People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of 8 May 1945 Guelma
Faculty of Mathematics, Computer Science & Science of Matter

Department of Computer Science

Master's Thesis

Field: Computer Science

Option: Information and Communication Science and Technology

Accelerating Parametric Nonlinear Dimension Reduction via Representative Selection

---

Supervised by:

– Dr. CHOHRA Chemseddine

Presented by:

ABDELLAOUI Issam Eddine

July 2026

Acknowledgement

Alhamdulillah – In the name of Allah, the Most Compassionate, the Most Merciful. All praise and gratitude are due to Him alone, for it is by His grace, guidance, and mercy that this work has reached completion. Whatever strength, patience, and clarity carried me through this journey were gifts from Him, and I remain humbly grateful.

I owe profound gratitude to my supervisor, **Dr. Chohra Chemseddine**.Few are fortunate enough to be guided by someone who was once a fellow student and has now become a trusted mentor, and I count myself among them. His expertise shaped this research, but it was his patience, his confidence in my abilities, and his constructive guidance at every stage that truly made this work possible. Thank you for a mentorship I will always value.

I extend my appreciation to the members of the jury members for the honor of evaluating this work and for their valuable remarks.

To my beloved wife, no acknowledgement could fully capture what your support has meant. Your patience through the long hours, your unwavering belief in me, and your steady encouragement carried me through the hardest parts of this journey. This achievement is as much yours as it is mine.

To my daughters, Darine and Nadine, you have been my light through every difficult moment. Your laughter and innocence reminded me why every effort was worthwhile, and your presence gave me the motivation to keep going when it mattered most.

To my parents, words will never be enough. Your unconditional love, your sacrifices, and your prayers have shaped everything I have become and everything I have achieved. Whatever I accomplish in life rests on the foundation you built for me, and I remain

forever in your debt.

To my brother and my two sisters, thank you for your steady encouragement, your understanding, and the moral support you offered throughout this journey. Having you by my side made the difficult moments easier to bear.

Finally, to everyone who contributed to this work in ways large or small, whether through advice, encouragement, practical help, or a quiet prayer on my behalf, I offer my sincere thanks. Your kindness has not gone unnoticed, and I will carry my gratitude for it always.

Abstract

High-dimensional data continues to pose fundamental challenges across scientific and engineering disciplines, including computational complexity, the curse of dimensionality, and the difficulty of extracting meaningful representations. A prior thesis proposed a two-phase dimensionality reduction framework in which clustering-based representative selection was followed by standard UMAP training, demonstrating significant gains in computational efficiency while preserving embedding quality. We extend that foundational work to parametric non-linear dimensionality reduction methods. Specifically, we investigate parametric UMAP and autoencoders as the second phase of the pipeline. Parametric approaches offer several advantages over their non-parametric counterparts: they learn an explicit mapping function that can be applied to unseen data without retraining, they integrate naturally with deep learning workflows, and they enable end-to-end optimization with task-specific objectives. We evaluate the proposed two phases parametric dimension reduction framework on high-dimensional benchmark datasets, comparing it against the original parametric methods in terms of embedding quality, computational efficiency, generalization to new data, and downstream classification performance. The results demonstrate that combining representative selection with parametric embedding models yields a scalable, transferable, and efficient dimensionality reduction pipeline suitable for real-world, large-scale applications.

Keywords: Dimensionality reduction, parametric UMAP, autoencoders, representative selection, clustering, neural embeddings, deep learning.

Résumé

Les données de haute dimension continuent de poser des défis fondamentaux dans les disciplines scientifiques et techniques, notamment la complexité computationnelle, la malédiction de la dimensionnalité et la difficulté d'extraire des représentations pertinentes. Un mémoire précédente a proposé un cadre de réduction de dimension en deux phases dans lequel une sélection de représentants basée sur le regroupement (clustering) était suivie d'un entraînement UMAP standard, démontrant des gains significatifs en efficacité de calcul tout en préservant la qualité des projections. Nous étendons ce travail aux méthodes paramétriques non linéaires de réduction de dimension. Plus précisément, nous étudions UMAP paramétrique et les autoencodeurs comme seconde phase du pipeline. Les approches paramétriques présentent plusieurs avantages par rapport à leurs homologues non paramétriques : elles apprennent une fonction de mappage explicite pouvant être appliquée à des données inédites sans réentraînement, elles s'intègrent naturellement dans les flux de travail de l'apprentissage profond, et elles permettent une optimisation de bout en bout avec des objectifs spécifiques à la tâche. Nous évaluons le cadre de réduction de dimension paramétrique en deux phases proposé sur des jeux de données de référence à haute dimension, en le comparant aux méthodes paramétriques originales en termes de qualité des projections, d'efficacité computationnelle, de généralisation à de nouvelles données et de performance en classification. Les résultats montrent que la combinaison d'une sélection de représentants avec des modèles d'embedding paramétriques conduit à un pipeline de réduction de dimension évolutif, transférable et efficace, adapté aux applications réelles à grande échelle.

Mots-clés: réduction de dimension, UMAP paramétrique, autoencodeurs, sélection de représentants, regroupement, embeddings neuronaux, apprentissage profond.

Contents

List of Figures **IV**

List of Tables **V**

1 High-Dimensional Data and Its Challenges **4**
1.1 Introduction . . . 4
1.2 Definition and Applications . . . 4
1.2.1 Application Domains . . . 5
1.3 Key Characteristics of High-Dimensional Data . . . 5
1.3.1 Sparsity and the Empty Space Phenomenon . . . 5
1.3.2 Distance Concentration . . . 5
1.3.3 Feature Redundancy and Correlation . . . 6
1.4 The Curse of Dimensionality . . . 6
1.5 Computational Challenges . . . 6
1.6 Dimensionality Reduction: Motivation and Goals . . . 7
1.7 Conclusion . . . 8

2 State of the Art **9**
2.1 Introduction . . . 9
2.2 Dimensionality Reduction: An Overview . . . 9
2.2.1 Feature Selection . . . 9
2.2.2 Feature Extraction: Linear Methods . . . 10
2.3 Non-Linear Dimensionality Reduction . . . 10
2.3.1 t-Distributed Stochastic Neighbor Embedding (t-SNE) . . . 11
2.3.2 Uniform Manifold Approximation and Projection (UMAP) . . . 11
2.3.3 Other Non-Linear Methods . . . 11
2.4 Representative Selection . . . 12
2.4.1 Clustering-Based Representative Selection . . . 12
2.4.2 Prototype Selection and Core-Sets . . . 12
2.5 Parametric Dimensionality Reduction . . . 13
2.5.1 Autoencoders . . . 13

2.5.2 Parametric UMAP . . . 14
2.5.3 Other Neural Embedding Approaches . . . 15
2.6 Two-Phase Dimensionality Reduction Pipelines . . . 15
2.6.1 Positioning of This Work . . . 16
2.7 Conclusion . . . 16

3 Approach and Methodology **17**
3.1 Introduction . . . 17
3.2 Overview of the Proposed Methodology . . . 17
3.3 Phase 1: Representative Selection . . . 19
3.3.1 K-Means . . . 20
3.3.2 BIRCH Clustering . . . 20
3.4 Phase 2: Parametric Embedding Models . . . 21
3.5 Dataset Description . . . 21
3.5.1 IoTID20 Dataset . . . 21
3.5.1.1 Dataset Preprocessing . . . 21
3.5.1.2 Multi-Tier Attack Taxonomy . . . 22
3.5.1.3 Attack Diversity . . . 22
3.5.1.4 Relevance to the Study . . . 23
3.5.2 X-IIoTID Dataset . . . 23
3.5.2.1 Key Characteristics . . . 23
3.5.2.2 Dataset Statistics and Feature Space . . . 24
3.5.2.3 Attack Taxonomy . . . 24
3.5.2.4 Attack Distribution . . . 24
3.5.2.5 Relevance to the Study . . . 25
3.6 Experimental Configuration . . . 25
3.6.1 Representative Selection . . . 26
3.6.2 Parametric Embedding Models . . . 26
3.6.2.1 Parametric UMAP . . . 26
3.6.2.2 Autoencoder . . . 27
3.6.3 Embedding Evaluation . . . 27
3.7 Evaluation Metrics: . . . 28
3.7.1 Mean Squared Error (MSE) . . . 28
3.7.2 Neighborhood Preservation: . . . 28
3.7.3 Classification Performance: . . . 29
3.7.4 Computational Efficiency: . . . 29
3.7.5 Model Size: . . . 30
3.8 conclusion . . . 30

4 Experimental Evaluation **31**
4.1 Introduction 31
4.2 Hardware and Software Environment 31
4.3 Embedding Quality 32
4.3.1 Neighborhood Preservation 32
4.3.2 Mean Squared Error (MSE) 35
4.3.3 Classification Accuracy 38
4.4 Computational Efficiency 44
4.5 Discussion 46
4.6 Comparison with Previous Work 48
4.7 Conclusion 50

Bibliography **55**

List of Figures

3.1 Overview of the proposed two-phase parametric dimensionality reduction framework. 19

4.1 Neighborhood preservation across dimensionality reduction methods and sampling strategies for `iotid20` dataset. Higher values indicate better preservation of the original neighborhood structure. The horizontal dashed line represents the neighborhood preservation score obtained when training on the complete dataset. 33
4.2 Neighborhood preservation across dimensionality reduction methods and sampling strategies for `xiiotid` dataset. Higher values indicate better preservation of the original neighborhood structure. The horizontal dashed line represents the neighborhood preservation score obtained when training on the complete dataset. 34
4.3 Mean Squared Error (MSE) across dimensionality reduction methods and sampling strategies for the `iotid20` dataset. Lower values indicate better reconstruction quality. The horizontal dashed line represents the MSE obtained when training the dimensionality reduction model on the complete dataset. 36
4.4 Mean Squared Error (MSE) across dimensionality reduction methods and sampling strategies for the `xiiotid` dataset. Lower values indicate better reconstruction quality. The horizontal dashed line represents the MSE obtained when training the dimensionality reduction model on the complete dataset. 37
4.5 Training time (in seconds) for `iotid20` dataset across different dimensionality reduction methods and representative selection strategies. 44
4.6 Training time (in seconds) for the `xiiotid` dataset across different dimensionality reduction methods and representative selection strategies. 45

List of Tables

3.1 Attack class distribution in the X-IIoTID dataset. 25
3.2 Mini-Batch K-Means configurations. 26
3.3 BIRCH threshold configurations. 26
3.4 Parametric UMAP hyperparameters. 27
3.5 Autoencoder hyperparameters. 27
3.6 Classification hyperparameters. 27

4.1 Test Environment 31
4.2 Classification accuracy (%) obtained by the SVM classifier for the different dimensionality reduction and representative selection strategies. 39
4.3 Classification accuracy (%) obtained by the CART classifier for the different dimensionality reduction and representative selection strategies. 40
4.4 Classification accuracy (%) obtained by the KNN classifier for the different dimensionality reduction and representative selection strategies. 42
4.5 Classification accuracy (%) obtained by the MLP classifier for the different dimensionality reduction and representative selection strategies. 43

General Introduction

The ever-growing volume and complexity of data generated by modern systems, ranging from Internet of Things (IoT) [1, 2] and biomedical instruments [3, 4] to natural language processing [5, 6] and high-resolution imagery [7, 8], have made high-dimensional data analysis a central concern in contemporary machine learning and data science. Datasets with hundreds or thousands of features are now commonplace, yet their sheer dimensionality introduces well-documented difficulties: the curse of dimensionality degrades model performance, computational costs escalate rapidly, and meaningful visualization becomes infeasible without effective reduction techniques.

Dimensionality reduction addresses these challenges by projecting high-dimensional data into compact, lower-dimensional representations that preserve the essential structure of the original space. While linear methods such as Principal Component Analysis (PCA) [9] remain widely used for their simplicity and efficiency, they are fundamentally limited in their ability to capture non-linear manifold structures. Non-linear techniques, most notably t-Distributed Stochastic Neighbor Embedding (t-SNE) [10] and Uniform Manifold Approximation and Projection (UMAP) [11] have therefore gained considerable traction, as they can faithfully represent complex data geometries in low-dimensional spaces. However, applying non-linear dimensionality reduction to large-scale datasets remains computationally demanding. Standard UMAP, for instance, operates over the entire dataset during training, leading to significant time and memory requirements that can be prohibitive for resource-constrained or real-time applications. A prior thesis by Amiri [12] addressed this limitation through a two-phase framework: first, a compact yet structurally representative subset is selected via clustering algorithms, second, UMAP is trained exclusively on this reduced subset, and the learned model is used to transform the full dataset. That work demonstrated that the two-phase pipeline achieves substantial reductions in training time and memory consumption while maintaining high embedding quality and downstream classification accuracy.

Motivation and Continuity

Despite the success of non-linear dimensionality reduction techniques such as UMAP, their standard non-parametric formulation presents several practical limitations. Non-

parametric UMAP generates low-dimensional embeddings by optimizing the positions of data points directly in the embedding space rather than learning an explicit mapping function from the original high-dimensional space. Although UMAP supports out-of-sample projection, the absence of a learned parametric mapping restricts its applicability in several modern machine learning scenarios.

Several challenges arise from this limitation:

- **Scalability of inference:** Embedding new samples requires nearest-neighbor searches and additional computations based on the original training data, which may become costly for large-scale or real-time applications.
- **Integration with deep learning systems:** Since no differentiable encoder is learned, standard UMAP cannot be naturally incorporated into end-to-end neural network architectures.
- **Model portability and deployment:** The embedding is tied to the training dataset and graph structure rather than being represented by a compact learned model that can be easily deployed across different environments.

To address these limitations, this thesis extends the second phase of the proposed dimensionality reduction framework by investigating *parametric* non-linear embedding methods. Specifically, we consider Parametric UMAP [13], and autoencoders [14]. Unlike non-parametric approaches, these methods learn an explicit encoder function, typically parameterized by a neural network, that maps high-dimensional inputs directly to low-dimensional representations. Once trained, the encoder can generate embeddings through a simple forward pass, enabling efficient inference, seamless integration into deep learning pipelines, and optimization with task-specific objectives. This extension preserves the computational advantages of the representative-instance selection phase while providing the flexibility and scalability of learned parametric models.

Objectives

The primary objectives of this work are:

1. **Extend the two-phase framework:** Retain the clustering-based representative selection from the prior work and replace the second phase with parametric embedding models (parametric UMAP, autoencoders, neural embeddings).
2. **Evaluate computational efficiency and embedding performance:** Compare Parametric UMAP and Autoencoder models trained on the full dataset with the same models trained on representative subsets obtained through representative selection. The comparison is conducted in terms of embedding quality, training time,

memory usage, and generalization capability, with the goal of quantifying the computational savings achieved by representative selection while preserving the quality of the learned embeddings.

3. **Assess downstream performance:** Measure the impact of parametric embeddings on downstream classification tasks across multiple evaluation scenarios.

4. **Demonstrate scalability:** Validate that the combined pipeline representative selection followed by parametric embedding delivers practical efficiency gains for large-scale, high-dimensional datasets.

Thesis Organization

The remainder of this thesis is organized as follows:

- **Chapter 1: High-Dimensional Data and Its Challenges** provides the necessary background on high-dimensional data, its characteristics, and the fundamental challenges it poses, including the curse of dimensionality and computational complexity.

- **Chapter 2: State of the Art** reviews dimensionality reduction techniques, covering both linear and non-linear methods, representative selection strategies, and the prior two-phase framework. It also positions this work within the existing literature.

- **Chapter 3: Proposed Approach** details the extended two-phase methodology, describing the parametric embedding models adopted as the second phase and the rationale behind their selection.

- **Chapter 4: Experimental Evaluation** presents the experimental setup, evaluation metrics, results, and discussion.

