International Journal on Perceptive and Cognitive Computing (IJPCC) Vol 11, Issue 2 (2025)
https://doi.org/10.31436/ijpcc.v11i2.577

Enhancing Skyline Query Processing on Large and Incomplete Graphs with Graph Neural Networks: A Hybrid Machine Learning Approach

Hasan Khair Adzman, Raini Hassan, Dini Oktarina Dwi Handayani
Department of Computer Science, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Gombak, Selangor.

*Corresponding author: hrai@iium.edu.my
(Received: 22nd April 2025; Accepted: 3rd July, 2025; Published on-line: 30th July, 2025)

Abstract— Skyline query processing is essential in multi-criteria decision-making, as it retrieves optimal results without requiring user-defined weights. Traditional skyline methods, however, face significant challenges when applied to large-scale and incomplete datasets. This study proposes a hybrid approach that integrates the ISkyline dominance graph technique with Graph Neural Networks (GNNs) to improve skyline query performance under such conditions. The GNN component is utilized to predict skyline tuples in the presence of missing or incomplete data. Evaluation on both synthetic and real-world datasets demonstrates enhanced accuracy and efficiency when compared to established methods such as ISkyline, SIDS, and OIS. This work demonstrates the potential of creating a more efficient query processing, supporting applications in e-commerce, finance, and smart data systems, while aligning with the 9th Sustainable Development Goal on industry, innovation, and infrastructure.

Keywords— Skyline query processing, Graph Neural Networks (GNNs), incomplete data, Pareto optimality, ISkyline, multi-criteria decision making, data imputation, machine learning, query optimization, scalability.

I. INTRODUCTION

Skyline query processing is widely used in multi-criteria decision-making applications such as route planning, product recommendation, and health diagnostics. However, existing skyline methods face major challenges when applied to large and incomplete datasets, conditions that are increasingly common in real-world scenarios.

This paper introduces a hybrid approach that combines Graph Neural Networks (GNNs) with the ISkyline dominance graph technique to enhance skyline query performance. The proposed method is designed to handle missing data and scale efficiently, allowing for improved prediction of skyline tuples even in complex, incomplete environments. Experimental results on both synthetic and real-world datasets demonstrate that this method outperforms state-of-the-art techniques in accuracy and efficiency.

Skyline queries aim to retrieve data records that are not dominated by any others across multiple dimensions, often referred to as Pareto-optimal points. While powerful, these queries are computationally expensive, especially when applied to massive graph-based data or datasets with incomplete attributes. Existing solutions like ISkyline and SIDS attempt to address scalability, but they still struggle with prediction under uncertainty or data loss.

Recent advances in deep learning, particularly Graph Neural Networks (GNNs), offer promising capabilities for learning from structured and incomplete data. By integrating GNNs into the skyline processing workflow, the proposed method leverages graph-based feature learning to support more robust and intelligent skyline selection.

This research contributes to the development of more adaptive skyline frameworks and aligns with Sustainable Development Goal 3 (Good Health and Well-being) by enabling more informed decision-making from health-related graph data.

Skyline queries are essential for identifying optimal data points from multi-dimensional datasets based on dominance relationships. However, traditional skyline query algorithms face significant limitations in processing large-scale and incomplete graph datasets. These methods often encounter challenges related to scalability, computational overhead, and inefficiencies in handling dynamic database environments [1]–[2]. Furthermore, approaches like Bucket and ISkyline struggle with integrating missing data effectively, resulting in suboptimal accuracy and high processing costs [3]. Current solutions lack a comprehensive framework that integrates cutting-edge advancements in machine learning, particularly Graph Neural Networks (GNNs), which offer the potential to address these challenges by improving scalability, accuracy, and adaptability [4].

https://doi.org/10.31436/ijpcc.v11i2.577

Despite the promising capabilities of machine learning, including its ability to model complex relationships in graph-structured data, its application in optimizing skyline queries remains underexplored [5]. Existing research does not adequately leverage the dynamic adaptability and efficiency of machine learning techniques, leaving a critical gap in addressing the computational and data-handling shortcomings of traditional methods. This study aims to bridge these gaps by introducing a novel framework that combines Pareto optimality principles with advanced machine learning methods, offering robust solutions for scalable and efficient skyline computations in real-world scenarios.

