GAIS: A Novel Approach to Instance Selection with Graph Attention Networks

Zahiriddin Rustamov [1,2], Ayham Zaitouny[3], Rafat Damseh[1], and Nazar Zaki[1,*]

[1]College of Information Technology, United Arab Emirates University, Al Ain, UAE
[2]Department of Computer Science, KU Leuven
[3]College of Science, United Arab Emirates University, Al Ain, UAE
nzaki@uaeu.ac.ae

Abstract—**Instance selection (IS) is a crucial technique in machine learning that aims to reduce dataset size while maintaining model performance. This paper introduces a novel method called Graph Attention-based Instance Selection (GAIS), which leverages Graph Attention Networks (GATs) to identify the most informative instances in a dataset. GAIS represents the data as a graph and uses GATs to learn node representations, enabling it to capture complex relationships between instances. The method processes data in chunks, applies random masking and similarity thresholding during graph construction, and selects instances based on confidence scores from the trained GAT model. Experiments on 13 diverse datasets demonstrate that GAIS consistently outperforms traditional IS methods in terms of effectiveness, achieving high reduction rates (average 96%) while maintaining or improving model performance. Although GAIS exhibits slightly higher computational costs, its superior performance in maintaining accuracy with significantly reduced training data makes it a promising approach for graph-based data selection. Code is available at https://github.com/zahiriddin-rustamov/gais.**

Index Terms—**instance selection, graph attention networks, machine learning, data reduction**

I. INTRODUCTION

In the rapidly evolving landscape of machine learning (ML), the drive towards efficiency is not just a wish but also a necessity, particularly in the context of big data, ML model deployment on edge devices, and the emphasis on the potential of TinyML [1]. While big data offers rich datasets leading to more accurate models, it also brings high computational demands, longer training times, and increased costs. These challenges are significant in fields requiring rapid model deployment, such as healthcare, finance, and autonomous systems, and in deploying ML models on resource-constrained edge devices. Moreover, many domains struggle with big data due to the costly and time-intensive nature of the data acquisition processes [2].

One way to tackle these issues is through instance selection (IS), which involves curating training datasets to include only the most informative instances. IS offers dual benefits: reducing computational demands and potentially enhancing model performance by eliminating redundant or noisy data. This is especially crucial in developing TinyML models, where the balance between model size, accuracy, and computational efficiency is critical.

Existing IS methods, such as random sampling, prototype-based methods [3], and active learning [4], often fail to capture complex relationships between instances. Hence, we propose a novel IS method using Graph Attention Networks (GATs), which effectively handle graph-structured data by attending to node neighbours. By representing the dataset as a graph and leveraging GATs' attention mechanisms, our method identifies the most informative instances while considering their relationships and overall data structure. The main contributions of this paper are as follows:

- We propose Graph Attention-based Instance Selection (GAIS), which utilizes GATs to identify the most informative instances in the training data based on their local relationships.
- We demonstrate GAIS's effectiveness on benchmark datasets, achieving significant dataset size reductions while maintaining model performance, highlighting its superiority over existing IS methods.

The paper is structured as follows: Section II presents related work; Section III formulates the problem; Section IV details the proposed method; Section V describes the experimental setup; Section VI discusses the results; and Section VII concludes the study.

II. RELATED WORKS

Significant efforts have addressed the challenges of IS and reduction across various domains. Hart's [5] condensed nearest neighbour rule laid the foundation for many subsequent IS algorithms by reducing dataset size while preserving classification integrity. Wilson [6] expanded on this by exploring asymptotic properties of nearest neighbour rules with edited data. Brighton and Mellish [7] advanced IS in instance-based learning, offering insights into optimizing IS algorithms.

A. Traditional IS Methods

Traditional IS methods have been extensively studied and applied in various domains. Zhang et al. [8] tackled large dataset challenges, while Yu et al. [9] improved speech synthesis via prosodic IS. Fu et al. [4] surveyed active learning from an IS perspective. Advancements in Multiple-Instance Learning (MIL) and optimization were contributed by Zhang et al. [8], and Peng et al. [10], introducing IS strategies and

outlier detection to enhance learning model efficiency and accuracy.

The versatility of IS techniques was further explored by Lee et al. [11] and Klinger et al. [12] through evolutionary algorithms based on fuzzy models and cross-lingual training. Lin et al. [13], Stanovov et al. [14], and Bicici [15] expanded IS applications to large-scale data detection, hybrid evolutionary algorithms, and statistical machine translation.

Building on these foundations, Carbonera and Abel introduced the centroid density-based IS algorithm (CDIS) [16], [17], utilizing density measures to enhance IS. Malhat et al. [18] proposed GDIS and EGDIS algorithms, leveraging global density and relevance functions to improve classification accuracy and reduction rates.

The field has continued to evolve, with Ashfaq et al. [19] and Villuendas-Rey et al. [20] introducing fuzziness-based intrusion detection and combined instance and feature selection. More recent studies by Kavrin et al. [21], Yuan et al. [22], and DeVries et al. [23] focus on bagging-based IS, MIL with multiple-point concepts, and IS for generative adversarial networks (GANs), respectively, indicating a trend towards more sophisticated and integrated approaches.

B. Graph-based IS Methods

Graph-based methods have advanced IS in ML by leveraging data structure, representing datasets as graphs with instances as nodes and relationships as edges. By exploiting the graph structure, this approach captures complex patterns and relationships, leading to more effective IS.