Chapter 1

High-Dimensional Data and Its Challenges

1.1 Introduction

Modern data acquisition technologies produce datasets of ever-increasing complexity. Fields as diverse as the Internet of Things (IoT), genomics, medical imaging, and natural language processing routinely generate observations described by hundreds, thousands, or even millions of features. This abundance of measured variables—referred to as *high-dimensional data*—creates new opportunities for pattern discovery and predictive modeling, but it simultaneously introduces fundamental obstacles that traditional analytical methods are ill-equipped to handle.

This chapter provides the foundational background necessary for the rest of this thesis. We define high-dimensional data and survey its principal application domains, then examine the structural characteristics that make such data difficult to analyze. We discuss the *curse of dimensionality* and its consequences for computational cost, statistical reliability, and model generalization. Finally, we introduce dimensionality reduction as the primary strategy for mitigating these challenges, setting the stage for the detailed review of specific techniques presented in Chapter 2.

1.2 Definition and Applications

High-dimensional data refers to datasets in which the number of features (variables or attributes) is large. This regime poses distinctive challenges for statistical analysis, model selection, and computational tractability [15, 16]. The notion of "high dimensionality" is context-dependent: in genomics, tens of thousands of gene expression values per sample constitute a high-dimensional setting, whereas in image analysis, even a modest 256×256 grayscale image corresponds to over 65,000 features.

1.2.1 Application Domains

High-dimensional datasets arise naturally in numerous fields:

- **Internet of Things (IoT):** Networks of interconnected sensors produce continuous streams of multi-attribute data, used for monitoring, anomaly detection, and predictive maintenance.
- **Bioinformatics and genomics:** DNA microarrays and RNA sequencing technologies measure expression levels for tens of thousands of genes, often with relatively few biological samples [15].
- **Medical imaging:** Modalities such as MRI, CT, and PET generate volumetric data with millions of voxel-based features per scan [16].
- **Computer vision:** Each pixel in a high-resolution image acts as a feature, yielding extremely large feature spaces, particularly for color or multi-spectral imagery.
- **Natural language processing:** Text representations such as TF-IDF or word embeddings produce high-dimensional and often sparse feature vectors [17].
- **Finance and economics:** Real-time monitoring of market indicators, macroeconomic variables, and transaction records yields high-dimensional time series.

These examples illustrate that high-dimensional data is not confined to a single discipline; rather, it is a pervasive characteristic of modern data-intensive analysis.

1.3 Key Characteristics of High-Dimensional Data

1.3.1 Sparsity and the Empty Space Phenomenon

As the number of dimensions grows, data points become progressively more isolated within the feature space. In high-dimensional hypercubes, most of the volume concentrates near the boundaries rather than the center, a phenomenon known as the *empty space phenomenon*. Consequently, algorithms that depend on local density or proximity, such as k-nearest neighbors or density-based clustering, become less reliable because the notion of a "dense neighborhood" loses its meaning [18].

1.3.2 Distance Concentration

In high-dimensional spaces, the ratio between the maximum and minimum pairwise distances among data points converges toward unity [19, 20, 21]. This effect, termed *distance*

concentration, erodes the discriminatory power of distance metrics. When all points appear roughly equidistant, distance-based methods such as k-NN classification and k-means clustering can no longer meaningfully distinguish between neighbors and non-neighbors.

1.3.3 Feature Redundancy and Correlation

High-dimensional datasets frequently contain correlated or linearly dependent features. Such redundancy inflates model complexity, increases variance in parameter estimates, and obscures the identification of truly informative variables. Feature selection and extraction techniques are therefore essential for isolating the relevant signal within the data [22, 23].

1.4 The Curse of Dimensionality

The term *curse of dimensionality*, originally coined by Bellman, encompasses a collection of adverse phenomena that emerge as the dimensionality of data increases [24]:

- **Exponential volume growth:** The volume of a p-dimensional hypercube grows exponentially with p.
- **Data sparsity:** As dimensionality increases, data points become increasingly isolated, making statistical estimation unstable and pattern recognition more difficult.
- **Overfitting:** The number of model parameters grows with dimensionality, and without proportionally more training data, models memorize noise rather than learning generalizable patterns.
- **Loss of meaningful neighborhoods:** The convergence of pairwise distances (distance concentration) means that defining similarity or proximity becomes unreliable in high-dimensional spaces [20, 21].

These effects compound to make high-dimensional data both computationally expensive and statistically unreliable for many traditional algorithms.

1.5 Computational Challenges

Beyond statistical issues, high dimensionality imposes severe computational burdens. Many machine learning algorithms scale at least linearly, and often quadratically or cubically, with the number of features:

- PCA requires an eigendecomposition of the $p \times p$ covariance matrix, yielding $\mathcal{O}(p^3)$ time complexity.

- t-SNE has $\mathcal{O}(n^2)$ time and space complexity, limiting its applicability to moderate-size datasets.
- UMAP scales more favorably but still incurs substantial memory overhead when applied to large, high-dimensional inputs.

For applications involving real-time data streams (e.g., IoT intrusion detection) or resource-constrained devices, these computational bottlenecks represent critical practical limitations [25].

1.6 Dimensionality Reduction: Motivation and Goals

Dimensionality reduction provides a principled strategy for transforming high-dimensional data into lower-dimensional representations while retaining the essential structure and information content of the original space [26]. The principal objectives are:

- **Visualization:** Projecting data into two or three dimensions enables visual exploration of clusters, outliers, and structural patterns.
- **Computational efficiency:** Reducing the number of features decreases the time and memory required for model training and inference.
- **Noise elimination:** By discarding dimensions associated with low variance or irrelevant variability, dimensionality reduction improves model robustness.
- **Improved generalization:** Lower-dimensional representations mitigate overfitting by constraining model complexity.
- **Interpretability:** Compact representations are easier to visualize, analyze, and explain.

Dimensionality reduction methods are broadly categorized into *linear* techniques (e.g., PCA, LDA), which seek linear projections, and *non-linear* techniques (e.g., t-SNE, UMAP, autoencoders), which can capture complex manifold structures. Non-linear methods are of particular interest when the intrinsic geometry of the data is curved or locally varying, which is a common scenario in real-world datasets. Chapter 2 provides a detailed review of both families of methods, with emphasis on the non-linear approaches that form the core of this thesis.

1.7 Conclusion

This chapter has established the foundational context for this thesis. We defined high-dimensional data, surveyed its prevalence across application domains, and examined the structural and computational challenges it introduces. The curse of dimensionality, encompassing sparsity, distance concentration, overfitting, and loss of meaningful neighborhoods, motivates the need for effective dimensionality reduction. We introduced the goals and broad categories of dimensionality reduction techniques, distinguishing between linear and non-linear approaches. The following chapter provides a comprehensive review of specific methods within each category, positioning the parametric approaches investigated in this thesis within the broader landscape of dimensionality reduction research.

Chapter 2

State of the Art

2.1 Introduction

Dimensionality reduction has been a central topic in machine learning and data analysis for decades, with an extensive body of work spanning linear projections, manifold learning, and more recently, deep neural network-based embedding methods. This chapter provides a structured review of the existing literature, organized into four parts: (1) classical dimensionality reduction techniques, (2) non-linear manifold learning methods with emphasis on UMAP, (3) representative selection strategies, and (4) parametric dimensionality reduction approaches, which constitute the novel aspect of this thesis.

2.2 Dimensionality Reduction: An Overview

Dimensionality reduction encompasses the set of methods that transform high-dimensional data into lower-dimensional representations while retaining meaningful properties of the original data. The primary motivations include reducing computational cost, mitigating overfitting, enabling visualization, and eliminating noise and redundancy [27, 10]. Methods are broadly categorized into *feature selection* (retaining a subset of original features) and *feature extraction* (constructing new, lower-dimensional features).

2.2.1 Feature Selection

Feature selection identifies a subset of the original variables that are most relevant for a given task, without transforming them. Methods are classified by strategy (filter, wrapper, embedded) and by supervision level (supervised, unsupervised) [28, 29].

- **Filter methods** rank features using statistical criteria (e.g., mutual information, ANOVA F-score, correlation) independently of any learning algorithm. They are computationally efficient but do not account for feature interactions [29].

- **Wrapper methods** evaluate subsets by training a classifier and selecting the combination that maximizes predictive performance. While more accurate, they are computationally expensive and prone to overfitting [30].
- **Embedded methods** perform selection during model training, as in LASSO regularization or tree-based importance scores, balancing efficiency and relevance [31].

Feature selection preserves original feature semantics, which aids interpretability, but may fail to capture non-linear relationships or latent structure in the data [32].

2.2.2 Feature Extraction: Linear Methods

Linear dimensionality reduction methods project high-dimensional data onto a lower-dimensional subspace using linear transformations. Among them, Principal Component Analysis (PCA) is the most widely used unsupervised approach, it identifies orthogonal directions that maximize the variance of the projected data, yielding a least-squares optimal low-rank approximation of the original dataset [33, 34]. Its computational efficiency and ability to quantify information retention make it particularly attractive. On the other hand, Linear Discriminant Analysis (LDA) is a supervised technique that exploits class labels to find projections maximizing class separability by increasing between-class variance while minimizing within-class variance [35, 36]. However, like for PCA, its performance is constrained by linearity assumptions and potential singularity issues in high-dimensional settings [37, 27, 38].
Other notable linear approaches include Factor Analysis (FA), which explains observed variables through a smaller set of latent factors and noise components [39, 40], and Multidimensional Scaling (MDS), which constructs low-dimensional representations that preserve pairwise distances between samples [41, 42]. Despite their methodological differences, these techniques share the common limitation of relying on linear projections, which restricts their ability to capture complex non-linear manifolds present in many real-world datasets.

2.3 Non-Linear Dimensionality Reduction

When data resides on or near a non-linear manifold embedded in high-dimensional space, linear methods cannot faithfully represent the intrinsic geometry. Non-linear techniques address this limitation by modeling curved structures, preserving local or global manifold properties in the embedding.

2.3.1 t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE converts pairwise similarities in the original space into probability distributions and minimizes the Kullback–Leibler divergence between these distributions and corresponding ones in the embedding space [43]. t-SNE effectively models distant points without crowding, producing visually distinctive clusters. However, t-SNE has quadratic computational complexity and does not reliably preserve global structure [44, 45, 46].

2.3.2 Uniform Manifold Approximation and Projection (UMAP)

UMAP represents the current state of the art among non-parametric non-linear dimensionality reduction techniques. Grounded in Riemannian geometry and algebraic topology, UMAP constructs a fuzzy simplicial complex from k-nearest neighbor graphs and optimizes a low-dimensional embedding by minimizing the cross-entropy between the high-dimensional and low-dimensional fuzzy topological representations [11]. Compared to t-SNE, UMAP offers several advantages: faster computation, better preservation of global structure, and greater flexibility through support for arbitrary distance metrics and target dimensionalities [47].

2.3.3 Other Non-Linear Methods

Isomap and Locally Linear Embedding (LLE) are among the most influential manifold learning techniques for non-linear dimensionality reduction. Isomap extends classical MDS by approximating geodesic distances through a k-nearest neighbor graph, enabling the recovery of low-dimensional embeddings that preserve the global geometry of non-linear manifolds [48]. However, its performance is sensitive to noise and relies on assumptions such as manifold convexity [49, 50]. In contrast, LLE preserves local geometric structure by representing each data point as a linear combination of its neighbors and seeking a low-dimensional embedding that maintains these reconstruction weights [51]. While effective for smoothly sampled manifolds, LLE is sensitive to neighborhood selection and often struggles with non-uniform ata densities [52, 53]. Other notable manifold learning approaches include Hessian LLE (HLLE), which improves the recovery of locally isometric manifolds through Hessian-based constraints [54], Laplacian Eigenmaps, which preserve local neighborhood relationships using spectral graph theory [55], and Diffusion Maps, which capture intrinsic geometric structures through random-walk diffusion processes on data graphs [56]. Together, these methods demonstrated the potential of manifold learning to uncover complex non-linear structures that cannot be adequately represented by linear projections.

Prior work. In Amiri's thesis [12], standard UMAP was used as the second phase of the two-phase pipeline: after clustering-based representative selection reduced the dataset, UMAP was trained on the representative subset and the learned model was applied to transform the full dataset. This yielded order-of-magnitude gains in training time and memory usage, with unsignificant loss in embedding quality and classification accuracy. However, standard UMAP remains fundamentally a non-parametric technique, optimizing the embedding coordinates of the training samples rather than learning an explicit mapping from the original feature space to the embedding space. Consequently, its ability to generalize to unseen data relies on an approximate out-of-sample transformation procedure. Parametric dimensionality reduction methods addresse these limitations by replacing the direct embedding optimization with a neural network that learns a continuous mapping between the input and low-dimensional spaces. This enables efficient projection of arbitrary unseen samples through a simple forward pass while facilitating integration into machine learning pipelines. Moreover, because the computational cost of Parametric UMAP is dominated by training the neural network, reducing the training set through representative selection is expected to provide substantial acceleration benefits, making it a natural extension of the two-phase dimensionality reduction framework.

2.4 Representative Selection

Representative selection denotes the process of identifying a data subset that faithfully captures the diversity and structural properties of the full dataset [57]. Unlike random sampling, which may miss rare patterns, representative selection employs structured strategies to ensure comprehensive coverage of the data distribution.

2.4.1 Clustering-Based Representative Selection

Clustering algorithms partition data into groups of similar objects. By selecting centroids, medoids, or boundary points from each cluster, one obtains a compact subset that approximates the original distribution [58, 59, 60].

2.4.2 Prototype Selection and Core-Sets

Prototype selection methods, originally developed for instance-based classifiers, retain the most informative training instances (e.g., boundary points) to reduce classification cost [61, 60]. Core-sets offer a theoretically grounded alternative: they are compact, weighted subsets with provable approximation guarantees for specific optimization objectives [62, 63]. Both approaches are complementary to clustering-based selection and may be explored in future extensions of this work.

2.5 Parametric Dimensionality Reduction

While non-parametric methods like standard UMAP and t-SNE optimize embeddings point by point, *parametric* approaches learn an explicit mapping function $f_\theta : \mathbb{R}^D \to \mathbb{R}^d$ (where D is the input dimensionality and d the target dimensionality) parameterized by a set of learnable weights θ. This function can be applied to new data without retraining, integrated with other differentiable modules, and optimized end-to-end for multiple objectives. This section reviews the principal parametric methods relevant to this thesis.

2.5.1 Autoencoders

Autoencoders are neural network architectures comprising an encoder $f_\theta : \mathbb{R}^D \to \mathbb{R}^d$ and a decoder $g_\phi : \mathbb{R}^d \to \mathbb{R}^D$, trained jointly to minimize reconstruction error:

$$\mathcal{L}_{\text{AE}} = \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{x}_i - g_\phi(f_\theta(\boldsymbol{x}_i))\|^2$$

The bottleneck layer (latent space) of dimensionality $d \ll D$ forces the network to learn a compact representation that captures the most salient information in the data [14, 64]. A standard autoencoder maps inputs through one or more hidden layers to the bottleneck and then reconstructs the output. With sufficient depth and non-linear activations, autoencoders can learn complex non-linear manifold structures that linear methods cannot capture [14]. Variants include:

- **Undercomplete autoencoders:** The bottleneck dimension is strictly smaller than the input, forcing compression.
- **Deep autoencoders:** Multiple hidden layers in both encoder and decoder enable hierarchical feature learning [14].
- **Denoising autoencoders:** Trained to reconstruct clean inputs from corrupted versions, improving robustness and regularization [64].
- **Contractive autoencoders:** A penalty on the Frobenius norm of the encoder Jacobian encourages the learned representation to be robust to small perturbations [65].

Strengths. Autoencoders learn explicit mapping functions that generalize to unseen data. Their architecture can be tailored to the data modality (e.g., convolutional layers for images, recurrent layers for sequences).

Limitations. Without regularization, autoencoders may overfit to the training set. The latent space may lack structure (e.g., discontinuities), and the learned representations are not guaranteed to preserve local neighborhood relationships as manifold learning methods do [66].

