable since it is not performing well on higher labelling budgets. Both RAND and MARGIN demonstrate consistent behaviors for the different classifiers, while CLAIRE demonstrates gradual and reliable progress for all labelling budgets. As for the stability, we can see that CLAIRE demonstrates competitive standard deviations ranging from 0.05 to 0.15, comparable to RAND and MARGIN, presenting stable performance across different classifiers. However, USAP shows higher variability at larger budgets (SD up to 0.18 at 250 instances), suggesting less consistent behaviors.

Table 8. Performance Comparison on *NB-V1* Dataset (Averaged Across Classifiers).

Metric	Method	Labeled Instances						
		30	**50**	**100**	**150**	**200**	**250**	**300**
Macro-F	BE	0.43 ± 0.04	0.47 ± 0.07	0.61 ± 0.09	0.65 ± 0.08	0.73 ± 0.03	0.82 ± 0.09	0.81 ± 0.12
	LHCEIII	0.50 ± 0.06	0.59 ± 0.06	0.74 ± 0.08	0.80 ± 0.07	0.81 ± 0.05	0.80 ± 0.05	0.83 ± 0.06
	MARGIN	0.47 ± 0.04	**0.84** ± 0.05	0.88 ± 0.03	0.85 ± 0.07	0.90 ± 0.05	**0.91** ± 0.05	0.90 ± 0.04
	RAND	0.57 ± 0.05	0.82 ± 0.06	**0.89** ± 0.04	**0.91** ± 0.05	**0.91** ± 0.03	0.90 ± 0.06	**0.92** ± 0.06
	USAP	**0.63** ± 0.07	0.60 ± 0.08	0.65 ± 0.10	0.65 ± 0.06	0.69 ± 0.05	0.72 ± 0.18	0.64 ± 0.10
	CLAIRE	0.59 ± 0.07	0.60 ± 0.10	0.68 ± 0.15	0.82 ± 0.13	0.83 ± 0.13	0.85 ± 0.12	0.89 ± 0.08
Macro-Recall	BE	0.53 ± 0.06	0.57 ± 0.07	0.67 ± 0.07	0.70 ± 0.06	0.76 ± 0.03	0.84 ± 0.06	0.84 ± 0.08
	LHCEIII	0.56 ± 0.05	0.63 ± 0.06	0.74 ± 0.07	0.81 ± 0.05	0.83 ± 0.04	0.82 ± 0.04	0.83 ± 0.04
	MARGIN	0.56 ± 0.05	**0.84** ± 0.04	0.89 ± 0.03	0.83 ± 0.07	0.90 ± 0.04	**0.91** ± 0.04	0.91 ± 0.04
	RAND	0.59 ± 0.05	0.81 ± 0.05	**0.89** ± 0.03	**0.91** ± 0.04	**0.91** ± 0.03	0.91 ± 0.04	**0.92** ± 0.05
	USAP	0.66 ± 0.06	0.60 ± 0.08	0.67 ± 0.08	0.66 ± 0.06	0.68 ± 0.05	0.73 ± 0.16	0.64 ± 0.09
	CLAIRE	**0.66** ± 0.05	0.65 ± 0.11	0.74 ± 0.10	0.84 ± 0.10	0.84 ± 0.10	0.86 ± 0.09	0.89 ± 0.06
Macro-Prec.	BE	0.41 ± 0.03	0.53 ± 0.12	0.71 ± 0.10	0.74 ± 0.10	0.83 ± 0.06	0.84 ± 0.09	0.84 ± 0.12
	LHCEIII	0.57 ± 0.11	0.63 ± 0.08	0.76 ± 0.09	0.82 ± 0.06	0.81 ± 0.06	0.83 ± 0.05	0.83 ± 0.07
	MARGIN	0.43 ± 0.03	0.86 ± 0.06	0.88 ± 0.05	**0.92** ± 0.05	0.90 ± 0.06	**0.91** ± 0.06	0.90 ± 0.05
	RAND	0.64 ± 0.02	**0.89** ± 0.05	**0.89** ± 0.05	0.92 ± 0.06	**0.92** ± 0.03	0.90 ± 0.08	**0.92** ± 0.07
	USAP	0.65 ± 0.08	0.64 ± 0.07	0.80 ± 0.10	0.73 ± 0.09	0.77 ± 0.07	0.80 ± 0.11	0.75 ± 0.11
	CLAIRE	**0.72** ± 0.03	0.64 ± 0.08	0.75 ± 0.13	0.84 ± 0.13	0.84 ± 0.14	0.87 ± 0.12	0.90 ± 0.08

± standard deviation across classifiers. **Bold values** highlight the highest metric score achieved by a competing method for each labeling budget size.