Several graph-based IS methods have been proposed. Nikolaidis et al. [24] introduced a spectral graph theory-based instance reduction algorithm. Nguyen et al. [25] focused on clustering visual instances from image collections, and Wang et al. [26] enhanced graph-based semi-supervised learning through IS. Yang et al. [27] developed the Natural Neighborhood Graph-based Instance Reduction (NNGIR), which automatically constructs a natural neighborhood graph to divide the training set into noisy, border, and internal instances, eliminating noisy and redundant points without requiring parameter tuning.

Other notable methods include Gabriel Graph Editing (GGE) and Relative Neighborhood Graph Editing (RNGE) [28], which use geometric criteria to connect close points and retain instances near decision boundaries. Hit Miss Networks (HMNs) [29] represent training sets as graphs where nodes are instances and edges represent nearest neighbor relations, leading to editing algorithms that remove instances based on graph properties. Class Conditional IS [30] builds on HMNs by introducing a scoring function for large margin IS. Recent studies by Nonaka et al. [31] and Lou et al. [32] demonstrate the adaptability of graph-based methods in IS, analyzing deep learning outputs using K-nearest neighbor graphs and adopting heterogeneous graph neural networks, respectively.

Despite advancements, traditional IS methods face challenges with scalability, noisy data, and balancing reduction rate with classification accuracy, especially with complex, high-dimensional data [3], [18]. Standard graph-based methods may still struggle with complex data and often lack the ability to effectively capture and leverage the varying importance of different neighboring instances, highlighting the need for more adaptive approaches.

GATs present a significant advancement over traditional methods [33]. GATs employ a dynamic attention mechanism that adaptively assigns importance to nodes, allowing for more nuanced handling of complex inter-instance relationships. This approach offers increased robustness against noise in the data. The key innovation of GATs lies in their ability to learn and apply varying weights to different neighboring instances, potentially capturing subtle but important patterns in the data structure. This dynamic weighting mechanism could lead to more informed IS decisions, particularly in datasets with complex or hierarchical relationships between data points. Building on these advancements, this paper introduces *GAIS*, a novel IS method that integrates GATs to explore how adaptive attention mechanisms can be applied to the challenge of data reduction in ML.

III. PROBLEM DEFINITION

The IS problem seeks to select a minimal subset $S \subseteq D$ from a dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the feature vector and y_i the corresponding label, such that a ML model trained on S performs comparably to one trained on D. Formally, the objective is:

$$\min_{S \subseteq D} |S|$$

subject to:

$$\text{Accuracy}(M(S)) \geq \text{Accuracy}(M(D)) - \Phi$$

where $|S|$ is the size of S, $M(S)$ is the model trained on S, Accuracy$(M(S))$ its accuracy, and Φ is a tolerance threshold. The goal is to find the smallest S that maintains acceptable performance compared to training on the full dataset D.

IV. GRAPH ATTENTION-BASED INSTANCE SELECTION (GAIS)

The proposed GAIS method aims to efficiently reduces dataset size by preserving representative samples using GATs. It processes data in chunks to select informative instances based on confidence scores. The workflow includes data chunking, graph construction, GAT model training, and IS.

A. Data Chunking

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, GAIS divides it into manageable overlapping chunks $\mathcal{C}_j$ of window size w (e.g., 8,000) with overlap o:

$$\mathcal{C}_j = \{(\mathbf{x}_i, y_i)\}_{i=(j-1)(w-o)+1}^{j(w-o)+o}, \quad j = 1, 2, \ldots, \left\lceil \frac{N - o}{w - o} \right\rceil$$

where $\lceil \cdot \rceil$ denotes the ceiling function. The motivation behind chunking the data is to address computational and memory constraints when constructing graphs from large

datasets. Chunking allows for localized graph construction within each chunk, speeding up computation. It also handles large datasets incrementally, preventing memory overload. Overlapping chunks maintain continuity between instances across boundaries.

Prior to chunking, we randomly shuffle the $\mathcal{D}$ to ensure that the chunking process is not dependent on the original order of the data. This step helps to mitigate any potential bias from inherent data ordering and improves the robustness of our method to data permutations.

B. Graph Construction

For each chunk $\mathcal{C}_j$, GAIS constructs an undirected graph $\mathcal{G}_j = (\mathcal{V}_j, \mathcal{E}_j)$ with edges determined by a chosen distance metric (e.g., Euclidean distance). Edge weights a_{ij} between nodes i and j are computed accordingly. To reduce complexity and focus on significant connections, GAIS applies two thresholds:

- **Random Threshold** (θ_r): Randomly removes a fraction of edges to introduce randomness and prevent overfitting.
- **Similarity Threshold** (θ_s): Removes edges with weights below θ_s, eliminating weak connections.

The resulting adjacency matrix $\mathbf{A}_j$ is defined as:

$$\mathbf{A}_j[i,j] = \begin{cases} a_{ij}, & \text{if } a_{ij} \geq \theta_s \text{ and } (i,j) \notin \mathcal{R}_j \\ 0, & \text{otherwise} \end{cases}$$

where $\mathcal{R}_j$ is the set of randomly removed edges based on θ_r.