2.5.2 Parametric UMAP

Parametric UMAP extends the original UMAP framework by replacing the direct optimization of embedding coordinates with a parametric model, typically a neural network, that learns an explicit mapping from the input space to the embedding space [13]. Instead of treating the low-dimensional representation as a set of independent variables optimized for a fixed dataset, the method trains a function that generalizes across all samples, allowing any new data point to be projected without recomputing the entire embedding.

The learning process preserves the same probabilistic interpretation as standard UMAP, where local neighborhood relationships in the high-dimensional space are modeled using fuzzy simplicial sets and matched in the embedding space. However, rather than directly updating the coordinates of individual points, the optimization adjusts the parameters of the neural network so that its outputs preserve these neighborhood relationships as faithfully as possible. This reformulation enables the use of standard backpropagation and stochastic gradient-based optimization, making the approach compatible with modern deep learning frameworks.

Advantages over standard UMAP.

- **Deep learning integration:** The encoder is a standard neural network and can be combined with other differentiable modules for end-to-end training.
- **Semi-supervised learning:** Label information can be incorporated into the loss function, guiding the embedding toward class-discriminative representations [13].
- **Scalability:** Mini-batch training enables efficient processing of large datasets.

Limitations.

- The quality of the embedding depends on the neural network architecture and hyperparameters.
- Training requires GPU resources and careful learning rate scheduling.
- Embedding quality may slightly differ from non-parametric UMAP due to the function approximation bottleneck.

2.5.3 Other Neural Embedding Approaches

Several neural approaches have been proposed to obtain parametric embeddings for dimensionality reduction. Parametric t-SNE trains a neural network to approximate the t-SNE embedding by optimizing the same divergence-based objective, but it inherits t-SNE's well-known limitations in preserving global structure despite improved out-of-sample generalization [43]. Siamese and triplet networks instead learn embeddings by optimizing contrastive objectives over pairs or triplets of samples, encouraging distance-preserving representations and performing particularly well when supervision or relational labels are available [64]. More recently, self-supervised representation learning methods have demonstrated strong performance by learning invariant embeddings through agreement between augmented views of the same instance, without requiring labeled data. Despite their differences, these approaches share the common principle of learning parametric mappings but differ in their objective functions and supervision requirements. In this work, the focus is restricted to Parametric UMAP as a representative manifold-preserving parametric method and Autoencoders as a reconstruction-based neural embedding framework, both of which are most directly aligned with the objectives of dimensionality reduction and scalable embedding generation.

2.6 Two-Phase Dimensionality Reduction Pipelines

Clustering-based acceleration of dimensionality reduction has recently emerged as an effective strategy for improving scalability in non-linear embedding methods. Amiri's work [12] introduced a two-phase pipeline in which a representative subset of the data is first extracted using clustering techniques such as Mini-Batch KMeans or BIRCH, and standard UMAP is subsequently trained on this reduced set. The resulting embedding is then extended to the full dataset using UMAP's out-of-sample projection mechanism. Experimental results demonstrate that this decoupled design can substantially reduce computational cost, achieving significant gains in both training time and memory efficiency. More broadly, multi-stage dimensionality reduction pipelines have been widely studied in the literature, typically combining a preprocessing or data reduction step with a downstream embedding technique. Representative examples include UMATO [67], which constructs a global structure from sampled representatives before refining local geometry using UMAP; clustering-based preprocessing combined with t-SNE for visualization tasks [68]; feature selection followed by PCA to improve linear embedding efficiency [69]; and hybrid approaches for intrusion detection that integrate clustering-based selection with non-linear dimensionality reduction methods [70]. Despite differences in methodology and application domains, these approaches share a common principle: reducing dataset complexity prior to applying a computationally intensive embedding model, thereby improving scal-

ability while preserving essential structure.

2.6.1 Positioning of This Work

Existing two-phase dimensionality reduction pipelines predominantly rely on non-parametric embedding methods, particularly standard UMAP, in the second stage. While effective, this design inherits several limitations, including the absence of an explicit learned mapping function, limited adaptability to distributional shifts, and restricted integration with end-to-end learning frameworks. In contrast, this work investigates parametric embedding models as replacements for the second phase, specifically Parametric UMAP and autoencoder-based representations. This transition enables the learning of explicit, reusable mappings from the original feature space to the embedding space, improving generalization to unseen data and facilitating integration with neural architectures. At the same time, the representative-selection stage is preserved to reduce training complexity, resulting in a hybrid framework that combines the efficiency of data reduction with the flexibility and generalization capabilities of parametric models.

2.7 Conclusion

This chapter has surveyed the landscape of dimensionality reduction, from classical linear techniques through non-linear manifold learning to modern parametric neural embedding methods. We reviewed feature selection, PCA, LDA, t-SNE, Isomap, LLE, UMAP, autoencoders, and parametric UMAP, highlighting their respective strengths and limitations. We also discussed representative selection strategies and two-phase pipelines, with particular emphasis on the prior work by Amiri that this thesis extends. The identification of parametric methods as a natural evolution of the two-phase framework motivates the proposed approach detailed in the following chapter.

Chapter 3

Approach and Methodology

3.1 Introduction

In this chapter, we present the approach and methodology used to evaluate the proposed two-phase framework for efficient dimensionality reduction and intrusion detection in high-dimensional datasets. The central hypothesis is that selecting a compact, representative subset of the data prior to training can substantially reduce computational cost while retaining the structural and discriminative information necessary for accurate downstream classification. First, we describe the two-phase pipeline that underpins our investigation. We motivate the choice of clustering-based representative selection and outline the rationale for using parametric embeddings that can generalize efficiently to unseen data. We also detail the preprocessing steps, dataset characteristics, and the experimental protocol used to assess embedding quality and intrusion detection performance. Embedding quality evaluation focuses on both classification accuracy across multiple taxonomies (binary, attack-category, and fine-grained) and on embedding structure preservation.

3.2 Overview of the Proposed Methodology

This work proposes a two-phase framework for efficient dimensionality reduction and intrusion detection in high-dimensional datasets. The primary objective is to reduce computational costs while preserving the essential structural information required for accurate data representation and classification. The proposed methodology consists of two sequential phases: representative selection and parametric embedding generation. In the first phase, representative selection techniques are applied to the preprocessed dataset to identify a subset of samples that accurately captures the diversity and distribution of the original data. By selecting representative instances rather than using the complete dataset, the framework aims to reduce training time and memory consumption while maintaining the most informative characteristics of the data. In the second phase, the

selected representative subset is used to train parametric dimensionality reduction models. Two embedding approaches are investigated:

- **Parametric UMAP:** which learns a neural-network-based mapping from the original high-dimensional feature space to a lower-dimensional embedding space while preserving the manifold structure of the data.
- **Autoencoders:** which learn compact latent representations by reconstructing the input data through an encoder-decoder architecture.

Unlike non-parametric dimensionality reduction methods, both approaches learn an explicit transformation function that can be directly applied to unseen samples, enabling efficient inference and improved scalability. After training, the learned embedding models are used to project both training and testing data into a low-dimensional representation. These embeddings are then evaluated through downstream classification tasks to determine their effectiveness in preserving discriminative information relevant to intrusion detection. The performance of the proposed framework is assessed under two experimental settings:

1. Training Parametric UMAP and Autoencoder models using the complete dataset.
2. Training Parametric UMAP and Autoencoder models using representative subsets obtained through representative selection.

The resulting embeddings are compared in terms of:

- Embedding quality and structure preservation.
- Classification performance.
- Training time.
- Memory consumption.

This evaluation makes it possible to quantify the benefits of combining representative selection with parametric dimensionality reduction and to determine whether significant computational savings can be achieved without sacrificing embedding quality or intrusion detection performance. A complete description of our experimental configuration is summarized in Figure 3.1, which illustrates the two-phase pipeline and the evaluation protocol used throughout this work. The following sections provide a detailed description of each phase, the datasets used, and the evaluation metrics employed to assess the effectiveness of the proposed methodology.

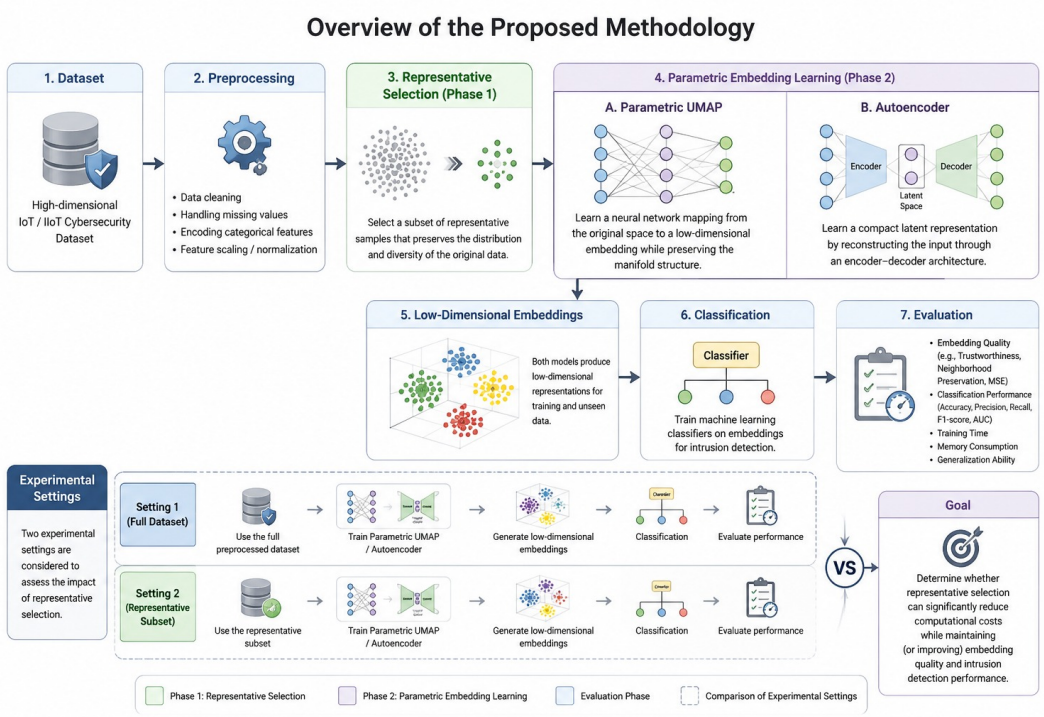


Figure 3.1: Overview of the proposed two-phase parametric dimensionality reduction framework.

3.3 Phase 1: Representative Selection

Representative selection is achieved using clustering-based techniques that partition the dataset into a set of homogeneous groups. Once the clusters are formed, representative samples are selected from each cluster, typically using the cluster centroids. The resulting subset is intended to preserve the underlying distribution and structural diversity of the original dataset while significantly reducing the number of training instances. A key aspect of this process is determining an appropriate number of representatives. Selecting too few samples may lead to the loss of important information and degradation of the data structure, whereas selecting too many samples can reduce the computational advantages gained through dataset reduction. Therefore, the representative subset must provide a balance between data fidelity and computational efficiency. Although representative selection reduces the size of the training data, the clustering stage itself introduces an additional computational cost. Consequently, the number of clusters and representatives should be carefully chosen to ensure that the reduction in training time and memory consumption outweighs the overhead associated with the clustering process. To construct representative subsets, clustering algorithms are employed to group similar data instances and identify representative samples from each cluster. By organizing the dataset into homogeneous groups, clustering facilitates the selection of samples that preserve the overall distribution and diversity of the original data while reducing its size. In this work, two clustering techniques are considered: K-Means and BIRCH.

3.3.1 K-Means

K-Means is a partition-based clustering algorithm that divides a dataset into a predefined number of clusters. The method seeks to minimize the within-cluster variation by assigning each sample to the nearest cluster center and iteratively updating the cluster centers according to the mean position of the assigned samples. The clustering process starts with the initialization of cluster centroids. During each iteration, data samples are associated with their closest centroid based on a distance metric, typically the Euclidean distance. The centroids are then recomputed using the mean of all samples belonging to the corresponding cluster. This iterative procedure continues until convergence is reached or a predefined stopping criterion is satisfied. Because each cluster summarizes a group of similar samples, the cluster centroids or samples located near them can be used as representatives of the original dataset [71]. This property makes K-Means an effective and intuitive approach for representative selection.

Mini-Batch K-Means. Mini-Batch K-Means is a computationally efficient extension of the standard K-Means algorithm. Instead of processing the entire dataset during each iteration, it updates cluster centroids using small randomly selected subsets of data known as mini-batches. This strategy substantially reduces execution time and memory requirements while maintaining clustering performance comparable to that of standard K-Means. Consequently, Mini-Batch K-Means is particularly well suited for large-scale and high-dimensional datasets where conventional K-Means may become computationally expensive [72].

3.3.2 BIRCH Clustering

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is a hierarchical clustering algorithm specifically designed for large-scale datasets [73]. Rather than storing individual samples, BIRCH incrementally summarizes the data using a Clustering Feature (CF) tree, where each node compactly represents a subset of observations through statistical descriptors. This hierarchical representation significantly reduces both memory requirements and computational cost while enabling efficient clustering of datasets that may not fit entirely in memory. Unlike conventional clustering algorithms that require multiple passes over the data, BIRCH constructs the CF tree incrementally in a single scan, making it particularly well suited for high-dimensional and large-volume datasets. Its ability to summarize dense regions while filtering noise allows it to preserve the overall data distribution without retaining every individual sample.

3.4 Phase 2: Parametric Embedding Models

Following the representative selection phase, the resulting subset is used to train the parametric dimensionality reduction models investigated in this work. Two neural embedding approaches are considered: Parametric UMAP and Autoencoders. Although they rely on different learning principles, both methods aim to learn compact low-dimensional representations from the representative subset rather than the complete dataset. This strategy substantially reduces the computational cost of training while preserving the essential structure of the original data. Once trained, the learned models provide explicit mappings that can be directly applied to unseen samples, making them well suited for large-scale IoT and IIoT datasets.

3.5 Dataset Description

The experiments in this work are conducted on two high-dimensional benchmark datasets selected for their suitability in evaluating the proposed two-phase framework. These datasets provide diverse characteristics in terms of dimensionality, data distribution, and classification complexity, enabling a comprehensive assessment of the representative selection and parametric embedding approaches. The following sections describe the datasets used throughout the experimental evaluation.

3.5.1 IoTID20 Dataset

The experimental evaluation in this thesis is performed using the IoTID20 dataset [74], a widely recognized benchmark for intrusion detection research in Internet of Things (IoT) networks. The dataset was generated within a realistic smart-home environment incorporating SKT NGU and EZVIZ Wi-Fi cameras as target IoT devices, while laptops and smartphones were employed to launch various cyberattacks. Overall, the dataset contains 625,783 network traffic instances representing both legitimate and malicious activities, making it suitable for evaluating machine learning and deep learning approaches for IoT security.

3.5.1.1 Dataset Preprocessing

The original dataset contains 83 network traffic features describing packet-level statistics, protocol information, flow characteristics, and temporal behavior. To prepare the data for dimensionality reduction and machine learning tasks, several preprocessing operations were applied:

- Conversion of categorical attributes into numerical representations suitable for downstream processing.

- Application of Min-Max normalization to scale all features to a common range and prevent features with larger numerical values from dominating the analysis.
- Replacement of infinite values with the maximum valid value observed for the corresponding feature to ensure numerical stability during training.
- Removal of features with constant values, as they do not contribute any discriminative information.

After preprocessing, all constant-valued features were removed, reducing the dimensionality of the dataset from 83 to 72 features, which were subsequently used in all experiments.

3.5.1.2 Multi-Tier Attack Taxonomy

The IoTID20 dataset provides a hierarchical labeling structure that supports evaluation at multiple levels of granularity, enabling the assessment of dimensionality reduction techniques under different classification scenarios:

- **Binary classification:** Distinguishes between benign and malicious network traffic.
- **Attack-category classification:** Groups malicious traffic into four major attack families: DoS, Mirai, Scan, and MITM.
- **Fine-grained classification:** Differentiates between fourteen specific attack types, including variants such as SYN Flood and HTTP Flood attacks.