For the second variant, NB-V2, the data distribution becomes highly uneven, showing a clear case of class imbalance. Almost 90% of all samples in the dataset account for Normal and TCP traffic, while combo, junk, and UDP attacks contribute only about 0.19% each, as shown in Table 1. Despite this distribution, CLAIRE demonstrates reliable performance. As presented in Table 9, it can be seen that macro F-measure improves from 0.58 to 0.83, while macro Recall increases in an even manner to 0.88 at 300 as well. This promising result demonstrates that CLAIRE is efficient in learning the representative and informative instances even with highly imbalanced data. On the other hand, we can see most baseline methods produce much lower scores, rarely exceeding a macro F-measure of 0.70. MARGIN

and USAP show moderate improvement at mid-range budgets, but they are not showing stable results at higher budget sizes. BE and RAND remain mostly below 0.60 across all metrics, and LHCEIII shows little variation throughout. With respect to stability, CLAIRE shows relatively higher standard deviations (ranging from 0.06 to 0.20) compared to other methods, reflecting some variability across classifiers. Overall, this trend shows that severe class imbalance can weaken traditional active learning methods, especially in the case of IoT intrusion detection systems where detecting the rare attacks is crucial.

Table 9. Performance Comparison on *NB-V2* Dataset (Averaged Across Classifiers).

Metric	Method	Labeled Instances						
		30	**50**	**100**	**150**	**200**	**250**	**300**
Macro-F	BE	0.34 ± 0.05	0.37 ± 0.07	0.37 ± 0.07	0.37 ± 0.08	0.53 ± 0.08	0.52 ± 0.13	0.50 ± 0.15
	LHCEIII	0.42 ± 0.04	0.41 ± 0.02	0.44 ± 0.02	0.50 ± 0.08	0.44 ± 0.02	0.45 ± 0.02	0.44 ± 0.01
	MARGIN	0.44 ± 0.02	0.42 ± 0.06	0.46 ± 0.01	0.47 ± 0.02	0.56 ± 0.04	0.67 ± 0.04	0.54 ± 0.05
	RAND	0.43 ± 0.02	0.44 ± 0.04	0.46 ± 0.01	0.54 ± 0.06	0.59 ± 0.10	0.47 ± 0.01	0.53 ± 0.06
	USAP	0.44 ± 0.01	0.44 ± 0.02	0.65 ± 0.06	0.59 ± 0.09	0.59 ± 0.02	0.55 ± 0.02	0.58 ± 0.07
	CLAIRE	**0.58** ± 0.18	**0.71** ± 0.11	**0.69** ± 0.20	**0.72** ± 0.20	**0.76** ± 0.17	**0.81** ± 0.15	**0.83** ± 0.13
Macro-Recall	BE	0.42 ± 0.04	0.48 ± 0.02	0.48 ± 0.02	0.48 ± 0.02	0.60 ± 0.01	0.59 ± 0.03	0.58 ± 0.06
	LHCEIII	0.42 ± 0.04	0.41 ± 0.02	0.44 ± 0.02	0.49 ± 0.06	0.45 ± 0.01	0.45 ± 0.01	0.44 ± 0.01
	MARGIN	0.43 ± 0.02	0.41 ± 0.05	0.46 ± 0.01	0.48 ± 0.01	0.55 ± 0.05	0.70 ± 0.05	0.53 ± 0.05
	RAND	0.43 ± 0.02	0.43 ± 0.04	0.47 ± 0.01	0.52 ± 0.05	0.59 ± 0.09	0.47 ± 0.02	0.53 ± 0.06
	USAP	0.43 ± 0.01	0.58 ± 0.02	0.69 ± 0.04	0.61 ± 0.13	0.61 ± 0.02	0.56 ± 0.07	0.56 ± 0.05
	CLAIRE	**0.61** ± 0.17	**0.72** ± 0.06	**0.79** ± 0.09	**0.82** ± 0.08	**0.86** ± 0.06	**0.87** ± 0.08	**0.88** ± 0.06
Macro-Prec.	BE	0.33 ± 0.06	0.44 ± 0.15	0.44 ± 0.15	0.44 ± 0.15	0.53 ± 0.10	0.53 ± 0.14	0.53 ± 0.14
	LHCEIII	0.44 ± 0.02	0.42 ± 0.03	0.44 ± 0.03	0.56 ± 0.09	0.44 ± 0.03	0.45 ± 0.02	0.44 ± 0.03
	MARGIN	0.45 ± 0.03	0.44 ± 0.06	0.47 ± 0.01	0.47 ± 0.02	0.62 ± 0.03	0.69 ± 0.05	0.64 ± 0.11
	RAND	0.43 ± 0.03	0.57 ± 0.09	0.46 ± 0.02	0.60 ± 0.08	0.71 ± 0.13	0.48 ± 0.01	0.58 ± 0.08
	USAP	0.46 ± 0.02	0.45 ± 0.01	0.65 ± 0.06	0.69 ± 0.02	0.57 ± 0.03	0.61 ± 0.04	0.68 ± 0.08
	CLAIRE	**0.72** ± 0.17	**0.84** ± 0.02	**0.76** ± 0.16	**0.75** ± 0.19	**0.79** ± 0.18	**0.82** ± 0.15	**0.86** ± 0.12