C. GAT Model Architecture

GAIS employs a GAT as its core component to assign attention weights to edges, capturing the importance of each neighbor. The GAT architecture consists of input, hidden, and output layers. A single GAT layer updates node features as:

$$\mathbf{h}'_i = \text{ELU}\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{W} \mathbf{h}_j \right)$$

where $\mathbf{h}'_i$ and $\mathbf{h}_i$ are the output and input features of node i, $\mathcal{N}(i)$ denotes its neighbors, $\mathbf{W}$ is a learnable weight matrix, and α_{ij} is the attention coefficient:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_j]\right)\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^\top [\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_k]\right)\right)}$$

with $\mathbf{a}$ as a learnable vector and $\|$ denoting concatenation.

Stacking multiple GAT layers captures higher-order relationships. The GAIS model architecture is defined as:

$$\mathbf{Z} = \text{GATLayer}_2\left(\text{ELU}\left(\text{GATLayer}_1(\mathbf{X}, \mathbf{A})\right), \mathbf{A}\right)$$

where $\mathbf{Z}$ is the output feature matrix, $\mathbf{X}$ the input features, and $\mathbf{A}$ the adjacency matrix. The final output is passed through a softmax function to obtain node representations $\mathbf{z}_i$ for IS.

Node representations are updated by aggregating weighted neighbor features:

$$\mathbf{h}_i^{(l)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} \mathbf{W}^{(l)} \mathbf{h}_j^{(l-1)} \right)$$

where σ is an activation function like ReLU or ELU. This architecture effectively weighs neighbor influences, transforming input features into intermediate representations and output probabilities.

D. Sequential Training

GAIS trains the GAT model sequentially on data chunks $\mathcal{C}_j$, initializing weights randomly and updating iteratively per chunk. This approach handles large datasets by focusing on subsets, reducing memory requirements and enabling efficient learning.

The training objective minimizes the weighted negative log-likelihood loss:

$$\mathcal{L}_{\text{w}} = \frac{1}{|\mathcal{V}_j|} \sum_{i=1}^{|\mathcal{V}_j|} \left(-\sum_{c=1}^{C} y_{ic} \log(\hat{y}_{ic}) \right) s_i$$

where C is the number of classes, y_{ic} is the true label, $\hat{y}_{ic}$ is the predicted probability (from softmax of $\mathbf{z}_i$), and s_i is the confidence score for node i. This weighting focuses the model on instances with higher confidence, enhancing learning and performance.

E. Instance Selection

After training on all data chunks, GAIS performs IS by reprocessing each chunk $\mathcal{C}_j$. For each instance i in $\mathcal{C}_j$, the model computes a confidence score s_i:

$$s_i = \max_c \hat{y}_{ic}$$

Instances with confidence scores above a predefined threshold θ_c are considered informative and included in the final dataset $\mathcal{D}_{\text{GAIS}}$:

$$\mathcal{D}_{\text{GAIS}} = \bigcup_j \{(\mathbf{x}_i, y_i) \in \mathcal{C}_j \mid s_i \geq \theta_c\}$$

This step allows GAIS to identify key instances that capture the dataset's essential characteristics, reducing size while preserving informative samples.

F. Hyperparameter Tuning Strategy

We employ Bayesian optimization via the Optuna framework [34] to tune the GAT's hyperparameters $\mathbf{H}^*$, maximizing performance metric E on a validation set. The hyperparameters include hidden dimension h, number of input and output heads n_{in}, n_{out}, dropout rate δ, distance metric d and thresholds (θ_r), (θ_s), and (θ_c) for random, similarity, and confidence, respectively.

Bayesian optimization involves modeling the objective function $f(\mathbf{H})$, representing the model's performance as a function of hyperparameters, with a surrogate probabilistic model (e.g., Gaussian Process). An acquisition function $a(\mathbf{H}; \mathcal{D})$, such as Expected Improvement (EI) or Upper Confidence Bound (UCB), guides the selection of the next hyperparameters by balancing exploration and exploitation:

$$\text{EI}(\mathbf{H}) = \mathbb{E}\left[\max(f(\mathbf{H}) - f(\mathbf{H}^+), 0)\right],$$

$$\text{UCB}(\mathbf{H}) = \mu(\mathbf{H}) + \kappa \sigma(\mathbf{H})$$

TABLE I: Datasets Overview. N, P, and C represent the number of instances, features, and classes, respectively.

Dataset Name	Domains	Objective	N	P	C
Heart Disease	Healthcare	Predict heart disease presence	303	14	2
HCC	Oncology	Identify Hepatocellular Carcinoma cases	615	13	2
Stroke Prediction	Healthcare	Predict stroke likelihood	749	11	2
Diabetes	Healthcare	Predict diabetes onset	768	9	2
German Credit	Financial Risk	Classify credit risks	1,000	20	2
Chess End-Game	Game Theory	Predict game outcome	3,196	36	2
Spam-base	Email Classification	Classify emails as spam or non-spam	4,601	57	2
Two-norm	Synthetic Data	Distinguish between two classes	7,400	20	2
Ring-norm	Synthetic Data	Classify into one of the rings	7,400	20	2
Coil-2000	Image Recognition	Object recognition and classification	9,822	85	2
Nursery	Social Science	Rank applications for nursery schools	12,960	8	2
Magic	Physics & Chemistry	Classify gamma-ray and hadron events	19,020	10	2
Fars	Traffic Safety	Classify injury levels suffered in car accidents	100,968	29	9

where $\mathbf{H}^+$ is the current best hyperparameter set, κ is a parameter, and $\sigma(\mathbf{H})$ is the standard deviation at $\mathbf{H}$. The next hyperparameters are chosen by maximizing the acquisition function:

$$\mathbf{H}_{\text{next}} = \arg\max_{\mathbf{H}} a(\mathbf{H}; \mathcal{D})$$

This iterative process updates the surrogate model with new evaluations until convergence, ensuring the GAT model is optimally tuned for each dataset.