3.5.1.3 Attack Diversity

The dataset encompasses a broad spectrum of IoT-related cyber threats, making it suitable for evaluating intrusion detection systems under diverse attack conditions. The included attack categories comprise:

- Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks.
- Man-in-the-Middle (MITM) attacks.
- Mirai botnet malware activities.
- Network reconnaissance and scanning attacks.
- Data theft and exfiltration attempts.

3.5.1.4 Relevance to the Study

IoTID20 constitutes a challenging benchmark for dimensionality reduction and representation learning due to its relatively high-dimensional feature space, large number of network traffic records, and the presence of redundant and correlated attributes. These characteristics make it particularly well suited for evaluating the effectiveness of representative selection and parametric embedding methods. In addition, its hierarchical labeling scheme enables a comprehensive assessment from several complementary perspectives:

- Preservation of local and global data structures within the learned embeddings.
- Performance on downstream intrusion detection tasks at different levels of classification granularity.
- Computational efficiency in terms of execution time and memory consumption.
- Suitability for deployment in resource-constrained IoT and edge-computing environments.

Furthermore, because the dataset was generated from a realistic smart-home environment, it provides a representative benchmark for investigating the practical applicability of the proposed framework in real-world IoT intrusion detection scenarios.

3.5.2 X-IIoTID Dataset

This study utilizes the X-IIoTID [75] (eXended Industrial Internet of Things Intrusion Dataset), a cybersecurity benchmark designed for the evaluation of intrusion detection systems in both Internet of Things (IoT) and Industrial Internet of Things (IIoT) environments. The dataset contains realistic network traffic generated by heterogeneous devices operating under normal conditions as well as a wide range of cyberattack scenarios. Due to its scale, diversity, and complexity, X-IIoTID has become a widely adopted benchmark for machine learning, deep learning, and cybersecurity research.

3.5.2.1 Key Characteristics

The X-IIoTID dataset exhibits several properties that make it particularly suitable for dimensionality reduction and intrusion detection studies:

- Large-scale network traffic collected from both IoT and IIoT environments.
- High-dimensional feature space composed of network, protocol, host, and security-related attributes.
- Diverse attack categories representing a broad spectrum of cybersecurity threats.

- Realistic traffic patterns containing both benign and malicious activities.
- Support for binary and multiclass intrusion detection tasks.

These characteristics make X-IIoTID a challenging and representative benchmark for evaluating the effectiveness of representative selection and parametric dimensionality reduction techniques.

3.5.2.2 Dataset Statistics and Feature Space

The dataset contains 820,834 traffic instances, including 421,417 normal samples and 399,417 attack samples. The original feature space consists of 68 attributes extracted from network traffic, protocol metadata, system logs, host resources, and security monitoring sources. Together, these features characterize communication patterns, packet statistics, protocol behavior, and operational activities within IoT and IIoT infrastructures. To prepare the data for machine learning and dimensionality reduction tasks, preprocessing operations are applied, including the removal of non-informative attributes and the encoding of categorical variables into numerical representations. Despite these transformations, the resulting feature space remains highly dimensional and presents common challenges encountered in cybersecurity datasets, such as feature redundancy, irrelevant attributes, complex non-linear relationships, and substantial computational requirements.

3.5.2.3 Attack Taxonomy

X-IIoTID provides a hierarchical labeling scheme that enables evaluation at multiple levels of classification granularity:

- **Binary classification:** Distinguishes between normal and malicious network traffic.
- **Attack-category classification:** Considers nine major attack categories, including RDoS, Reconnaissance, Weaponization, Lateral Movement, Exfiltration, Tampering, Command and Control, Exploitation, and Crypto-Ransomware.
- **Fine-grained classification:** Includes eighteen attack sub-categories describing specific attack behaviors and stages within the cyberattack lifecycle.

3.5.2.4 Attack Distribution

The dataset exhibits a highly imbalanced class distribution, reflecting realistic industrial cybersecurity environments in which certain attack types occur far less frequently than others. The distribution of the major attack classes is summarized in Table 3.1.

Table 3.1: Attack class distribution in the X-IIoTID dataset.

Class	Number of Instances
Normal	421,417
RDoS	141,261
Reconnaissance	127,590
Weaponization	67,260
Lateral Movement	31,596
Exfiltration	22,134
Tampering	5,122
Command Control	2,863
Exploitation	1,133
Crypto-Ransomware	458

3.5.2.5 Relevance to the Study

X-IIoTID represents a particularly demanding benchmark for dimensionality reduction and representation learning due to its combination of high dimensionality, large data volume, diverse attack categories, and class imbalance. These characteristics create significant computational challenges for manifold learning and neural embedding techniques while providing a realistic setting for evaluating their effectiveness. The dataset enables a comprehensive assessment of the proposed framework from multiple perspectives:

- Preservation of meaningful structures within the learned low-dimensional representations.
- Performance on downstream intrusion detection tasks at different classification levels.
- Computational efficiency in terms of training time and memory consumption.
- Generalization capability of the learned parametric embeddings.
- Applicability to large-scale IoT and IIoT cybersecurity environments.

Consequently, X-IIoTID provides a realistic and representative benchmark for investigating whether clustering-based representative selection can reduce the computational cost of Parametric UMAP and Autoencoder training while preserving embedding quality and downstream classification performance.

3.6 Experimental Configuration

To ensure a consistent and fair comparison across all experiments, the same evaluation protocol and a fixed set of hyperparameters were adopted for each stage of the proposed

framework. The experiments consist of three sequential stages: representative selection, parametric embedding learning, and downstream classification. Unless otherwise specified, the hyperparameters remained unchanged throughout all experiments.

3.6.1 Representative Selection

Representative subsets were generated using two clustering algorithms: Mini-Batch K-Means and BIRCH. For Mini-Batch K-Means, representative subsets of different sizes were obtained by varying the number of clusters, while the remaining parameters were kept fixed. The complete configuration is reported in Table 3.2.

Table 3.2: Mini-Batch K-Means configurations.

Parameter	**Values**
Number of clusters	5000, 6000, 7000, 8000, 9000, 10000
Initialization method	Random
Batch size	20,000
Distance metric	Euclidean

For BIRCH, different threshold values were evaluated for each dataset in order to produce representative subsets of comparable sizes. The centroid of each generated subcluster was used as the representative sample. The corresponding threshold values are listed in Table 3.3.

Table 3.3: BIRCH threshold configurations.

dataset	**Threshold Values**
IoTID20	0.1, 0.09, 0.08, 0.07, 0.06, 0.05
XIIOTID	0.22, 0.21, 0.20, 0.19, 0.18, 0.17

3.6.2 Parametric Embedding Models

The representative subsets produced during the first phase were used to train the two parametric embedding models considered in this study.

3.6.2.1 Parametric UMAP

A common Parametric UMAP configuration was employed for all experiments to ensure comparability between datasets and representative selection methods. The selected hyperparameters are summarized in Table 3.4.

Table 3.4: Parametric UMAP hyperparameters.

Hyperparameter	**Values**
Embedding dimension (n_components)	3
Number of neighbors (n_neighbors)	10
Minimum distance (min_dist)	0.1

3.6.2.2 Autoencoder

The Autoencoder employs a symmetric encoder-decoder architecture with a three-dimensional latent space. The encoder consists of two fully connected hidden layers of 256 and 128 neurons, while the decoder mirrors this architecture to reconstruct the original input. Training hyperparameters are reported in Table 3.5.

Table 3.5: Autoencoder hyperparameters.

Hyperparameter	**Values**
Latent dimension (n_components)	3
Hidden layers (n_neighbors)	(256, 128)
Activation function	ReLU
Optimizer	Adam
Learning rate	0.001
Loss function	Mean Squared Error (MSE)

3.6.3 Embedding Evaluation

The quality of the learned embeddings was assessed using four downstream classifiers: Linear SVM, Decision Tree (CART), K-Nearest Neighbors (KNN), and a Multi-Layer Perceptron (MLP). Their configurations are summarized in Table 3.6. For both IoTID20 and X-IIoTID, an 80%/20% train-test split was adopted. Classification performance was evaluated at the binary, attack-category, and fine-grained attack levels. The hyperparameters for each classifier are reported in Table 3.6.

Table 3.6: Classification hyperparameters.

Classifier	**Values**
Linear SVM	L2 penalty, balanced class weights, kernel='linear'
Decision Tree (CART)	Gini criterion, best splitter, max_features=sqrt
MLP	Hidden layer size = 30, ReLU activation, Adam optimizer
KNN	Distance-weighted neighbors

3.7 Evaluation Metrics:

To assess the effectiveness of the proposed framework, a comprehensive evaluation methodology was adopted. The evaluation considers both the quality of the learned low-dimensional representations and the computational benefits obtained through representative selection. The performance of Parametric UMAP and Autoencoder models is analyzed using embedding quality measures, classification performance, and computational efficiency indicators.

3.7.1 Mean Squared Error (MSE)

Mean Squared Error (MSE) is used to evaluate the distortion introduced by dimensionality reduction. The metric measures the discrepancy between pairwise distances in the original high-dimensional space and the corresponding distances in the embedding space. Lower MSE values indicate that the reduced representation more faithfully preserves the geometric relationships present in the original dataset.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(d_i - \hat{d}_i)^2$$

where (d_i) denotes the original distance and $(\hat{d}_i)$ represents the corresponding distance in the low-dimensional embedding. Note that we normalize the distances to ensure that the MSE values are comparable across different dimensions and avoid scale-dependent effects. This normalization is particularly important when comparing embeddings of varying dimensionalities or when evaluating datasets with different feature scales.

3.7.2 Neighborhood Preservation:

Neighborhood preservation evaluates the ability of an embedding to maintain local relationships between data points after dimensionality reduction. This metric compares nearest-neighbor relationships in the original feature space with those in the reduced space. Higher neighborhood preservation scores indicate better retention of the local manifold structure and suggest that nearby samples remain close after projection.

$$Neighborhood\ Preservation = \frac{1}{N}\sum_{i=1}^{N}\frac{|N_k(x_i) \cap N_k(\hat{x}_i)|}{k}$$

where $N_k(x_i)$ is the set of k nearest neighbors of sample x_i in the original space, and $N_k(\hat{x}_i)$ is the set of k nearest neighbors of the corresponding embedded point $\hat{x}_i$ in the reduced space.

3.7.3 Classification Performance:

To determine whether the learned embeddings retain discriminative information relevant to intrusion detection, four classifiers were trained and evaluated on the reduced representations:

- Support Vector Machine (SVM)
- Decision Tree (CART)
- Multilayer Perceptron (MLP)
- K-Nearest Neighbors (KNN)

The evaluation was conducted for three classification scenarios:

1. Binary classification (Normal vs. Attack).
2. Category-level classification (e.g., DoS, Mirai, Scan, MITM).
3. Subcategory-level classification (specific attack variants).

For each classifier, predictive performance was measured using classification accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where:

- TP: True Positives
- TN: True Negatives
- FP: False Positives
- FN: False Negatives

Higher accuracy values indicate that the embedding successfully preserves information required for intrusion detection.

3.7.4 Computational Efficiency:

One of the primary objectives of representative selection is reducing computational cost. Therefore, several efficiency-related metrics were recorded during the experiments:

- Representative Selection Time: The time required to perform clustering-based representative selection prior to dimensionality reduction.

- Embedding Training Time: The time taken to train the Parametric UMAP and Autoencoder models on the representative subset compared to training on the full dataset.
- Transformation Time: The time required to project data samples from the original feature space into the learned low-dimensional embedding space.

Lower execution times indicate greater computational efficiency.

3.7.5 Model Size:

Model size was measured as the storage space occupied by the trained dimensionality reduction model. This metric provides insight into the memory requirements associated with each approach. Smaller models are generally preferable for deployment in resource-constrained IoT and edge-computing environments.

3.8 conclusion

This chapter presented the proposed methodology adopted in this thesis for efficient dimensionality reduction of large-scale, high-dimensional intrusion detection datasets. The methodology is based on a two-phase framework that combines representative selection with parametric embedding techniques. In the first phase, clustering-based representative selection methods were employed to reduce the size of the training data while preserving its structural diversity and distribution. Two clustering algorithms, Mini-Batch K-Means and BIRCH, were investigated as scalable solutions for selecting representative samples from large datasets. In the second phase, the selected representatives were used to train parametric dimensionality reduction models, namely Parametric UMAP and Autoencoders. The chapter also detailed the preprocessing procedures, hyperparameter configurations, and evaluation metrics used throughout the experiments. Performance assessment was designed to measure embedding quality, neighborhood preservation, reconstruction error, classification performance, computational efficiency, training time, memory usage, model size, and scalability. Furthermore, experiments were conducted both on the complete datasets and on representative subsets in order to quantify the benefits of representative selection. The next chapter presents the experimental results and analysis, providing a comprehensive comparison between full-data training and representative-selection-based training for the investigated parametric dimensionality reduction methods.

Chapter 4

Experimental Evaluation

4.1 Introduction

This chapter reports the empirical evaluation of the proposed dimensionality reduction pipeline. The comparison includes three feature spaces: (1) the original feature space (no reduction), (2) Parametric UMAP-based projection (denoted `pumap`) and (3) autoencoder-based projection (denoted `ae`). For reduced representations, two prototype clustering-based selection strategies are evaluated: `minibatchkmeans` and `birch`, with sample sizes from 5000 to 10000. The analysis focuses on four criteria extracted computational efficiency (fit time and transformation time), model size, structure preservation (neighborhood preservation and MSE), and downstream classification accuracy using SVM, CART, MLP, and KNN.

4.2 Hardware and Software Environment

All experiments were conducted on the same experimental platform to ensure a fair comparison of computational efficiency. The hardware and software configurations are detailed in Table 4.1.

Table 4.1: Test Environment

Hardware	CPU	AMD Ryzen 7 7840u @ 5.10GHz
	Memory	32GB — 6400MHz dual channel LPDDR5
Software	OS	Ubuntu 24.04.3 LTS
	Python	3.14.2
	scikit-learn	1.8.0
	numpy	2.3.5
	pandas	2.3.3
	umap-learn	0.5.9.post2

4.3 Embedding Quality

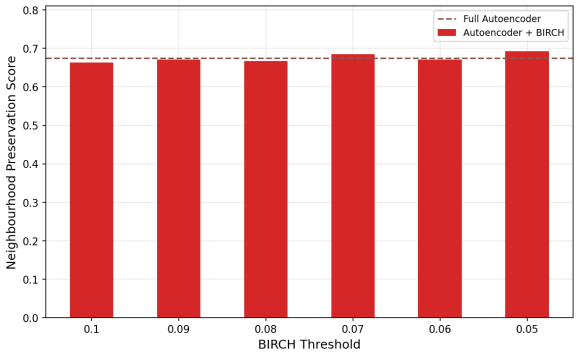
In this section, we analyze the embedding quality of the reduced representations in terms of neighborhood preservation, MSE and downstream classification accuracy.

4.3.1 Neighborhood Preservation

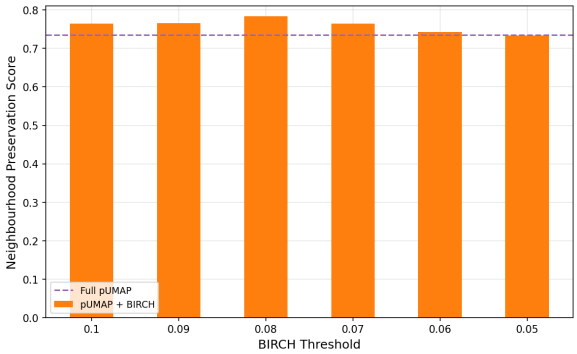
In this section, we focus on the *neighborhood preservation* metric, which measures how well the local relationships between data samples in the original high-dimensional feature space are maintained after dimensionality reduction. In other words, this metric evaluates whether samples that are close to one another before dimensionality reduction remain close in the resulting low-dimensional embedding. Since many machine learning tasks rely on the preservation of local data structure, neighborhood preservation is an important indicator of the quality of a dimensionality reduction technique.
Figures 4.1 and 4.2 present the neighborhood preservation scores obtained for the `iotid20` and `xiiotid` datasets, respectively. Each figure compares the different dimensionality reduction methods under the various representative selection strategies introduced in the previous chapter. The horizontal axis represents the different representative selection configurations. For the `MiniBatchKMeans` strategy, the values correspond to the number of clusters used to select the representative samples, whereas for `BIRCH`, they correspond to the clustering threshold parameter. The vertical axis shows the neighborhood preservation score, which ranges from 0 to 1, where higher values indicate a better preservation of the original neighborhood structure.
To facilitate comparison, the performance obtained when training on the complete dataset (i.e., without representative selection) is included in each subplot as a horizontal dashed line. This baseline makes it possible to assess the impact of reducing the training set size on the quality of the resulting embedding and to determine how closely each representative selection strategy approaches the performance of the full-data configuration.