± standard deviation across classifiers. **Bold values** highlight the highest metric score achieved by a competing method for each labeling budget size.

Table 10. Performance Comparison on *NB-V3* Dataset (Averaged Across Classifiers).

Metric	Method	Labeled Instances						
		30	**50**	**100**	**150**	**200**	**250**	**300**
Macro-F	BE	0.43 ± 0.04	0.46 ± 0.08	0.51 ± 0.04	0.52 ± 0.04	0.65 ± 0.06	0.67 ± 0.05	0.72 ± 0.05
	LHCEIII	0.48 ± 0.07	0.48 ± 0.07	0.55 ± 0.02	0.59 ± 0.05	0.66 ± 0.06	0.69 ± 0.06	0.84 ± 0.12
	MARGIN	0.49 ± 0.07	0.48 ± 0.08	0.50 ± 0.06	0.48 ± 0.04	0.57 ± 0.06	0.66 ± 0.11	0.73 ± 0.09
	RAND	0.42 ± 0.04	0.44 ± 0.02	0.51 ± 0.07	0.52 ± 0.07	0.48 ± 0.04	0.46 ± 0.04	0.67 ± 0.13
	USAP	0.52 ± 0.05	0.44 ± 0.03	0.46 ± 0.03	0.49 ± 0.06	0.55 ± 0.03	0.42 ± 0.07	0.46 ± 0.03
	CLAIRE	**0.58** ± 0.06	**0.73** ± 0.09	**0.80** ± 0.14	**0.88** ± 0.12	**0.91** ± 0.08	**0.91** ± 0.08	**0.93** ± 0.07
Macro-Recall	BE	0.43 ± 0.02	0.45 ± 0.06	0.49 ± 0.04	0.50 ± 0.04	0.62 ± 0.07	0.65 ± 0.05	0.70 ± 0.07
	LHCEIII	0.49 ± 0.09	0.50 ± 0.09	0.55 ± 0.05	0.60 ± 0.04	0.65 ± 0.06	0.70 ± 0.02	0.88 ± 0.07
	MARGIN	0.48 ± 0.06	0.48 ± 0.08	0.51 ± 0.07	0.48 ± 0.04	0.56 ± 0.07	0.63 ± 0.10	0.72 ± 0.07
	RAND	0.43 ± 0.03	0.45 ± 0.02	0.51 ± 0.06	0.51 ± 0.06	0.47 ± 0.04	0.47 ± 0.04	0.64 ± 0.13
	USAP	0.55 ± 0.04	0.44 ± 0.03	0.46 ± 0.02	0.58 ± 0.06	0.57 ± 0.02	0.43 ± 0.07	0.46 ± 0.03
	CLAIRE	**0.61** ± 0.05	**0.73** ± 0.07	**0.82** ± 0.10	**0.88** ± 0.11	**0.91** ± 0.08	**0.90** ± 0.09	**0.92** ± 0.09
Macro-Prec.	BE	0.55 ± 0.09	0.55 ± 0.12	0.61 ± 0.03	0.62 ± 0.02	0.77 ± 0.04	0.77 ± 0.03	0.90 ± 0.08
	LHCEIII	0.53 ± 0.09	0.54 ± 0.08	0.59 ± 0.05	0.59 ± 0.05	0.81 ± 0.10	0.80 ± 0.10	0.85 ± 0.12
	MARGIN	0.55 ± 0.07	0.52 ± 0.07	0.56 ± 0.08	0.58 ± 0.08	0.60 ± 0.05	0.83 ± 0.14	0.83 ± 0.08
	RAND	0.42 ± 0.04	0.44 ± 0.02	0.56 ± 0.07	0.59 ± 0.04	0.58 ± 0.07	0.46 ± 0.04	0.85 ± 0.14
	USAP	0.51 ± 0.07	0.44 ± 0.02	0.46 ± 0.02	0.53 ± 0.07	0.56 ± 0.07	0.42 ± 0.07	0.46 ± 0.03
	CLAIRE	**0.67** ± 0.13	**0.76** ± 0.12	**0.84** ± 0.13	**0.91** ± 0.11	**0.92** ± 0.08	**0.94** ± 0.06	**0.95** ± 0.05