V. EXPERIMENTAL SETUP

This section details the experimental framework used to evaluate the GAIS method.

A. Datasets

In this study, we used a diverse set of well-known datasets to evaluate the effectiveness of GAIS, summarised in Table I. The datasets cover domains such as healthcare, financial risk assessment, game theory, oncology, and image recognition. This diversity tests the robustness and applicability of our method across different scenarios.

The datasets are split into training (80%), validation (10%), and testing (10%) subsets. The training set was used to apply the proposed GAIS method for IS; the validation set optimized hyperparameters for GAIS and GAT; the testing set evaluated ML model performance on reduced datasets. We controlled randomness in splitting for reproducibility. All features were retained; categorical features were label encoded, numerical features scaled to [0,1], and target columns standardized for consistency.

B. Graph Construction

To apply GAIS, we converted datasets into graph representations using three distance metrics: Manhattan, Euclidean, and cosine similarity, capturing different aspects of instance similarity. For a dataset X with n instances, we computed pairwise distances $D \in \mathbb{R}^{n \times n}$. We transformed distances into similarity scores using:

$$S_{ij} = \begin{cases} 1 - D_{ij}, & \text{for cosine similarity} \\ 1/(1 + D_{ij}), & \text{for Manhattan and Euclidean} \end{cases}$$

A similarity threshold (θ_s) was applied to create edges only between instances with similarity above this threshold, ensuring connections among the most similar instances.

C. Machine Learning Models

We evaluated GAIS using five ML models: naive bayes (NB), random forest (RF), logistic regression (LR), eXtreme gradient boosting (XGB), and k-nearest neighbours (KNN), chosen for their diverse learning paradigms and ability to handle various data types and complexities.

D. Evaluation Metrics

We assess the performance of the GAIS method using a range of standard evaluation metrics. Accuracy (AC) measures the proportion of correctly classified instances out of the total instances, calculated as:

$$AC = (TP + TN)/(TP + TN + FP + FN)$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. Precision (PR) represents the proportion of true positive instances among the instances predicted as positive:

$$PR = TP/(TP + FP)$$

Recall (RE) is the proportion of true positive instances among the actual positive instances:

$$RE = TP/(TP + FN)$$

F1-score (F1) is the harmonic mean of precision and recall, providing a balanced measure of a model's performance, calculated as:

$$F1 = 2 \times (PR \times RE)/(PR + RE)$$

To evaluate the balance between dataset reduction and accuracy, we employ the Effectiveness metric [18]. The Reduction Rate (R) quantifies the proportion of instances removed:

$$R = 1 - |S|/|D|$$

where $|S|$ and $|D|$ are the sizes of the reduced and the original dataset, respectively. The Effectiveness (E) is calculated as:

$$E = AC \times R$$

TABLE II: Performance Metrics Over 13 Datasets Based on Reduced and Full Training Data.

Dataset	TR_A	Reduced Data				Original Data				R	E
		AC	PR	RE	$F1$	AC	PR	RE	$F1$		
Heart Disease	244	0.9355	1.0000	0.8824	0.9375	0.9032	0.9375	0.8824	0.9091	0.9549	0.8933
HCC	497	0.9516	1.0000	0.6250	0.7692	0.9839	0.8889	1.0000	0.9412	0.9799	0.9324
Stroke Prediction	606	0.8133	0.7200	0.7200	0.7200	0.7733	0.6538	0.6800	0.6667	0.9884	0.8862
Diabetes	621	0.7273	0.6250	0.5556	0.5882	0.7013	0.6250	0.3704	0.4651	0.9871	0.7179
German Credit	810	0.8000	0.7778	0.4667	0.5833	0.7600	0.6000	0.6000	0.6000	0.9580	0.7664
Chess End-Game	2588	0.9469	0.9573	0.9401	0.9486	0.9906	0.9824	1.0000	0.9911	0.9629	0.9118
Spam-base	3726	0.8980	0.9592	0.7747	0.8571	0.9501	0.9492	0.9231	0.9359	0.9868	0.9655
Ring-norm	5994	0.9828	0.9661	0.9893	0.9775	0.9770	0.9661	0.9893	0.9775	0.9828	0.9655
Two-norm	5994	0.9730	0.9534	0.9946	0.9735	0.9703	0.9780	0.9622	0.9700	0.9863	0.8039
Coil-2000	7955	0.9156	0.1667	0.1017	0.1263	0.9390	0.0000	0.0000	0.0000	0.9901	0.9064
Nursery	10495	0.9460	0.9248	0.8533	0.8804	0.9992	0.9994	0.9994	0.9994	0.9334	0.8830
Magic	15406	0.8617	0.8652	0.9319	0.8973	0.8906	0.8891	0.9497	0.9184	0.9627	0.8296
Fars	81783	0.7784	0.5371	0.5113	0.5182	0.8015	0.6043	0.5552	0.5606	0.9278	0.7221
Average		**0.8869**	**0.8046**	**0.7192**	**0.7525**	**0.8954**	**0.7749**	**0.7624**	**0.7642**	**0.9693**	**0.8599**

We also measure the total computation time (T_{IS}) of the GAIS method, including preprocessing, GAT training, thresholding, and ML model training.