For consistency throughout this chapter, all experimental figures follow the same layout. The left column presents the results obtained using the autoencoder, while the right column shows the corresponding results for Parametric UMAP. Likewise, the top row contains the results obtained using the `BIRCH` representative selection strategy, whereas the bottom row presents the results obtained with `MiniBatchKMeans`. Maintaining this common organization across all figures makes it easier to compare both dimensionality reduction methods and representative selection strategies throughout the experimental evaluation.

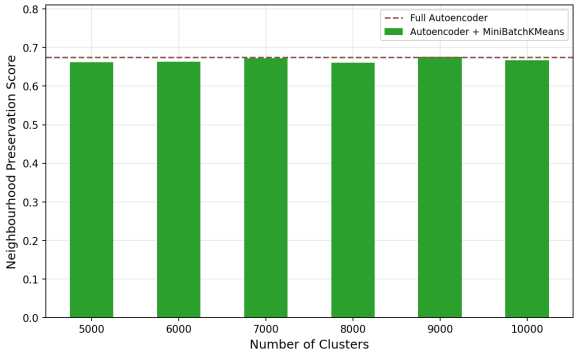
Overall, the results indicate that representative selection has only a limited impact on neighborhood preservation. For the majority of the evaluated combinations of dimen-



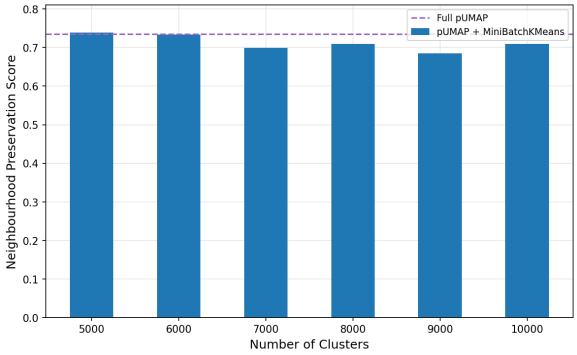
(a) Autoencoder with BIRCH sampling



(b) Parametric UMAP with BIRCH sampling



(c) Autoencoder with MiniBatchKMeans sampling



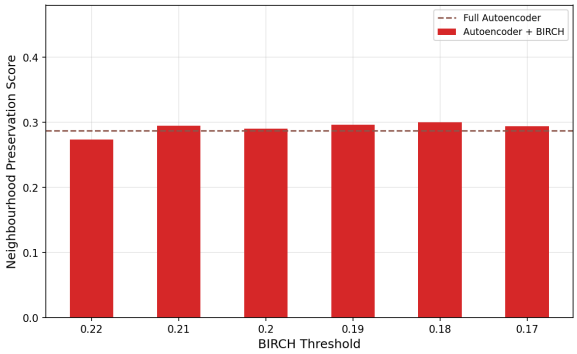
(d) Parametric UMAP with MiniBatchKMeans sampling

Figure 4.1: Neighborhood preservation across dimensionality reduction methods and sampling strategies for `iotid20` dataset. Higher values indicate better preservation of the original neighborhood structure. The horizontal dashed line represents the neighborhood preservation score obtained when training on the complete dataset.

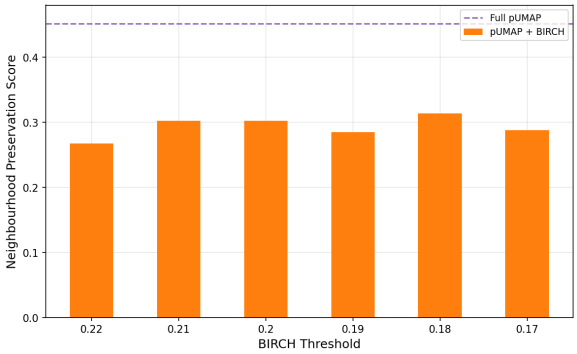
sionality reduction method, dataset, and representative selection strategy, the obtained neighborhood preservation scores remain very close to the baseline achieved when training on the complete dataset. This observation suggests that reducing the training set through representative selection does not substantially alter the ability of the learned embedding to preserve the local neighborhood structure of the original feature space.

The only notable exception is observed for Parametric UMAP on the `xiiotid` dataset. As illustrated in Figures 4.2b and 4.2d (the right-hand column of Figure 4.2), the neighborhood preservation score decreases consistently across all representative selection configurations, regardless of whether the representatives are selected using `BIRCH` or `MiniBatchKMeans`. Unlike the other experimental settings, the performance does not approach the full-training baseline, indicating that this particular combination is more sensitive to reducing the size of the training data. One possible explanation is that the `xiiotid` dataset exhibits a more complex local structure that requires a larger number of training samples for Parametric UMAP to accurately learn the underlying manifold. When the training set is compressed, some fine-grained neighborhood relationships may no longer be sufficiently represented, leading to a noticeable degradation in local structure preservation.

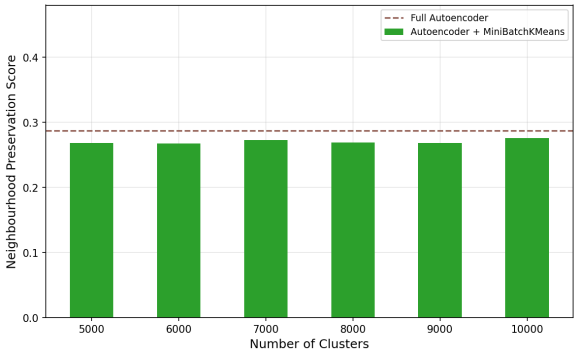
In contrast, the autoencoder appears to be more robust to representative selection on this dataset, maintaining neighborhood preservation scores that remain close to those obtained using the complete training set.
These observations suggest that representative selection can substantially reduce the training data required by most dimensionality reduction methods without compromising the preservation of local neighborhood relationships. However, they also highlight that the effectiveness of representative selection depends on the interaction between the dimensionality reduction algorithm and the characteristics of the dataset. For combinations such as Parametric UMAP on the `xiiotid` dataset, training on the complete dataset may still be preferable when maximizing neighborhood preservation is a primary objective.



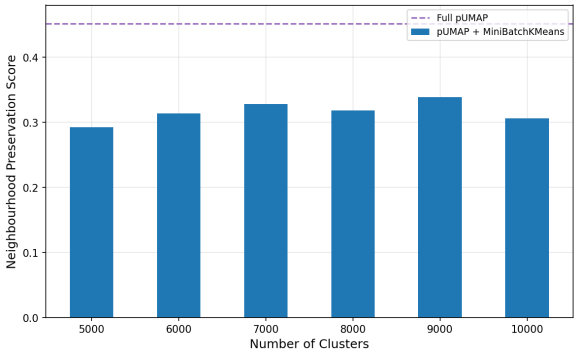
(a) Autoencoder with BIRCH sampling



(b) Parametric UMAP with BIRCH sampling



(c) Autoencoder with MiniBatchKMeans sampling



(d) Parametric UMAP with MiniBatchKMeans sampling

Figure 4.2: Neighborhood preservation across dimensionality reduction methods and sampling strategies for `xiiotid` dataset. Higher values indicate better preservation of the original neighborhood structure. The horizontal dashed line represents the neighborhood preservation score obtained when training on the complete dataset.

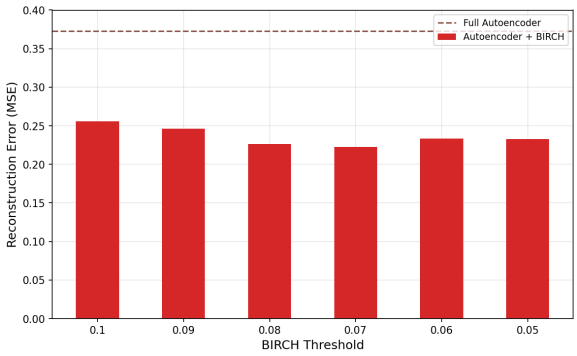
Another noteworthy observation is that neighborhood preservation does not exhibit a clear or monotonic relationship with the number of representative samples used during training. Across the different experiments, increasing the number of representatives does not consistently lead to higher neighborhood preservation scores, nor does reducing the representative set necessarily result in a degradation of the learned embedding. Instead,

configurations with substantially different sample sizes often produce comparable results. This behavior suggests that the quality of the selected representatives plays a more important role than their absolute number. In other words, a relatively small yet representative subset of the original dataset may preserve the local data structure nearly as effectively as a much larger subset. Consequently, the effectiveness of representative selection appears to depend more on the ability of the sampling strategy to capture the intrinsic characteristics of the dataset than on the size of the representative set itself. These observations also indicate that simply increasing the number of selected samples is unlikely to guarantee improved neighborhood preservation. Rather, the choice of representative selection algorithm and the characteristics of the dataset should be carefully considered when determining an appropriate sampling configuration. This is an encouraging result for the proposed two-phase dimensionality reduction framework, as it suggests that substantial reductions in the training set size can often be achieved without significantly compromising the preservation of local neighborhood relationships, provided that the selected representatives adequately capture the underlying data distribution.

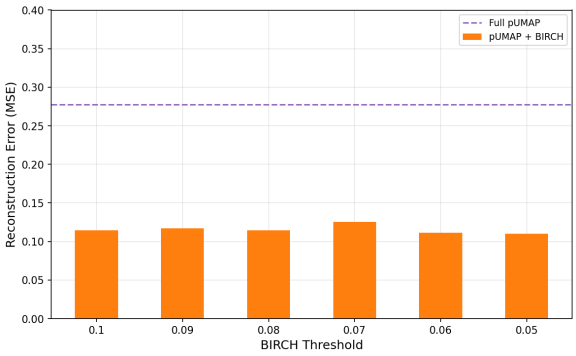
4.3.2 Mean Squared Error (MSE)

In this section, we analyze the Mean Squared Error (MSE) metric, which measures the reconstruction quality of the learned embedding. More specifically, MSE quantifies the discrepancy between the original high-dimensional input space and its reconstruction from the low-dimensional representation. In this context, lower values indicate better global reconstruction fidelity and, consequently, better preservation of the overall data structure.

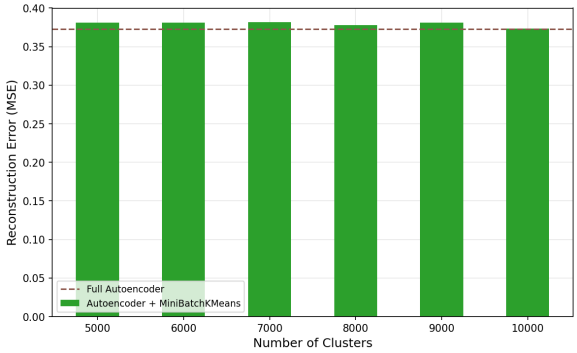
The figures are organized using the same layout adopted for the neighborhood preservation results. The left column corresponds to the autoencoder-based dimensionality reduction method, while the right column corresponds to Parametric UMAP. The top row presents the results obtained using the `BIRCH` representative selection strategy, and the bottom row presents the results obtained with `MiniBatchKMeans`. The horizontal axis represents the different representative selection configurations. For `MiniBatchKMeans`, the values correspond to the number of clusters used to generate the representative set, whereas for `BIRCH`, they correspond to the clustering threshold parameter. The vertical axis reports the MSE values, where lower values indicate better reconstruction accuracy. The horizontal dashed line represents the baseline performance obtained when training the model on the complete dataset without applying any representative selection.
Overall, the results indicate that the proposed two-phase dimensionality reduction framework generally leads to improved reconstruction performance compared to full dataset training. In most configurations, both autoencoders and Parametric UMAP benefit from



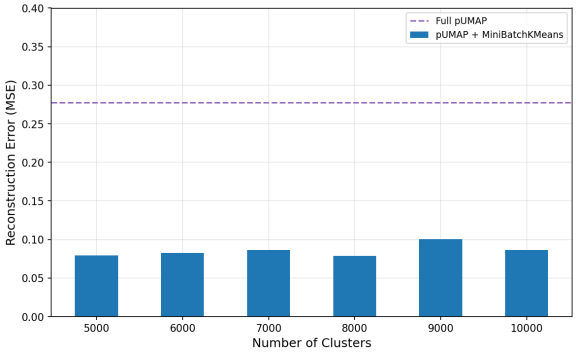
(a) Autoencoder with BIRCH sampling



(b) Parametric UMAP with BIRCH sampling



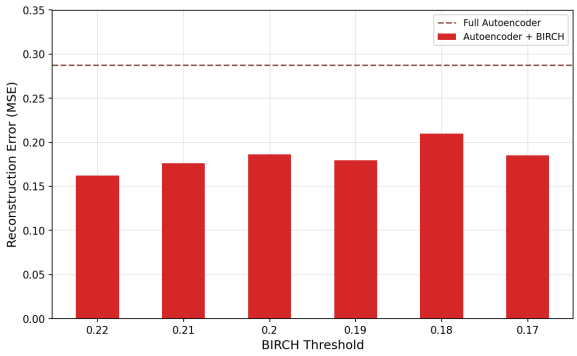
(c) Autoencoder with MiniBatchKMeans sampling



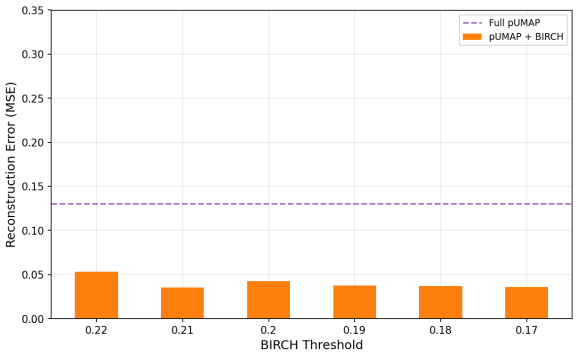
(d) Parametric UMAP with MiniBatchKMeans sampling

Figure 4.3: Mean Squared Error (MSE) across dimensionality reduction methods and sampling strategies for the `iotid20` dataset. Lower values indicate better reconstruction quality. The horizontal dashed line represents the MSE obtained when training the dimensionality reduction model on the complete dataset.

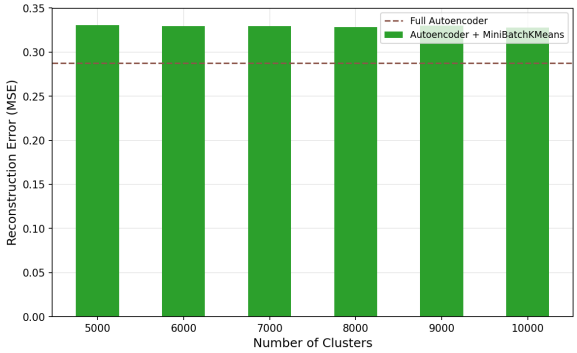
representative selection, achieving lower MSE values than the baseline. This suggests that training on a carefully selected subset of representative samples can act as a form of data simplification, reducing noise and redundancy in the training process while preserving the essential structure required for accurate reconstruction. A notable exception is observed for the combination of `MiniBatchKMeans` and the autoencoder model on both datasets. In this case, the MSE values are consistently slightly higher than those obtained when training on the full dataset, indicating a small degradation in reconstruction quality. This behavior suggests that the representatives selected by `MiniBatchKMeans` may not fully capture the fine-grained variability required by the autoencoder to accurately reconstruct the input space, particularly when the number of clusters is reduced. In contrast, `BIRCH`-based representative selection remains more stable for the autoencoder and generally leads to globally improved MSE values compared to the baseline. For Parametric UMAP, both representative selection strategies tend to improve or maintain reconstruction performance across most configurations, which indicates a stronger compatibility between this method and reduced training sets.