± standard deviation across classifiers. **Bold values** highlight the highest metric score achieved by a competing method for each labeling budget size.

Similar to NB-V2, NB-V3 represents very imbalanced data but with different types of behaviors. As can be seen in Table 1, attack traffic represents the largest portion of the dataset compared to few normal types. The combo and junk categories jointly comprise more than 80% of all the samples. TCP and scan traffic appear only occasionally. Despite this asymmetrical distribution, CLAIRE performs reliably and improves as more labeled data become available. Its macro F-measure increases from 0.58 to 0.93, macro Recall rises from 0.61 to 0.92, and macro Precision reaches 0.95, as shown in Table 10. These results show that CLAIRE adjusts gracefully even when most of the data are concentrated in a few dominant attack types. Contrarily, the baseline approaches struggle to match this level of consistency. At a labelling budget of 300, LHCEIII reaches a macro F-measure of 0.84, followed by BE at approximately 0.72. MARGIN and RAND perform slightly lower, averaging around 0.73 and 0.67, respectively. USAP performs below average across all budget levels, reaching only up to 0.46. These results show the adaptive effectiveness of CLAIRE in case of pronounced class imbalance where it consistently maintains both precision and recall, even when the dataset is dominated by a few prevalent attack types. This balanced performance is highly valuable in real-world network environments where missing rare but critical threats can have serious consequences.

Overall, for every variant of the CIC and NB datasets, CLAIRE maintains consistently high performance compared to baseline methods. The results highlight the capability of CLAIRE to learn effective decision boundaries by identifying and leveraging the most informative and representative instances from the training dataset, thereby demonstrating the robustness of active learning with highly imbalanced data. By contrast, the baseline methods exhibit marked volatility, finding it challenging to learn informative and representative instances in such imbalanced scenarios.

Computational Efficiency

In this section, we demonstrate the computational cost of the proposed approach. We only focus on the labelling process, while the time of classification and monitoring phases are considered to be the same as they only work on the same number of instances. As depicted in Figure 2, CLAIRE consists of four layers. Layer 1 adapts the VWC algorithm to partition unlabeled data into hundreds of micro-clusters. This layer performs a single scan to cluster the data based on a fixed width learned from the data. Any cluster exceeding the maximum cluster size s_{max} is recursively partitioned using a new fixed width learned from that cluster. This recursive process continues until no clusters exceed s_{max}. As shown in Table 11, smaller values of s_{max} increase the total execution time. Layer 2 applies EM clustering with complexity O($K \cdot d \cdot C \cdot I$). However, as illustrated in Figure 6, the computational cost of Layer 2 (L2) remains manageable because this layer operates only on the medoids produced by Layer 1, not on the original large dataset. Layer 3 requires similar time. Layer 4 also operates on very few instances and shows very small time. Overall, as shown in Figure 6, the proposed approach demonstrates large computational time compared to the baseline methods. For example, on the CIC-IoT dataset it takes about 189 s, while the worst and best baseline methods take approximately 118 and 2 s, respectively. In fact, RAND demonstrates the fastest one and this is because it has no intelligent learning part and demonstrates poor results in imbalanced classes (see Section 5.1). Similarly, all methods have the same behaviors and order of computational time on a much larger dataset named N-BaIoT. As can be seen, most of the time for the proposed approach is consumed in the first layer, which involves micro-clustering technique and medoid calculations for each cluster, which consumed a substantial part of the time. Table 11 shows that VWC with s_{max} = 1000 only takes about 121 s. This means most of the time in Layer 1, which is 451 s, is