Experiments were conducted on a Windows 10 system with an Intel Core i7-12700 CPU (12 cores, 20 threads), 128 GB RAM, and an NVIDIA GeForce RTX 4090 GPU (with 24 GB memory). GAT was implemented using PyTorch 2.1.2 and PyTorch Geometric 2.4.0.

VI. RESULTS & DISCUSSIONS

This section presents the experimental results obtained by evaluating the performance of the proposed GAIS method on various datasets. Table II summarizes the results, comparing the performance metrics achieved on the reduced datasets obtained using GAIS with those obtained on the original datasets. The table includes metrics such as the number of original training instances (TR_A), accuracy (AC), precision (PR), recall (RE), F1 score ($F1$), reduction rate (R), and effectiveness (E). From the results presented, several insights can be drawn:

- The proposed GAIS method achieves high reduction rates (R) across all datasets, with an average reduction rate of 0.9693. This indicates that GAIS effectively reduces the size of the datasets while maintaining comparable performance to the original datasets.
- The accuracy (AC) obtained on the reduced datasets is close to that of the original datasets, with an average difference of only 0.0085. This suggests that GAIS is able to select informative instances that contribute to the learning process.
- The effectiveness (E) metric, which combines accuracy and reduction rate, shows that GAIS achieves a good balance between dataset reduction and performance, with an average effectiveness of 0.8599.
- While the precision (PR), recall (RE), and F1 score ($F1$) vary across datasets, the average values for these metrics on the reduced datasets are comparable to those obtained on the original datasets, indicating that GAIS maintains the overall performance of the models.

- In some cases, the reduced datasets obtained using GAIS perform better than the original datasets. For example, on the Heart Disease dataset, the accuracy on the reduced dataset (0.9355) is higher than that on the original dataset (0.9032). Similarly, on the Diabetes dataset, the precision, recall, and F1 score are all higher on the reduced dataset compared to the original dataset. These instances demonstrate that GAIS can potentially remove noisy or less informative instances, leading to improved model performance.
- There are also cases where the performance on the reduced datasets is slightly lower than that on the original datasets. For instance, on the Nursery dataset, the accuracy on the reduced dataset (0.9460) is lower than that on the original dataset (0.9992). However, it is important to consider the trade-off between dataset reduction and performance. In this case, GAIS achieves a high reduction rate of 0.9334 while still maintaining a relatively high accuracy.

The results indicate that leveraging GATs for IS can effectively reduce dataset size while maintaining model performance. The high reduction rates achieved, coupled with the preservation of performance metrics, suggest that the adaptive attention mechanism contributes to more informed IS.

Figure 1 illustrates the optimal GAT hyperparameters across datasets, revealing several insights:

- **Hidden Units:** Optimal values range from 4 to 16, with most datasets between 7 and 11. Higher units (e.g., 16 in "heart" and "hcc") suggest more complex patterns, while lower units (e.g., 4 in "fars" and 5 in "stroke") indicate simpler architectures suffice.
- **Input Heads:** Optimal values range from 5 to 15, commonly between 8 and 10. Higher input heads (e.g., 15 in "chess" and "ringnorm") are beneficial for capturing complex relationships, whereas lower input heads (e.g., 5 in "nursery" and "fars") are adequate for simpler datasets.
- **Output Heads:** Less variation is observed, with optimal values from 1 to 4, mostly 3 or 4. Multiple output representations generally enhance performance, but datasets
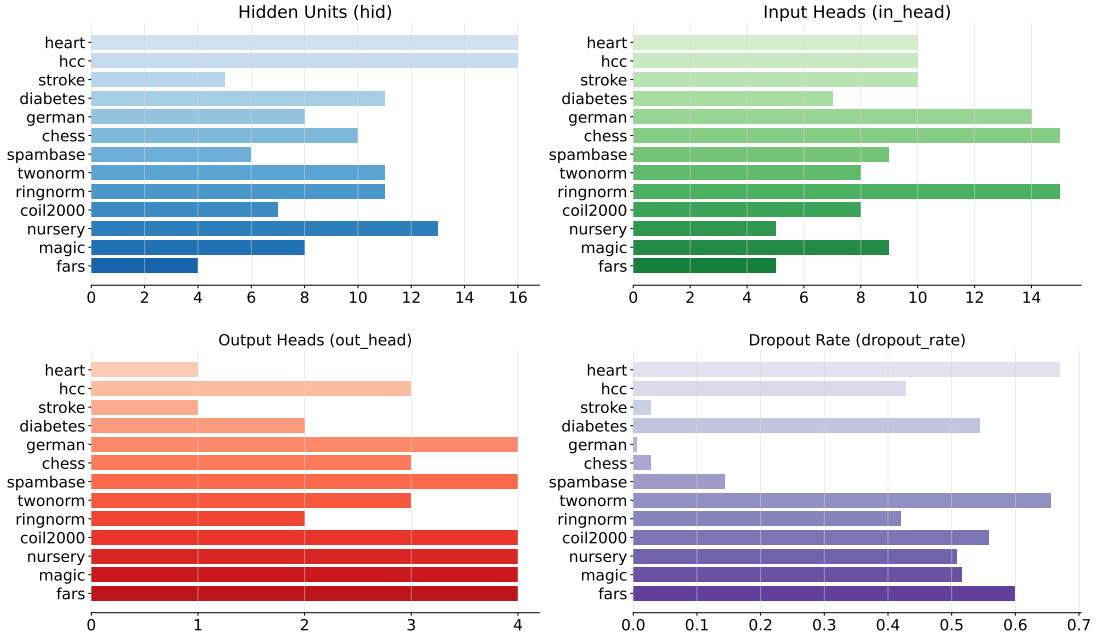
Fig. 1: GAT Hyperparameter Distribution Over the Datasets.