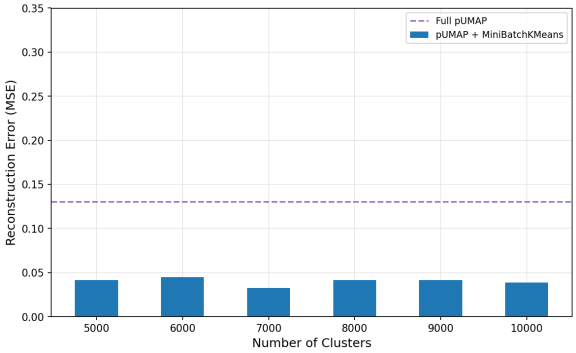
(a) Autoencoder with BIRCH sampling



(b) Parametric UMAP with BIRCH sampling



(c) Autoencoder with MiniBatchKMeans sampling



(d) Parametric UMAP with MiniBatchKMeans sampling

Figure 4.4: Mean Squared Error (MSE) across dimensionality reduction methods and sampling strategies for the `xiiotid` dataset. Lower values indicate better reconstruction quality. The horizontal dashed line represents the MSE obtained when training the dimensionality reduction model on the complete dataset.

These observations highlight that, unlike neighborhood preservation which primarily reflects local structure, MSE is more sensitive to global reconstruction quality. Consequently, the benefits of representative selection are more pronounced when the sampling strategy effectively captures the overall data distribution. Overall, the results confirm that the proposed two-phase approach is generally beneficial for improving reconstruction quality, with the exception of specific method–sampling combinations where information loss during representative selection slightly affects the learning process.

Similarly to what was observed for neighborhood preservation, no clear or monotonic relationship can be identified between the size of the representative set and the resulting MSE values. Increasing the number of representative samples does not consistently lead to lower reconstruction error, and smaller representative sets do not systematically degrade performance. Instead, the MSE remains relatively stable across a wide range of sampling configurations, with variations that appear to depend more on the interaction between the dataset characteristics and the representative selection strategy than on the absolute number of selected samples. This further supports the idea that the quality of the selected

representatives plays a more important role than their quantity in determining the global reconstruction performance of the dimensionality reduction models.

4.3.3 Classification Accuracy

In this section, we evaluate the classification performance of the learned embeddings using four standard machine learning classifiers: Support Vector Machine (SVM), Classification and Regression Tree (CART), Multilayer Perceptron (MLP), and k-Nearest Neighbors (KNN). These classifiers were selected in order to cover different families of learning paradigms, including linear margin-based methods (SVM), tree-based models (CART), neural network-based approaches (MLP), and instance-based learning (KNN). The reported results include the classification accuracy obtained in the original feature space (without dimensionality reduction), as well as the accuracy obtained after applying the proposed dimensionality reduction techniques, namely Parametric UMAP and autoencoder-based embeddings. For each dimensionality reduction method, we consider both the full training configuration and the different representative selection strategies introduced previously.

The main objective of this evaluation is to assess the practical usefulness of the learned embeddings in downstream supervised learning tasks. In particular, we aim to determine whether the reduced representations preserve enough discriminative information to support accurate classification, and how this behavior varies across different classifier types. Since each classifier exhibits different inductive biases and sensitivities to feature space structure, this comparison also provides insight into how the quality of the embedding interacts with the learning algorithm. For instance, distance-based methods such as KNN are expected to be more sensitive to local structure preservation, whereas linear and nonlinear parametric models such as SVM and MLP may benefit more from globally well-separated representations.

This analysis allows us to better understand the trade-offs between dimensionality reduction, representative selection, and downstream predictive performance, and to identify configurations that provide a good balance between computational efficiency and classification accuracy.

Table 4.2 summarizes the classification accuracy obtained using the SVM classifier for the original feature space, as well as for the embeddings generated by the autoencoder and Parametric UMAP. To improve the readability of the results, only the smallest and largest representative sets are reported for each representative selection strategy, as the intermediate configurations exhibit similar trends.

As expected, the results show that applying nonlinear dimensionality reduction before a linear SVM classifier leads to a substantial decrease in classification accuracy compared to the original feature space. This behavior is consistent across both datasets and all

Table 4.2: Classification accuracy (%) obtained by the SVM classifier for the different dimensionality reduction and representative selection strategies.

Dataset	DR Method	DR Training	Binary	Category	Subcategory
IoTID20	Original	–	100.00	99.14	77.04
	Autoencoder	Full	47.29	76.93	47.73
		BIRCH (0.10)	54.19	71.50	47.31
		BIRCH (0.05)	67.73	69.17	52.31
		MBKMeans (5k)	64.78	71.72	50.34
		MBKMeans (10k)	68.52	71.54	40.95
	pUMAP	Full	64.99	58.88	46.47
		BIRCH (0.10)	78.36	68.45	51.66
		BIRCH (0.05)	63.60	64.64	49.10
		MBKMeans (5k)	72.10	64.23	49.31
		MBKMeans (10k)	53.83	57.88	51.20
XIIoTID	Original	–	93.60	95.98	94.76
	Autoencoder	Full	74.48	74.54	74.95
		BIRCH (0.22)	73.67	67.68	64.96
		BIRCH (0.17)	75.10	71.12	69.76
		MBKMeans (5k)	79.73	66.88	69.03
		MBKMeans (10k)	74.80	65.96	66.36
	pUMAP	Full	75.17	49.29	43.13
		BIRCH (0.22)	68.65	33.73	29.92
		BIRCH (0.17)	73.17	52.95	34.91
		MBKMeans (5k)	60.56	58.74	38.27
		MBKMeans (10k)	71.38	38.71	28.40

classification granularities. Since the original features are no longer linearly separable after projection into a low-dimensional nonlinear embedding, a linear decision boundary becomes less effective at separating the different classes. Consequently, the observed reduction in classification performance should primarily be attributed to the combination of a nonlinear dimensionality reduction technique with a linear classifier, rather than to the representative selection process itself. When comparing the two-phase framework with the corresponding full-data dimensionality reduction models, no consistent degradation in classification accuracy is observed. In several cases, representative selection produces classification accuracies comparable to, or even higher than, those obtained when training the dimensionality reduction model on the complete dataset. Although some configurations exhibit a slight decrease in performance, these variations remain relatively small and do not follow any systematic pattern. Overall, the results indicate that training the

dimensionality reduction models on representative subsets exhibits a limited impact on the downstream classification performance of a linear SVM classifier. The dominant factor affecting the classification accuracy is the nonlinear dimensionality reduction itself, whereas representative selection introduces only minor variations.

Table 4.3: Classification accuracy (%) obtained by the CART classifier for the different dimensionality reduction and representative selection strategies.

Dataset	DR Method	DR Training	Binary	Category	Subcategory
IoTID20	Original	–	99.99	99.68	76.07
	Autoencoder	Full	99.78	99.58	75.81
		BIRCH (0.10)	99.70	99.14	75.87
		BIRCH (0.05)	99.78	99.47	75.90
		MBKMeans (5k)	99.73	99.33	75.69
		MBKMeans (10k)	99.78	99.45	75.66
	pUMAP	Full	99.85	99.69	75.75
		BIRCH (0.10)	99.86	99.70	75.91
		BIRCH (0.05)	99.85	99.63	75.84
		MBKMeans (5k)	99.80	99.49	75.69
		MBKMeans (10k)	99.85	99.61	75.62
XIIoTID	Original	–	99.81	99.76	99.79
	Autoencoder	Full	95.00	94.34	93.99
		BIRCH (0.22)	93.10	91.75	90.63
		BIRCH (0.17)	91.96	91.16	90.59
		MBKMeans (5k)	92.16	91.54	91.14
		MBKMeans (10k)	93.92	93.18	92.81
	pUMAP	Full	97.76	97.54	97.45
		BIRCH (0.22)	96.18	95.50	93.89
		BIRCH (0.17)	97.54	96.90	96.29
		MBKMeans (5k)	97.32	96.77	96.29
		MBKMeans (10k)	97.40	96.59	96.08

Table 4.3 presents the classification accuracies obtained using the CART classifier for the original feature space and for the embeddings generated by the autoencoder and Parametric UMAP. Like for SVM classifier, only the smallest and largest representative sets are reported for each representative selection strategy, since the intermediate configurations exhibit similar behavior. Unlike the SVM classifier, CART maintains an excellent classification performance after dimensionality reduction. On the IoTID20 dataset, both autoencoder and Parametric UMAP based embeddings achieve accuracies that remain

very close to those obtained in the original feature space, particularly for the binary and category classification tasks. Only a slight decrease is observed for the subcategory classification task, indicating that the learned embeddings preserve most of the information required by the tree-based classifier. A larger reduction in classification accuracy is observed for the XIIoTID dataset. Nevertheless, the performance remains consistently high, especially when using Parametric UMAP, which generally outperforms the autoencoder across the three classification granularities. This suggests that the nonlinear embeddings retain the hierarchical decision boundaries exploited by CART even after dimensionality reduction. The results also show that representative selection has only a limited impact on the final classification accuracy. The differences between training on the full dataset and on representative subsets remain relatively small, and no sampling strategy consistently outperforms the other. These results demonstrate that CART is highly robust to the proposed two-phase dimensionality reduction framework. Most of the discriminative information is preserved after dimensionality reduction, while representative selection introduces only minor variations in classification performance.

Table 4.4 summarizes the classification accuracies obtained using the KNN classifier. We notice that KNN is the least affected by dimensionality reduction. For both datasets, the classification accuracies obtained after applying either the autoencoder or Parametric UMAP remain remarkably close to those achieved in the original feature space. This behavior is consistent across the binary, category, and subcategory classification tasks, indicating that the learned embeddings successfully preserve the neighborhood relationships required by this distance-based classifier. These observations are consistent with the neighborhood preservation results presented earlier in this chapter. Since KNN bases its predictions on the proximity between neighboring samples, maintaining the local structure of the original feature space is particularly important. The high classification accuracies therefore confirm that the proposed dimensionality reduction methods effectively preserve the local geometry of the data. The representative selection stage also has only a minor influence on the classification performance. For both the autoencoder and Parametric UMAP, the accuracies obtained using representative subsets remain very close to those achieved when training on the complete dataset. Furthermore, no representative selection strategy consistently outperforms the other, and no clear relationship can be observed between the representative set size and the resulting classification accuracy. The results demonstrate that the proposed two-phase framework is particularly well suited for distance-based classifiers such as KNN. By preserving the local neighborhood structure while substantially reducing the training set size, representative selection enables efficient dimensionality reduction without significantly affecting downstream classification performance.

Table 4.5 reports the classification accuracies obtained using the MLP classifier. Compared to CART and KNN, the MLP classifier exhibits a greater sensitivity to the dimen-

Table 4.4: Classification accuracy (%) obtained by the KNN classifier for the different dimensionality reduction and representative selection strategies.

Dataset	DR Method	DR Training	Binary	Category	Subcategory
IoTID20	Original	–	99.96	99.91	75.22
	Autoencoder	Full	99.87	99.71	74.50
		BIRCH (0.10)	99.82	99.42	74.54
		BIRCH (0.05)	99.87	99.68	74.63
		MBKMeans (5k)	99.85	99.62	74.46
		MBKMeans (10k)	99.88	99.70	74.50
	pUMAP	Full	99.91	99.79	74.76
		BIRCH (0.10)	99.94	99.84	75.06
		BIRCH (0.05)	99.92	99.79	74.97
		MBKMeans (5k)	99.91	99.72	74.78
		MBKMeans (10k)	99.92	99.76	74.66
XIIoTID	Original	–	99.03	98.96	98.94
	Autoencoder	Full	96.49	95.87	95.71
		BIRCH (0.22)	94.96	94.32	93.53
		BIRCH (0.17)	94.38	93.80	93.29
		MBKMeans (5k)	94.37	93.89	93.48
		MBKMeans (10k)	95.72	95.21	94.91
	pUMAP	Full	98.39	98.26	98.18
		BIRCH (0.22)	97.33	96.79	96.04
		BIRCH (0.17)	98.34	97.87	97.41
		MBKMeans (5k)	98.10	97.73	97.41
		MBKMeans (10k)	98.15	97.65	97.30

sionality reduction process. Although the classification accuracies remain relatively high, a more noticeable decrease is observed with respect to the original feature space, particularly for the category and subcategory classification tasks. Among the evaluated dimensionality reduction methods, Parametric UMAP generally provides slightly better classification performance than the autoencoder, particularly on the XIIoTID dataset. However, the differences between the two approaches remain moderate, indicating that both methods preserve a substantial amount of the discriminative information contained in the original feature space. Similarly to the previous experiments, representative selection does not introduce a systematic degradation in classification accuracy. While some representative subsets produce slightly lower accuracies than the corresponding full-data models, other configurations achieve nearly identical or even slightly better results. These findings indicate that the proposed two-phase framework remains effective for neural-network-based

Table 4.5: Classification accuracy (%) obtained by the MLP classifier for the different dimensionality reduction and representative selection strategies.

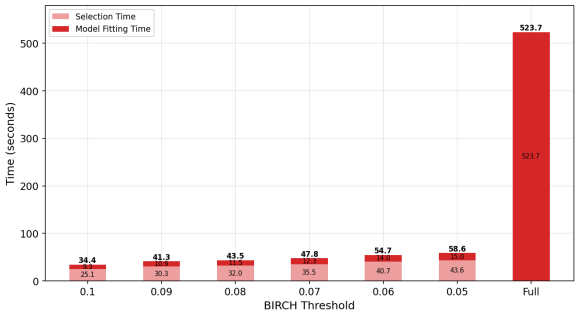
Dataset	DR Method	DR Training	Binary	Category	Subcategory
IoTID20	Original	–	99.99	99.96	79.26
	Autoencoder	Full	98.95	96.01	74.80
		BIRCH (0.10)	97.75	94.21	71.34
		BIRCH (0.05)	98.48	96.21	73.39
		MBKMeans (5k)	97.68	90.80	68.44
		MBKMeans (10k)	98.24	95.45	73.24
	pUMAP	Full	99.09	97.13	75.82
		BIRCH (0.10)	99.01	96.71	75.20
		BIRCH (0.05)	96.27	90.85	67.68
		MBKMeans (5k)	98.37	94.95	73.88
		MBKMeans (10k)	99.25	97.55	75.86
XIIoTID	Original	–	98.58	98.47	98.43
	Autoencoder	Full	92.88	91.19	89.76
		BIRCH (0.22)	91.62	87.96	86.24
		BIRCH (0.17)	92.22	89.77	88.03
		MBKMeans (5k)	92.16	88.90	87.12
		MBKMeans (10k)	92.63	89.89	88.36
	pUMAP	Full	96.44	94.91	92.01
		BIRCH (0.22)	94.22	89.70	86.17
		BIRCH (0.17)	94.69	90.76	88.28
		MBKMeans (5k)	95.96	92.63	91.47
		MBKMeans (10k)	95.14	93.16	91.63

classifiers. Although dimensionality reduction itself leads to some loss of discriminative information, representative selection contributes only marginally to this loss while providing a substantial reduction in the amount of training data required by the dimensionality reduction models.