consumed by medoid calculation. However, this time can be much reduced as each cluster can be treated independently in a parallelized fashion.

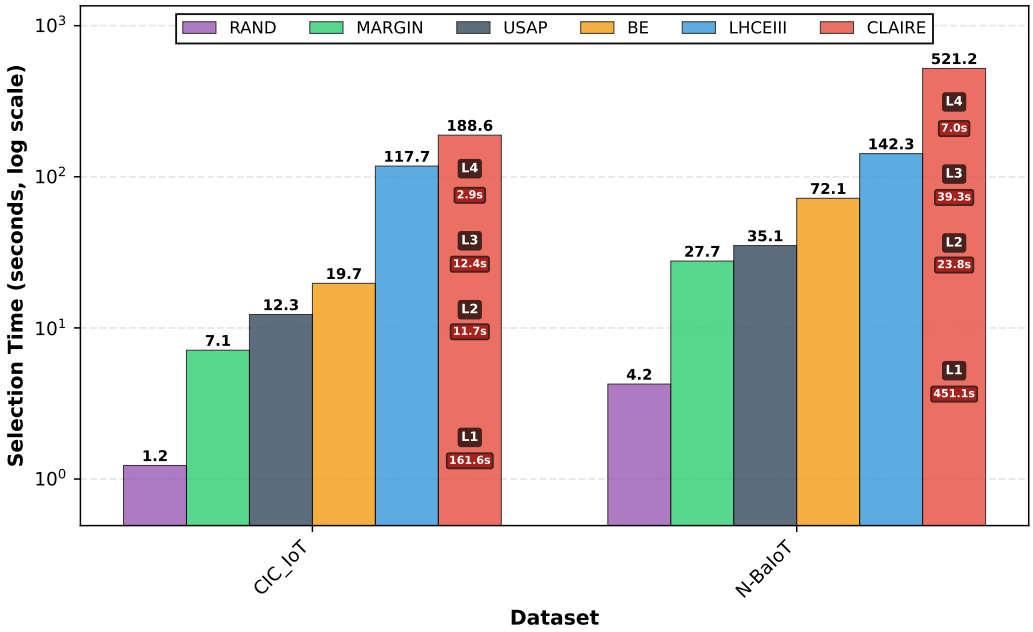

Figure 6. Selection time comparison for six active learning methods generating 300 representative instances from two IoT security datasets (N-BaIoT and CIC-IoT).

Table 11. Clustering Construction Time Comparison between K-means and VWC.

	K-Means				**VWC**					
	$k = 100$	$k = 300$	$k = 700$	$k = 1000$	$s_{max} = 100$	$s_{max} = 500$	$s_{max} = 1000$	$s_{max} = 5000$	$s_{max} = 7000$	$s_{max} = 10{,}000$
Clusters	100	300	700	999	52,947	15,622	9200	2874	2317	1809
Time (s)	443.6	2453.9	4597.0	10,600.4	231.6	145.5	120.9	86.8	85.1	77.4

6. Conclusions

In this paper, we have presented a four-layer active learning approach called CLAIRE that effectively learns the most representative and informative instances from training data, leading to the development of an efficient IoT-based IDS. The experimental results validate the effectiveness of CLAIRE, demonstrating promising performance over established baselines across multiple evaluation scenarios. In particular, CLAIRE excels at exploring and learning informative and representative instances that capture most class labels (behaviors) even in unlabelled imbalanced datasets. Additionally, the learned labeled data demonstrate high quality when used in the classification learning process. While the computation time of the proposed approach is a limitation, it is designed to operate offline. Importantly, the framework's logic can be parallelized to significantly reduce this computational overhead.