TABLE III: Comparative Performance Analysis of IS Methods Based on Effectiveness Metric.

Dataset	CNN	DROP3	ENN	IB3	ICF	LDIS	RMHC	GAIS
Heart Disease	0.460	0.590	0.645	0.168	0.553	0.550	0.500	**0.893**
HCC	0.839	-	0.067	0.750	0.851	0.610	0.428	**0.932**
Stroke Prediction	0.375	0.651	0.213	0.482	0.473	0.644	0.379	**0.804**
Diabetes	0.360	0.641	0.185	0.381	0.487	0.652	0.388	**0.718**
German Credit	0.375	0.711	0.224	0.202	0.335	0.660	0.385	**0.766**
Chess End-Game	0.778	0.588	0.102	0.618	0.844	0.753	0.496	**0.912**
Spam-base	0.716	0.832	0.093	0.602	0.779	0.756	0.447	**0.886**
Ring-norm	0.717	-	0.250	0.749	0.784	0.405	0.486	**0.966**
Two-norm	0.806	0.936	0.324	0.826	0.915	0.824	0.492	**0.960**
Coil-2000	0.707	0.303	0.145	0.809	0.739	0.838	0.463	**0.906**
Nursery	0.703	-	0.093	0.697	0.370	0.550	0.504	**0.883**
Magic	0.562	0.720	0.169	0.639	0.624	0.731	0.445	**0.830**
Fars	0.465	-	-	-	0.528	0.516	0.400	**0.722**

like "heart" and "stroke" perform best with a single output head.

- **Dropout Rate:** Optimal dropout rates vary widely from 0.005 to 0.67, indicating dataset-dependent regularization needs. Minimal dropout (e.g., 0.005 in "german") suggests less overfitting risk, while higher dropout (e.g., 0.67 in "heart") indicates a need for stronger regularization.

Figure 2 illustrates the effect of the confidence threshold (θ_c) and the random threshold (θ_r) on test effectiveness across five datasets. The confidence threshold in GAIS filters confidence scores from the GAT model to select the most representative instances.

For θ_c, the general trend shows that at lower values, test effectiveness remains relatively constant. As θ_c increases (particularly around 0.6–0.7 and above), test effectiveness rises, suggesting that higher thresholds allow GAIS to select more informative instances, enhancing performance. However, when θ_c reaches very high values (e.g., 0.95), test effectiveness

drops significantly, likely due to selecting too few instances to capture underlying data patterns.

The random threshold θ_r is used to randomly mask training data during GAT model training. Its impact varies across datasets. In some datasets like *magic* and *nursery*, test effectiveness steadily increases with higher θ_r, indicating that masking more training data helps the GAT model learn more robust representations. In others like *chess* and *spambase*, the increase is marginal. Similar to θ_c, when θ_r is very high (e.g., 0.95), test effectiveness dips across all datasets due to excessive masking leaving insufficient information for learning.

These findings emphasize the importance of carefully tuning θ_c and θ_r based on dataset characteristics, balancing between masking training data and preserving enough information for effective learning.

The results presented in Tables III and IV demonstrate the superior performance of GAIS across all datasets compared
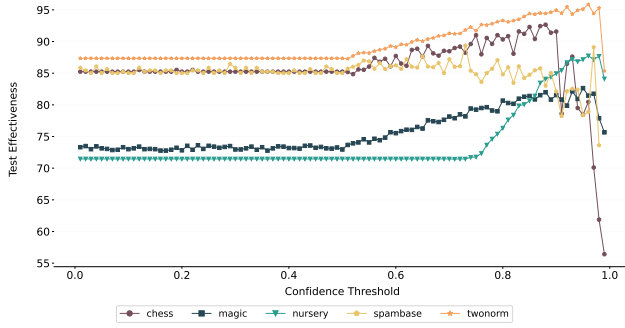
(a) Confidence Threshold
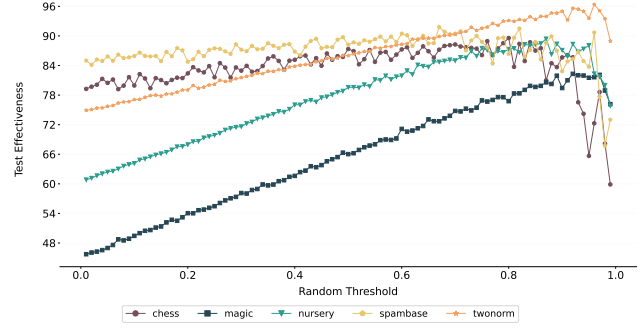
(b) Random Threshold

Fig. 2: The Effect of GAIS Hyperparameters on Effectiveness Across Five Datasets.