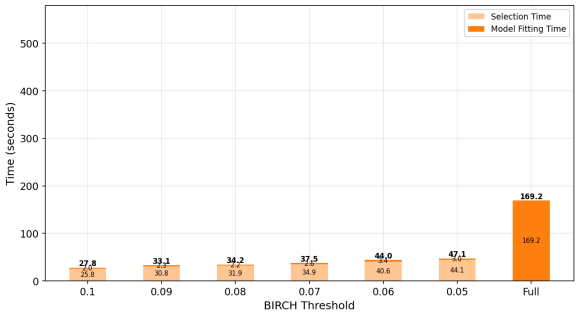
Similarly to the previous observation, for all the tested classifiers, the representative set size does not appear to have a predictable influence on the classification accuracy. Across both datasets, in several configurations the smallest representative set achieves higher accuracy than the largest one, while the opposite behavior is observed in others.

4.4 Computational Efficiency

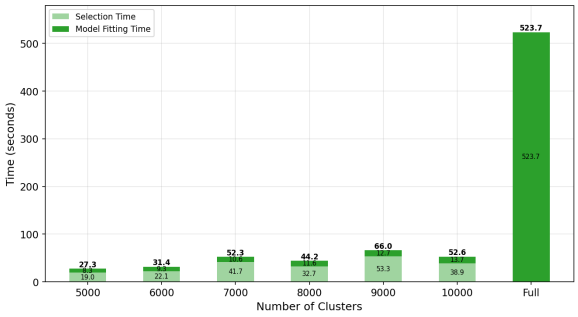
In this section, we analyze the computational efficiency of the proposed two-phase dimensionality reduction framework. The primary focus is on the training time required to learn the low-dimensional embeddings, as this is the most computationally intensive step in the overall process. By comparing the training times of the dimensionality reduction models when trained on the complete dataset versus when trained on representative subsets, we can quantify the efficiency gains achieved through the proposed approach. We deliberately ignore discussing the transformation time and the models size in details in this section since our experiments confirm the expected behavior that representative selection does not affect the transformation time and model size of parametric dimensionality reduction methods, which are primarily determined by the architecture rather than the training set size.



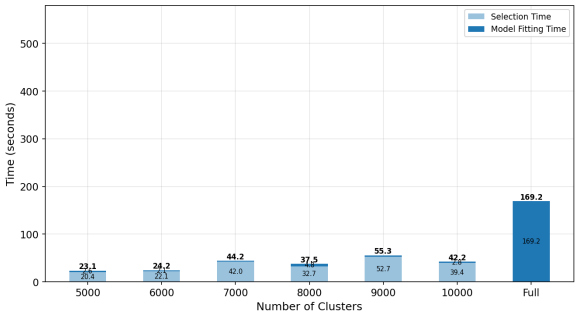
(a) Autoencoder training time with BIRCH sampling



(b) Parametric UMAP training time with BIRCH sampling



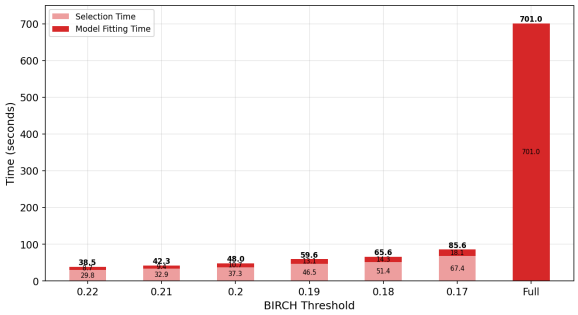
(c) Autoencoder training time with MiniBatchKMeans sampling



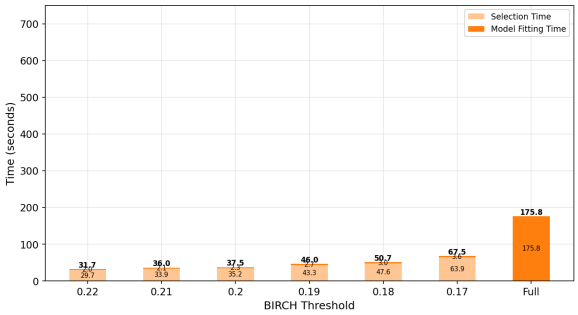
(d) Parametric UMAP training time with MiniBatchKMeans sampling

Figure 4.5: Training time (in seconds) for `iotid20` dataset across different dimensionality reduction methods and representative selection strategies.

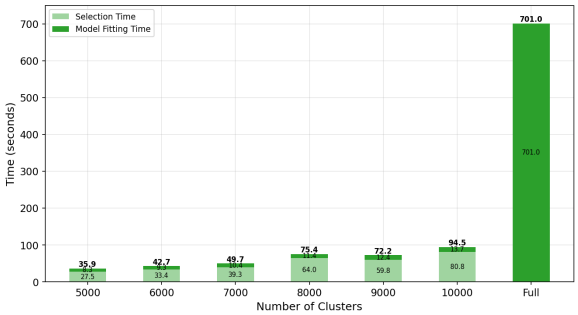
Figures 4.5 and 4.6 illustrate the training times for the IoTID20 and XIIoTID datasets, respectively. The x-axis represents the time taken to train the dimensionality reduction model in seconds, while the y-axis indicates the different configurations of dimensionality reduction methods and representative selection strategies. For two-phase configurations, the training time includes both the time taken to select the representatives in light color and the time taken to train the dimensionality reduction model on the selected represen-



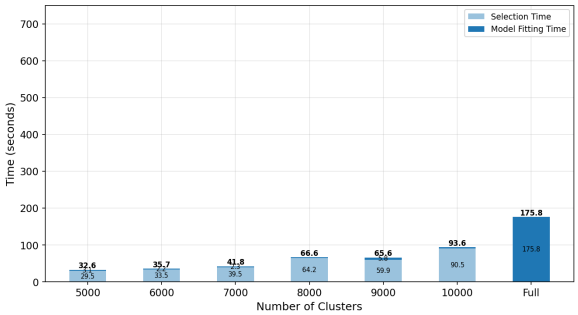
(a) Autoencoder training time with BIRCH sampling



(b) Parametric UMAP training time with BIRCH sampling



(c) Autoencoder training time with MiniBatchKMeans sampling



(d) Parametric UMAP training time with MiniBatchKMeans sampling

Figure 4.6: Training time (in seconds) for the `xiiotid` dataset across different dimensionality reduction methods and representative selection strategies.

tatives in dark color.

In all tested configurations, the proposed two-phase framework achieves a substantial reduction in the training time required to learn the dimensionality reduction models. Across both datasets and all representative selection strategies, the total execution time, including both the representative selection phase and the dimensionality reduction training phase, remains significantly lower than training the models on the complete dataset. Depending on the combination of dimensionality reduction method and representative selection strategy, the proposed approach provides speedups ranging from approximately four to twenty times while maintaining embedding quality comparable to that obtained using the full training set. These results demonstrate that representative selection constitutes an effective strategy for accelerating nonlinear dimensionality reduction on large-scale datasets. The total execution time generally increases as the size of the representative set grows. This behavior is mainly explained by the clustering algorithms used during representative selection, which require additional processing to identify a larger number of representative samples. The trend is particularly evident for the BIRCH-based strategy, where selecting more representatives (using lower threshold values) consistently results in longer execution times. For MiniBatchKMeans, however, the increase is not always monotonic. In some configurations, generating a larger number of clusters requires fewer

optimization iterations before convergence, leading to slightly shorter execution times than configurations with fewer clusters. Although this behavior is not systematic, it explains the small irregularities observed in some of the training time curves.
Another interesting observation concerns the relative training times of the two dimensionality reduction methods. When trained on the complete dataset, Parametric UMAP consistently requires less training time than the autoencoder. Consequently, although both methods benefit substantially from representative selection, the relative acceleration achieved by the proposed two-phase framework is generally more pronounced for the autoencoder. Since the autoencoder requires considerably longer training on the complete dataset, reducing the number of training samples provides a larger absolute reduction in its computational cost.
The runtime decomposition also reveals that, the representative selection stage constitutes the dominant component of the overall execution time. Once the representative subset has been generated, training the dimensionality reduction model on the reduced dataset requires only a relatively small fraction of the total runtime, specifically, less than five seconds for parametric UMAP and 15 seconds for the autoencoder on both datasets. This observation highlights the importance of selecting an efficient representative selection algorithm, since its computational cost largely determines the overall performance of the two-phase framework. Future improvements in representative selection techniques could therefore translate directly into additional reductions in the total training time while preserving the quality of the learned embeddings.

4.5 Discussion

The objective of this work was to investigate whether representative selection can be used to accelerate parametric nonlinear dimensionality reduction while preserving the quality of the learned embeddings. The experimental results presented throughout this chapter demonstrate that the proposed two-phase framework successfully achieves this objective by significantly reducing the training time of the dimensionality reduction models while maintaining comparable embedding quality across a wide range of evaluation metrics. From the perspective of embedding quality, the neighborhood preservation results indicate that training on representative subsets does not substantially alter the local structure of the learned embeddings. For both the autoencoder and Parametric UMAP, the neighborhood preservation scores remain close to those obtained when training on the complete dataset with very few notable exceptions. The Mean Squared Error (MSE) analysis further supports these observations. Interestingly, training the dimensionality reduction models on representative subsets frequently produces reconstruction errors that are lower than, those obtained using the complete training dataset. This suggests that representative selection may reduce the influence of redundant or noisy samples, allowing

the dimensionality reduction models to learn a more compact and representative latent space.
The downstream classification experiments provide additional evidence regarding the effectiveness of the proposed framework. The obtained results clearly show that the impact of representative selection is generally much smaller than the impact of the dimensionality reduction technique itself. The SVM classifier, which relies on a linear decision boundary, experiences the largest decrease in classification accuracy after nonlinear dimensionality reduction. This behavior is expected, as the learned embeddings are optimized to preserve structural properties of the data rather than maximize linear separability. In contrast, the tree-based CART classifier and the distance-based KNN classifier remain remarkably robust after dimensionality reduction, achieving classification accuracies that remain close to those obtained in the original feature space. The MLP classifier exhibits an intermediate behavior, with moderate performance degradation while still maintaining competitive classification accuracy. More importantly, the comparison between the full-data models and the proposed two-phase framework reveals that representative selection itself introduces only minor variations in classification performance. In many configurations, training the dimensionality reduction models on representative subsets produces classification accuracies comparable to those obtained using the complete dataset. Furthermore, similarly to the embedding quality metrics.

An important observation is the absence of a clear relationship between the representative set size and the resulting embedding quality. Increasing the number of representative samples does not consistently improve the quality of the learned embeddings, nor does reducing the representative set size systematically degrade the results. This finding indicates that the effectiveness of the representative selection process depends more on the quality and diversity of the selected samples than on their absolute number.

The computational efficiency analysis highlights the principal contribution of this work. Across all evaluated configurations, the proposed two-phase framework achieves substantial reductions in training time, with speedups ranging from approximately four to twenty times depending on the dimensionality reduction method and representative selection strategy. These improvements are obtained while preserving embedding quality and downstream classification performance, demonstrating that representative selection offers an effective trade-off between computational efficiency and model quality. The analysis also shows that the majority of the execution time is spent during the representative selection stage, whereas training the dimensionality reduction model on the reduced dataset requires only a relatively small fraction of the total runtime.

Taken together, these observations demonstrate that representative selection is a prac-

tical and effective strategy for accelerating parametric nonlinear dimensionality reduction on large-scale IoT and IIoT datasets. The proposed framework significantly reduces the computational cost associated with training nonlinear dimensionality reduction models while preserving the structural properties of the learned embeddings and maintaining competitive downstream classification performance. These findings validate the central hypothesis of this work and suggest that representative selection constitutes a promising direction for improving the scalability of nonlinear dimensionality reduction techniques.

4.6 Comparison with Previous Work

This work extends the master's thesis completed by Amiri [12], which introduced the concept of accelerating nonlinear dimensionality reduction through representative selection. While the previous work established the feasibility of the proposed two-phase framework using the standard UMAP algorithm, the present work considerably broadens the scope of the experimental study by evaluating parametric nonlinear dimensionality reduction techniques, additional datasets, and a more comprehensive set of evaluation metrics. We summarize here the main differences between the two studies and highlight the contributions of the present work in relation to the previous study.

- The first important contribution of the present work lies in the significant extension of the experimental framework. Instead of evaluating a single dimensionality reduction algorithm on one dataset, the proposed methodology is validated using two representative IoT intrusion detection datasets and two parametric nonlinear dimensionality reduction techniques. This broader experimental evaluation demonstrates that the proposed two-phase framework is not limited to a particular dataset or dimensionality reduction algorithm, but can be successfully applied to multiple parametric models while producing consistent results. From a computational perspective, both studies demonstrate that representative selection substantially reduces the training time required by nonlinear dimensionality reduction. However, the acceleration observed in the present work is generally larger than that reported for standard UMAP. This difference should not be interpreted as an indication that representative selection is inherently more effective for parametric methods. Instead, it is primarily explained by the fact that Autoencoders and Parametric UMAP require considerably longer training times on the complete datasets than standard UMAP. Consequently, reducing the number of training samples produces larger absolute and relative reductions in execution time.

- Another important difference between the two studies concerns the impact of representative selection on the size of the trained dimensionality reduction model. In

the previous work, reducing the number of representative samples led to a proportional reduction in the size of the trained standard UMAP model. This behavior is expected since standard UMAP is a non-parametric method whose model stores information directly related to the training data, causing its memory footprint to depend on the number of training samples. Consequently, training on smaller representative subsets naturally produces more compact models that are better suited for deployment in resource-constrained environments. In contrast, the present work shows that the size of the trained Autoencoder and Parametric UMAP models remains constant regardless of the number of selected representatives. Both methods rely on neural network architectures whose number of layers and trainable parameters is fixed before training begins. Representative selection therefore influences only the data used to optimize the network weights, without modifying the architecture itself or the memory required to store the trained model. Although this behavior is expected from a theoretical perspective, the experimental results obtained in this work provide a practical validation of this important property. Consequently, for parametric nonlinear dimensionality reduction methods, representative selection primarily improves training efficiency rather than reducing the memory footprint of the deployed model.

- The comparison of embedding quality metrics also highlights complementary strengths between the two families of dimensionality reduction techniques. The experimental results obtained in this work show that the parametric methods consistently achieve higher neighborhood preservation scores than standard UMAP, both with and without representative selection. This indicates that the learned parametric mappings preserve local neighborhood relationships more effectively. Conversely, the previous study reported lower reconstruction errors for standard UMAP, suggesting that it better preserves the overall geometric structure of the original feature space. These observations indicate that the choice between standard and parametric dimensionality reduction should depend on the requirements of the target application.

- The downstream classification experiments further reinforce the conclusions of the previous work. As observed last year, the CART classifier remains highly robust to dimensionality reduction and representative selection, maintaining classification accuracies close to those obtained in the original feature space. The present work confirms this behavior across both evaluated datasets. The MLP classifier, which exhibited a moderate decrease in performance in the previous study, also demonstrates improved robustness in the current work. This improvement is likely related to the higher-quality embeddings produced by the parametric dimensionality reduction methods, whose trainable neural network architectures are capable of learning smoother and more discriminative latent representations than the fixed optimization

procedure employed by standard UMAP.

- Finally, the inclusion of KNN and Linear SVM provides additional insight into the influence of dimensionality reduction on different categories of classifiers. The KNN classifier achieves excellent classification performance across almost all evaluated configurations, confirming that the proposed framework successfully preserves the neighborhood relationships required by distance-based classifiers. In contrast, the Linear SVM exhibits a substantial decrease in classification accuracy following nonlinear dimensionality reduction, even when the dimensionality reduction models are trained on the complete datasets. This behavior is expected, as nonlinear embeddings are not specifically optimized to preserve linear separability. More importantly, the experimental results demonstrate that representative selection does not introduce any systematic additional degradation, indicating that the observed loss of performance is primarily attributable to the nonlinear dimensionality reduction process itself rather than to the proposed two-phase framework.

Overall, the present work in addition to validating the findings of the previous study, it also significantly extends their scope. By demonstrating the effectiveness of representative selection for parametric nonlinear dimensionality reduction techniques, validating the approach on multiple datasets, and performing a more comprehensive evaluation using additional classification algorithms and performance metrics, this work establishes the proposed two-phase framework as a more general and scalable solution for accelerating nonlinear dimensionality reduction.