Several directions for future research are planned. First, we aim to extend the framework to handle streaming data scenarios common in IoT environments, including the implementation of concept drift handling to maintain detection accuracy in dynamic environments with evolving attack patterns. Second, we will investigate federated learning integration for collaborative security across IoT networks while preserving privacy. Third, we will explore adaptation to emerging IoT protocols and device types.

Funding: The project was funded by KAU Endowment (WAQF) at king Abdulaziz University, Jeddah, Saudi Arabia.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The project was funded by KAU Endowment (WAQF) at king Abdulaziz University, Jeddah, Saudi Arabia. The author, therefore, acknowledge with thanks WAQF and the Deanship of Scientific Research (DSR) for technical and financial support.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Nguyen, D.C.; Ding, M.; Pathirana, P.N.; Seneviratne, A.; Li, J.; Niyato, D.; Dobre, O.; Poor, H.V. 6G Internet of Things: A Comprehensive Survey. *IEEE Internet Things J.* **2021**, *9*, 359–383. [CrossRef]
2. Kumar, S.; Tiwari, P.; Zymbler, M. Internet of Things is a Revolutionary Approach for Future Technology Enhancement: A Review. *J. Big Data* **2019**, *6*, 111. [CrossRef]
3. Transforma Insights. Global IoT Forecast Highlights (2023–2033). 2025. Available online: https://transformainsights.com/research/forecast/highlights (accessed on 24 May 2025).
4. Cisco Systems. Cisco Annual Internet Report (2018–2023). 2023. Available online: https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internet-report/index.html (accessed on 22 May 2025).
5. Saheed, Y.K.; Abdulganiyu, O.H.; Tchakoucht, T.A. A Novel Hybrid Ensemble Learning for Anomaly Detection in Industrial Sensor Networks and SCADA Systems for Smart City Infrastructures. *J. King Saud Univ.—Comput. Inf. Sci.* **2023**, *35*, 101532. [CrossRef]
6. Asif, M.; Abbas, S.; Khan, M.A.; Fatima, A.; Khan, M.A.; Lee, S.W. MapReduce-Based Intelligent Model for Intrusion Detection Using Machine Learning Technique. *J. King Saud Univ.—Comput. Inf. Sci.* **2022**, *34*, 9723–9731. [CrossRef]
7. Zarpelão, B.B.; Miani, R.S.; Kawakani, C.T.; Alvarenga, S.C.D. A survey of intrusion detection in Internet of Things. *J. Netw. Comput. Appl.* **2017**, *84*, 25–37. [CrossRef]
8. Jamalipour, A.; Murali, S. A taxonomy of machine-learning-based intrusion detection systems for the internet of things: A survey. *IEEE Internet Things J.* **2021**, *9*, 9444–9466. [CrossRef]
9. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 20. [CrossRef]
10. Abbas, S.; Alsubai, S.; Haque, M.I.U.; Sampedro, G.A.; Almadhor, A.; Al Hejaili, A.; Ivanochko, I. Active machine learning for heterogeneity activity recognition through smartwatch sensors. *IEEE Access* **2024**, *12*, 22595–22607. [CrossRef]
11. Mahdi, H.F.; Khadhim, B.J. Enhancing IoT Security: A Deep Learning and Active Learning Approach to Intrusion Detection. *J. Robot. Control. (JRC)* **2024**, *5*, 1525–1535.
12. Zakariah, M.; Almazyad, A.S. Anomaly detection for IoT systems using active learning. *Appl. Sci.* **2023**, *13*, 12029. [CrossRef]
13. Jian, C.; Yang, K.; Ao, Y. Industrial fault diagnosis based on active learning and semi-supervised learning using small training set. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104365. [CrossRef]
14. Almalawi, A.M.; Fahad, A.; Tari, Z.; Cheema, M.A.; Khalil, I. *k*-NNVWC: An Efficient *k*-Nearest Neighbors Approach Based on Various-Widths Clustering. *IEEE Trans. Knowl. Data Eng.* **2015**, *28*, 68–81. [CrossRef]
15. Tharwat, A.; Schenck, W. A Survey on Active Learning: State-of-the-Art, Practical Challenges and Research Directions. *Mathematics* **2023**, *11*, 820. [CrossRef]
16. Settles, B. *Active Learning Literature Survey*; Computer Sciences Technical Report 1648; University of Wisconsin-Madison, Department of Computer Sciences: Madison, WI, USA, 2009.
17. Angluin, D. Queries and Concept Learning. *Mach. Learn.* **1988**, *2*, 319–342. [CrossRef]
18. Cohn, D.A.; Atlas, L.E.; Ladner, R.E. Improving Generalization with Active Learning. *Mach. Learn.* **1994**, *15*, 201–221. [CrossRef]
19. Lewis, D.D. A Sequential Algorithm for Training Text Classifiers: Corrigendum and Additional Data. In *Proceedings of the ACM SIGIR Forum*; ACM: New York, NY, USA, 1995; Volume 29, pp. 13–19.
20. Tharwat, A.; Schenck, W. Balancing Exploration and Exploitation: A Novel Active Learner for Imbalanced Data. *Knowl.-Based Syst.* **2020**, *210*, 106500. [CrossRef]
21. Wang, H.; Chang, X.; Shi, L.; Yang, Y.; Shen, Y.D. Uncertainty Sampling for Action Recognition via Maximizing Expected Average Precision. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18). International Joint Conferences on Artificial Intelligence Organization, Stockholm, Sweden, 13–19 July 2018; pp. 964–970. [CrossRef]
22. Lu, J.; Zhao, P.; Hoi, S.C. Online Passive-Aggressive Active Learning. *Mach. Learn.* **2016**, *103*, 141–183. [CrossRef]