TABLE IV: Comparative Performance Analysis of IS Methods Based on the IS Time.

Dataset	CNN	DROP3	ENN	IB3	ICF	LDIS	RMHC	GAIS
Heart Disease	0.689	0.309	**0.003**	0.731	1.195	0.006	0.175	17.4431
HCC	1.415	-	**0.005**	2.492	1.614	**0.005**	0.121	7.179
Stroke Prediction	3.219	0.766	**0.006**	1.007	1.304	**0.006**	0.274	15.410
Diabetes	1.729	0.559	**0.006**	2.415	3.601	0.007	0.018	7.241
German Credit	13.571	10.594	**0.026**	11.634	15.046	0.033	0.165	9.080
Chess End-Game	75.336	24.462	0.104	11.883	26.673	**0.080**	0.811	19.927
Spam-base	100.473	3.981	0.041	72.207	29.926	**0.035**	1.519	17.456
Ring-norm	178.647	-	0.059	68.873	34.232	**0.046**	0.946	79.585
Two-norm	264.087	6.505	**0.043**	62.518	26.739	0.206	0.457	44.670
Coil-2000	239.117	6.993	0.126	58.203	69.286	**0.106**	0.827	43.727
Nursery	744.851	-	0.189	100.068	4.222	**0.177**	3.487	48.376
Magic	729.109	23.478	0.315	219.844	270.974	**0.298**	6.495	205.855
Fars	2306.831	-	-	-	2578.844	**2.380**	63.949	387.421

to well-known IS methods: CNN [5], DROP3 [35], ENN [6], IB3 [36], ICF [7], LDIS [16], and RMHC [37]. We also wanted to include graph-based methods, GGE and RNGE, in the comparison, but they were found to be extremely slow and computationally expensive, making them infeasible for practical use. Table III shows the effectiveness (E) of various IS methods. The GAIS method consistently achieves the highest effectiveness scores, indicating its ability to select informative and representative instances while maintaining high classification accuracy. However, as shown in Table IV, it is important to note that GAIS's IS time (T_{IS}) is slightly higher than some of the other methods. This can be attributed to the fact that GAIS involves preprocessing the data, training the GAT model, tuning the confidence threshold, and training the ML model, which are computationally more expensive than traditional IS techniques. Despite the longer processing time, the significant improvement in effectiveness demonstrates that GAIS provides a valuable trade-off between computational cost and performance in IS tasks.

VII. Conclusion

The proposed GAIS method emerges as a novel and effective approach to IS in ML. Its ability to adapt to different datasets, combined with its innovative use of attention mechanisms, sets it apart from traditional methods. The results across the various datasets not only validate the method's effectiveness but also highlight its potential to advance IS,

particularly in complex and diverse data environments. The proposed method shows promising results in various domains, particularly in maintaining or improving performance with significantly reduced training data. However, there are variations in precision across different datasets, indicating the need for method refinement in certain scenarios. The high effectiveness scores across the board highlight the method's potential in practical applications where reducing computational resources without sacrificing performance is crucial. These results advocate for further investigation and application of the GAIS method, particularly in environments where computational resources are limited or data acquisition is costly.

While GAIS demonstrates significant adaptability and effectiveness across various datasets, it has limitations. The method's performance depends on resource-intensive hyperparameter tuning, and its scalability is challenged by the computational demands of GAT when dealing with extremely large datasets. Furthermore, GAIS's effectiveness with highly imbalanced datasets and its interpretability in complex scenarios require further exploration.

Future enhancements for the GAIS method could explore several avenues. Algorithmic advancements might focus on expanding network architectures and incorporating recent developments in attention mechanisms. To address scalability challenges, particularly for larger datasets, future research could explore more efficient graph construction techniques

and investigate alternative IS methods that do not rely on deep learning approaches. The method's robustness could potentially be improved by integrating data augmentation, transfer learning, and adversarial testing techniques. To enhance the method's applicability in real-world scenarios, future work could emphasize model interpretability and evaluate the method's performance in practical settings.

REFERENCES

[1] P. Warden and D. Situnayake, *Tinyml: Machine learning with tensorflow lite on arduino and ultra-low-power microcontrollers*. O'Reilly Media, 2019.

[2] R. Rodríguez Outeiral, P. J. González, E. E. Schaake, U. A. van der Heide, and R. Simões, "Deep learning for segmentation of the cervical cancer gross tumor volume on magnetic resonance imaging for brachytherapy," *Radiation Oncology*, vol. 18, no. 1, p. 91, 2023.

[3] S. Garcia, J. Derrac, J. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 3, pp. 417–435, 2012.

[4] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *Knowledge and information systems*, vol. 35, pp. 249–283, 2013.

[5] P. Hart, "The condensed nearest neighbor rule (corresp.)," *IEEE transactions on information theory*, vol. 14, no. 3, pp. 515–516, 1968.

[6] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 3, pp. 408–421, 1972.

[7] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," *Data mining and knowledge discovery*, vol. 6, pp. 153–172, 2002.

[8] C. Zang, Z. Pei, and J. Cheng, "A method of instance selection for large size data set," *ICIC express letters. Part B, Applications : an international journal of research and surveys*, vol. 4, no. 4, pp. 1015–1022, 2013.