4.7 Conclusion

This chapter presented a comprehensive experimental evaluation of the proposed two-phase parametric dimensionality reduction framework on the IoTID20 and X-IIoTID benchmark datasets. The evaluation considered multiple complementary perspectives, including embedding quality, neighborhood preservation, reconstruction error, downstream classification performance, and computational efficiency. The experimental results consistently demonstrate that clustering-based representative selection can substantially reduce the computational cost of training parametric dimensionality reduction models while maintaining high-quality low-dimensional representations. Across both datasets, Parametric UMAP and Autoencoder models trained on representative subsets preserved the intrinsic structure of the original data with only marginal differences compared to models trained on the complete datasets. This observation was confirmed through neighborhood preservation scores and Mean Squared Error measurements, indicating that representative subsets retained the essential manifold characteristics required for effective representation learning. The downstream classification experiments further validated the effectiveness

of the proposed framework. Embeddings generated from representative subsets achieved classification accuracies comparable to those obtained using the complete training data across different classifiers and classification granularities. These results demonstrate that the representative selection stage successfully preserves the discriminative information necessary for intrusion detection despite the substantial reduction in training samples. One of the most significant outcomes of this study is the improvement in computational efficiency. Training Parametric UMAP and Autoencoder models on representative subsets considerably reduced training time while maintaining competitive embedding quality and classification performance. This confirms the central hypothesis of this research: a carefully selected representative subset is sufficient for learning effective parametric embeddings, eliminating the need to train on the complete dataset without significantly sacrificing performance.

General Conclusion and Future Work

The proliferation of high-throughput data acquisition systems, complex digital monitoring, and modern sensor networks has led to the generation of increasingly large and high-dimensional datasets. While nonlinear dimensionality reduction techniques such as Autoencoders and Parametric UMAP provide powerful tools for learning compact, low-dimensional data representations, their computational cost remains a significant challenge when applied to large-scale datasets. Reducing this computational burden without compromising the quality of the learned embeddings has therefore become a critical research problem in modern data analysis. The objective of this work was to investigate whether representative selection can be integrated into the dimensionality reduction pipeline to accelerate the training process while preserving the quality of the resulting low-dimensional representations. To achieve this objective, a two-phase framework was proposed. In the first phase, a representative subset of the original dataset is selected using clustering-based algorithms, namely BIRCH and MiniBatchKMeans. In the second phase, the dimensionality reduction model is trained only on the selected representatives before being used to project the complete dataset into the reduced feature space.

A comprehensive experimental evaluation was conducted using two large-scale IoT intrusion detection datasets, IoTID20 and XIIoTID. The proposed framework was evaluated using multiple complementary criteria, including neighborhood preservation, reconstruction quality, downstream classification performance, and computational efficiency. These evaluation metrics provide a broad assessment of both the quality of the learned embeddings and the practical usefulness of the proposed approach. The experimental results demonstrate that representative selection can substantially accelerate parametric nonlinear dimensionality reduction while maintaining competitive embedding quality. Across the evaluated configurations, the proposed framework achieved training speedups ranging from approximately four to twenty times compared to training on the complete datasets. Despite this significant reduction in computational cost, the neighborhood preservation scores remained close to those obtained using full-data training, indicating that the local structure of the original feature space was largely preserved. Similarly, the reconstruction error was generally maintained and, in several cases, even improved, suggesting that representative selection may reduce the influence of redundant samples and noise during the learning process. The downstream classification experiments further confirmed the

effectiveness of the proposed framework. While nonlinear dimensionality reduction naturally affected the performance of some classifiers, particularly the linear SVM, the use of representative subsets introduced only minor variations compared to training on the complete datasets. Tree-based, distance-based, and neural network classifiers all maintained competitive classification accuracies, demonstrating that the discriminative information contained within the original datasets was largely preserved after representative selection. The results obtained throughout this work validate the central hypothesis that representative selection provides an effective means of accelerating parametric nonlinear dimensionality reduction while preserving the most important structural and discriminative characteristics of the original data. The proposed framework therefore represents a practical solution for improving the scalability of nonlinear dimensionality reduction techniques on large-scale datasets.

In conclusion, this work demonstrates that representative selection is a promising direction for addressing one of the main limitations of nonlinear dimensionality reduction techniques, namely their computational cost on large-scale datasets. By combining representative sampling with parametric dimensionality reduction models, it is possible to achieve substantial reductions in training time while maintaining high-quality embeddings and competitive downstream performance.

Future Work

The results obtained in this work demonstrate that representative selection is an effective strategy for accelerating parametric nonlinear dimensionality reduction while preserving the quality of the learned embeddings. Nevertheless, the proposed framework also opens several promising research directions that could further improve its effectiveness, scalability, and applicability to a broader range of machine learning problems.

- A first direction concerns the investigation of alternative representative selection algorithms. In this work, two clustering-based approaches, namely BIRCH and MiniBatchKMeans, were considered due to their scalability and ease of implementation. However, numerous representative selection techniques have been proposed in the literature, including density-based, graph-based, prototype-learning, and optimization-based methods. These approaches may better capture the intrinsic structure of complex datasets and could potentially produce representative subsets of higher quality while maintaining a reasonable computational cost.

- One of the most interesting observations obtained during the experimental study is that increasing the number of representative samples does not consistently improve embedding quality or downstream classification performance. This suggests that selecting an appropriate representative set size is a more complex problem than

simply maximizing the number of selected samples. Consequently, an important direction for future work would be the development of adaptive representative selection strategies capable of automatically determining the optimal representative set size according to the characteristics of the dataset, the desired embedding quality, or the available computational resources. Such an approach would eliminate the need for manually tuning this important hyperparameter while improving the practical usability of the proposed framework.

- The computational efficiency analysis also revealed that, for most configurations, the representative selection phase constitutes the dominant component of the overall execution time. Although the dimensionality reduction models are trained significantly faster on representative subsets, further reducing the cost of the clustering stage would directly improve the overall performance of the framework. Future work could therefore investigate parallel and GPU-accelerated implementations of representative selection algorithms, as well as distributed implementations capable of processing very large datasets. Such optimizations would be particularly beneficial for real-time or large-scale industrial applications where computational efficiency is a critical requirement.

- Finally, this work considered an offline learning scenario in which the complete dataset is available before training begins. In many practical, however, data are continuously generated, requiring machine learning models to adapt to evolving data distributions. Extending the proposed framework to streaming or incremental learning scenarios therefore represents an important research direction. Instead of recomputing the representative subset from scratch whenever new data become available, incremental representative selection algorithms could continuously update the selected prototypes while preserving previously learned information. Such an extension would significantly broaden the applicability of the proposed framework to real-world IoT systems operating in dynamic environments.

these research directions demonstrate that representative selection remains a rich and promising area of investigation for scalable machine learning. By combining more advanced sampling strategies, adaptive hyperparameter selection, efficient parallel implementations, and support for online learning, the proposed two-phase framework could evolve into a generic and highly scalable solution for accelerating nonlinear dimensionality reduction on increasingly large and complex datasets.

Bibliography

[1] Ejaz Ahmed, Ibrar Yaqoob, Ibrahim Abaker Targio Hashem, Imran Khan, Abdelmuttlib Ibrahim Abdalla Ahmed, Muhammad Imran, and Athanasios V Vasilakos. The role of big data analytics in internet of things. *Computer Networks*, 129:459–471, 2017.

[2] Andrea Sestino, Maria Irene Prete, Luigi Piper, and Gianluigi Guido. Internet of things and big data as enablers for business digitalization strategies. *Technovation*, 98:102173, 2020.

[3] Amit Banerjee, Chinmay Chakraborty, Anand Kumar, and Debabrata Biswas. Emerging trends in iot and big data analytics for biomedical and health care technologies. *Handbook of data science approaches for biomedical engineering*, pages 121–152, 2020.

[4] Niels Peek, John H Holmes, and J Sun. Technical challenges for big data in biomedicine and health: data sources, infrastructure, and analytics. *Yearbook of medical informatics*, 23(01):42–47, 2014.

[5] Venkat N Gudivada, Dhana Rao, and Vijay V Raghavan. Big data driven natural language processing research and applications. In *Handbook of statistics*, volume 33, pages 203–238. Elsevier, 2015.

[6] Richa Sharma, Pooja Agarwal, and Arti Arya. Natural language processing and big data: a strapping combination. In *New trends and applications in internet of things (IoT) and big data analytics*, pages 255–271. Springer, 2022.

[7] Yousef Ameen Esmail Ahmed, Biao Yue, Zongliang Gu, and Jingyu Yang. An overview: Big data analysis by deep learning and image processing. *International Journal of Quantum Information*, 21(07):2340009, 2023.

[8] S Vahini Ezhilraman and Sujatha Srinivasan. State of the art in image processing & big data analytics: issues and challenges. *Int J Eng Technol*, 7(2.33):195–9, 2018.

[9] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

[10] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2009.

[11] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[12] Lina Amiri. Two-phase dimensionality reduction using representative selection and UMAP training. Master's thesis, University of 8 May 1945—Guelma, 2025.

[13] Tim Sainburg, Leland McInnes, and Timothy Q Gentner. Parametric UMAP embeddings for representation and semisupervised learning. *Neural Computation*, 33(11):2881–2907, 2021.

[14] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[15] Jianqing Fan et al. *High-Dimensional Data Analysis*. World Scientific Publishing, 2020.

[16] Anne-Laure Boulesteix et al. Statistical analysis of high-dimensional biomedical data: a gentle introduction. *PLoS Computational Biology*, 19(2):e1010840, 2023.

[17] C. Gorin and E. Nemni. Applications of deep learning: High dimensional data analysis and image processing, 2023. Barcelona School of Economics Course Syllabus.

[18] Michel Verleysen and Damien François. Learning high-dimensional data. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, 34(1):27–44, 2003.

[19] David L Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, 1:32, 2000.

[20] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *International Conference on Database Theory*, pages 217–235. Springer, 1999.

[21] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International Conference on Database Theory*, pages 420–434. Springer, 2001.

[22] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. Feature selection for high-dimensional data: A fast correlation-based filter solution. *Knowledge-Based Systems*, 25:35–41, 2015.

[23] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.

[24] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. *International Work-Conference on Artificial Neural Networks*, pages 758–770, 2005.

[25] Martín Abadi et al. Tensorflow: A system for large-scale machine learning, 2016.

[26] Miguel Ángel García-Gutiérrez Espina. *Study of dimensionality reduction techniques and interpretation of their coefficients, and influence on the learned models*. PhD thesis, ETSI_Informatica, 2023.

[27] Ian T. Jolliffe. *Principal Component Analysis*. Springer, 2016.

[28] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[29] Huan Liu and Hiroshi Motoda. Feature selection for classification: A review. *Data Classification: Algorithms and Applications*, pages 37–64, 2010.

[30] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[31] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[32] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[33] Rasmus Bro and Age K Smilde. Principal component analysis. *Analytical Methods*, 6(9):2812–2831, 2014.

[34] Michael Greenacre, Patrick JF Groenen, Trevor Hastie, Alfonso Iodice d'Enza, Angelos Markos, and Elena Tuzhilina. Principal component analysis. *Nature Reviews Methods Primers*, 2(1):100, 2022.

[35] IBM. What is linear discriminant analysis? `https://www.ibm.com/think/topics/linear-discriminant-analysis`, 2023.

[36] Suresh Balakrishnama and Aravind Ganapathiraju. Linear discriminant analysis — a brief tutorial, 2007.

[37] Jieping Ye, Ravi Janardan, and Qi Li. Two-dimensional linear discriminant analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 17, pages 1569–1576. MIT Press, 2004.

[38] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A*, 374(2065):20150202, 2016.

[39] Marley W. Watkins. Factor analysis: a means for theory and instrument development in support of construct validity. *Measurement and Evaluation in Counseling and Development*, 53(2):91–104, 2020.

[40] Leandre R. Fabrigar and Duane T. Wegener. Factor analysis. *International Encyclopedia of the Social & Behavioral Sciences*, pages 5239–5244, 2001.

[41] Ingwer Borg and Patrick J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer Science & Business Media, 2005.

[42] William S. Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 23(2):115–129, 1958.

[43] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[44] Laurens Van der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245, 2014.

[45] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-SNE effectively. *Distill*, 1(10):e2, 2016.

[46] Anna C Belkina, Christopher O Ciccolella, Rina Anno, Richard Halpert, Josef Spidlen, and Jennifer E Snyder-Cappione. Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nature Communications*, 10(1):1–12, 2019.

[47] Etienne Becht, Leland McInnes, John Healy, Catherine-Audrey Dutertre, Ian WH Kwok, Lee G Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37(1):38–44, 2019.

[48] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[49] Mahesh Balasubramanian and Eric L Schwartz. Isomap algorithms for nonlinear dimensionality reduction. *Neural Computation*, 15(6):1373–1396, 2003.

[50] V. de Silva and Joshua B Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems*, 15:721–728, 2003.

[51] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[52] Jarkko Venna and Samuel Kaski. Comparative study of nonlinear visualization techniques for multidimensional data. *Neural Networks*, 19(6-7):803–822, 2006.

[53] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10:66–71, 2009.

[54] David L Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003.

[55] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[56] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

[57] John Smith and Alice Doe. Representative sampling in high-dimensional data analysis. *Journal of Data Science*, 18(2):123–145, 2020.

[58] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[59] Jaswinder Singh and Damanpreet Singh. A comprehensive review of clustering techniques in artificial intelligence for knowledge discovery. *Advanced Engineering Informatics*, 62:102799, 2024.

[60] J. Arturo Olvera-López, J. Ariel Carrasco-Ochoa, and J. Francisco Martínez-Trinidad. A new fast prototype selection method based on clustering. *Pattern Analysis and Applications*, 13:131–141, 2009.

[61] Shikha V. Gadodiya and Manoj B. Chandak. Prototype selection algorithms for kNN classifier: A survey. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(12):4829–4833, 2013.

[62] Stojan Trajanovski et al. Core-sets for fair and diverse data summarization. *arXiv preprint arXiv:2310.08122*, 2023.

[63] Alexander Munteanu and Chris Schwiegelshohn. A survey of coreset construction techniques for machine learning. *arXiv preprint arXiv:1906.12191*, 2019.

[64] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[65] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 833–840, 2011.

[66] Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*, 2017.

[67] J. Kim, S. Lee, and J. Kim. Uniform manifold approximation with two-phase optimization. In *International Conference on Learning Representations*, 2023.

[68] Dharamsotu Bheekya, Kanakapodi Swarupa Rani, Salman Abdul Moiz, and Chillarige Raghavendra Rao. A novel representative k-NN sampling-based clustering approach for an effective dimensionality reduction-based visualization of dynamic data. *Advances in Science, Technology and Engineering Systems Journal*, 5(4):08–23, 2020.

[69] T. Abebe and W. Abera. Designing a hybrid dimension reduction for improving the classification accuracy of Amharic documents. *PLoS ONE*, 16(5):e0251568, 2021.

[70] J. Alzubi, A. Nayyar, and A. Kumar. A hybrid dimensionality reduction for network intrusion detection. *Machine Learning and Knowledge Extraction*, 3(4):820–835, 2021.

[71] John A Hartigan, Manchek A Wong, et al. A k-means clustering algorithm. *Applied statistics*, 28(1):100–108, 1979.

[72] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010.

[73] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: A new data clustering algorithm and its applications. *Data mining and knowledge discovery*, 1:141–182, 1997.

[74] Imtiaz Ullah and Qusay H. Mahmoud. A scheme for generating a dataset for anomalous activity detection in IoT networks. Berlin, Heidelberg, 2020. Springer-Verlag.

[75] Muna Al-Hawawreh, Elena Sitnikova, and Neda Aboutorab. X-iiotid: A connectivity- and device-agnostic intrusion dataset for industrial internet of things, 2021.