23. Qin, J.; Wang, C.; Zou, Q.; Sun, Y.; Chen, B. Active learning with extreme learning machine for online imbalanced multiclass classification. *Knowl.-Based Syst.* **2021**, *231*, 107385. [CrossRef]
24. Cacciarelli, D.; Kulahci, M.; Tyssedal, J. Online active learning for soft sensor development using semi-supervised autoencoders. *arXiv* **2022**, arXiv:2212.13067.
25. Huang, S.J.; Jin, R.; Zhou, Z.H. Active Learning by Querying Informative and Representative Examples. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1936–1949. [CrossRef]
26. Ash, J.T.; Zhang, C.; Krishnamurthy, A.; Langford, J.; Agarwal, A. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv* **2019**, arXiv:1906.03671.
27. Flesca, S.; Mandaglio, D.; Scala, F.; Tagarelli, A. A meta-active learning approach exploiting instance importance. *Expert Syst. Appl.* **2024**, *247*, 123320. [CrossRef]
28. Soltani, N.; Zhang, J.; Salehi, B.; Roy, D.; Nowak, R.; Chowdhury, K. Learning from the Best: Active Learning for Wireless Communications. *IEEE Wirel. Commun.* 2024, *early access*. [CrossRef]
29. Halder, B.; Hasan, K.M.A.; Amagasa, T.; Ahmed, M.M. Autonomic active learning strategy using cluster-based ensemble classifier for concept drifts in imbalanced data stream. *Expert Syst. Appl.* **2023**, *231*, 120578. [CrossRef]
30. Meidan, Y.; Bohadana, M.; Mathov, Y.; Mirsky, Y.; Breitenbacher, D.; Shabtai, A. N-BaIoT Dataset: Detection of IoT Botnet Attacks. 2018. Available online: https://archive.ics.uci.edu/dataset/442/detection+of+iot+botnet+attacks+n+baiot (accessed on 25 May 2025).
31. Neto, E.C.P.; Dadkhah, S.; Ferreira, R.; Zohourian, A.; Lu, R.; Ghorbani, A.A. CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors* **2023**, *23*, 5941. [CrossRef]
32. Tong, S.; Koller, D. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.* **2001**, *2*, 45–66.
33. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2014.
34. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Prentice Hall: Hoboken, NJ, USA, 2010.
35. Guo, G.; Wang, H.; Bell, D.; Bi, Y.; Greer, K. KNN model-based approach in classification. In Proceedings of the On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, 3–7 November 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 986–996.
36. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
37. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]