[9] Y. Yu, F. Zhu, and X. Wu, "Multi-speaker prosodic instance selection for hmm-based speech synthesis," in *2013 IEEE China Summit and International Conference on Signal and Information Processing*. IEEE, 2013, pp. 142–146.

[10] Q. Peng and Y.-m. Cheung, "Outlier detection based on local kernel regression for instance selection," *International Journal of Computational Intelligence Systems*, vol. 7, no. 4, pp. 748–757, 2014.

[11] S.-H. Lee and J. S. Lim, "Evolutionary instance selection algorithm based on takagi-sugeno fuzzy model," *Applied Mathematics & Information Sciences*, vol. 8, no. 3, p. 1307, 2014.

[12] R. Klinger and P. Cimiano, "Instance selection improves cross-lingual model training for fine-grained sentiment analysis," in *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, 2015, pp. 153–163.

[13] W.-C. Lin, C.-F. Tsai, S.-W. Ke, C.-W. Hung, and W. Eberle, "Learning to detect representative data for large scale instance selection," *Journal of Systems and Software*, vol. 106, pp. 1–8, 2015.

[14] V. Stanovov, E. Semenkin, and O. Semenkina, "Self-configuring hybrid evolutionary algorithm for fuzzy imbalanced classification with adaptive instance selection," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 6, no. 3, pp. 173–188, 2016.

[15] E. Biçici, "Parfda for instance selection for statistical machine translation," in *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, 2016, pp. 252–258.

[16] J. L. Carbonera and M. Abel, "A density-based approach for instance selection," in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2015, pp. 768–774.

[17] ——, "A Novel Density-Based Approach for Instance Selection," in *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, nov 2016, pp. 549–556.

[18] M. Malhat, M. El Menshawy, H. Mousa, and A. El Sisi, "A new approach for instance selection: Algorithms, evaluation, and comparisons," *Expert Systems with Applications*, vol. 149, p. 113297, 2020.

[19] R. A. R. Ashfaq, Y.-l. He, and D.-g. Chen, "Toward an efficient fuzziness based instance selection methodology for intrusion detection system," *International Journal of Machine Learning and Cybernetics*, vol. 8, pp. 1767–1776, 2017.

[20] Y. Villuendas-Rey, C. Rey-Benguría, M. Lytras, C. Yáñez-Márquez, and O. Camacho-Nieto, "Simultaneous instance and feature selection for improving prediction in special education data," *Program*, vol. 51, no. 3, pp. 278–297, 2017.

[21] D. Kavrin and S. Subbotin, "Bagging-based instance selection for instance-based classification." in *CMIS*, 2020, pp. 769–783.

[22] L. Yuan, G. Xu, L. Zhao, X. Wen, and H. Xu, "Multiple-instance learning via multiple-point concept based instance selection," *International Journal of Machine Learning and Cybernetics*, vol. 11, pp. 2113–2126, 2020.

[23] T. DeVries, M. Drozdzal, and G. W. Taylor, "Instance selection for gans," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 285–13 296, 2020.

[24] K. Nikolaidis, E. Rodriguez-Martinez, J. Y. Goulermas, and Q. H. Wu, "Spectral graph optimization for instance reduction," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 7, pp. 1169–1175, 2012.

[25] N.-B. Nguyen, K. M. Nguyen, C. M. Van, and D.-D. Le, "Graph-based visual instance mining with geometric matching and nearest candidates selection," in *2017 9th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, 2017, pp. 263–268.

[26] H. Wang, S.-B. Wang, and Y.-F. Li, "Instance selection method for improving graph-based semi-supervised learning," *Frontiers of Computer Science*, vol. 12, pp. 725–735, 2018.

[27] L. Yang, Q. Zhu, J. Huang, D. Cheng, Q. Wu, and X. Hong, "Natural neighborhood graph-based instance reduction algorithm without parameters," *Applied Soft Computing*, vol. 70, pp. 279–287, 2018.

[28] J. S. Sánchez, F. Pla, and F. J. Ferri, "Prototype selection for the nearest neighbour rule through proximity graphs," *Pattern Recognition Letters*, vol. 18, no. 6, pp. 507–513, 1997.

[29] E. Marchiori, "Hit miss networks with applications to instance selection," 2008.

[30] ——, "Class conditional nearest neighbor for large margin instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 364–370, 2009.

[31] K. Nonaka, S. Shekkizhar, and A. Ortega, "Graph-based deep learning analysis and instance selection," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2020, pp. 1–6.

[32] X. Lou, G. Liu, and J. Li, "Asiam-hgnn: Automatic selection and interpretable aggregation of meta-path instances for heterogeneous graph neural network," *Computing and Informatics*, vol. 42, no. 2, pp. 257–279, 2023.

[33] N. Zaki, W. Qin, and A. Krishnan, "Graph-based methods for cervical cancer segmentation: Advancements, limitations, and future directions," *AI Open*, 2023.

[34] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: ACM, jul 2019, pp. 2623–2631.

[35] W. D Randall and M. Tony R, "Reduction Techniques for Instance-Based Learning Algorithms," *Machine Learning*, vol. 38, pp. 257–286, 2000.

[36] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 37–66, jan 1991.

[37] D. B. Skalak, "Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms," in *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 293–301.