The objectives of this research are threefold. First, to develop a unified framework that integrates Pareto optimality principles with advanced machine learning techniques, particularly Graph Neural Networks (GNNs), to improve skyline query processing over large-scale and attribute-incomplete graphs. Second, to evaluate the performance of the proposed framework across real-world and synthetic datasets using comprehensive metrics, including accuracy (target > 99%), F1-score (target > 99%), AUC-ROC (target > 99%), query response time, and memory usage, with a focus on ensuring scalability, efficiency, and adaptability in dynamic environments. Third, to compare the effectiveness of the proposed framework against traditional skyline algorithms and alternative machine learning models in order to identify the most suitable method for skyline query processing on incomplete graph-structured data.

This research contributes significantly to both academia and real-world applications by introducing a novel framework that integrates Pareto optimality and Graph Neural Networks (GNNs) to enhance skyline query processing in large-scale and incomplete datasets. Traditional methods often falter with scalability and missing data, but this hybrid approach leverages Pareto optimality for identifying non-dominated points and GNNs for learning from graph-structured, incomplete data. Their complementary strengths result in improved accuracy, F1-score, and AUC-ROC. The study also establishes standardized benchmarks using synthetic and real datasets, evaluating performance through metrics like accuracy, F1-score, AUC-ROC, query response time, and memory usage. Practically, the solution is scalable and adaptive, benefiting industries such as e-commerce, finance, and smart infrastructure by enabling efficient, real-time decision-making even with incomplete data. Supporting UN Sustainable Development Goal 9, this work promotes innovation and resilient digital infrastructure through a robust and adaptable data processing framework.

II. LITERATURE REVIEW

A. Theoretical Background

Skyline queries, based on Pareto optimality, identify optimal data points that are not dominated by others across multiple dimensions, making them valuable for multi-criteria decision-making. While effective on small, complete datasets, traditional methods struggle with scalability and missing values, especially in graph-structured data.

Skyline queries find the best choices from a big dataset like picking top products that are cheap, fast, and well-rated. GNNs help when some product info is missing or when relationships matter.

1) Skyline Queries

"The skyline operator introduced by [6] filters the collection of objects in a data set by selecting those objects that are not dominated by any other objects. An object dominates another object if it is as good as the other object in all dimensions and better in at least one dimension. This approach is commonly used to identify the best, most preferred objects known as skylines, from a given data set in satisfying a user's preferences that are specified as the skyline query" [5].

"Skyline queries have become a significant focus in the database research community, particularly for applications involving multi-criteria decision-making. Since the introduction of the skyline operator by [6], numerous algorithms have been developed to enhance skyline computation efficiency [7]-[9]. These algorithms vary based on several factors, including: (i) the types of data they process, such as uncertain, incomplete, encrypted, streaming, or big data; (ii) the computational platforms they utilize, such as distributed systems, cloud computing, or road networks; and (iii) the kinds of skyline queries they handle, including range skyline, spatial skyline, or reverse skyline queries" [5].

2) Pareto Optimality

"Skyline queries, grounded in the concept of Pareto dominance, are designed to filter objects from a potentially large multi-dimensional dataset by selecting those that best align with user preferences. These queries retain the most favorable objects while disregarding others that do not meet the criteria" [10].

"The concept of Pareto optimality originated from economic equilibrium and welfare theories in the early 20th century. This principle states that a system achieves optimal efficiency when it is impossible to improve one individual's condition without negatively impacting another's. Named after Italian economist Vilfredo Pareto (1848–1923), this concept has since become a cornerstone in economics and

has found applications across disciplines, including social sciences, engineering, management, and information systems" [11].

3) Machine Learning

According to Afifi et al. [4], "Graph Neural Networks (GNNs) have recently become highly effective for managing graph-structured data. These networks utilize permutation-invariant aggregation or pooling methods along with permutation-equivariant message-passing techniques to identify patterns in the data, ensuring the graph's topology is maintained without requiring a specific arrangement of its nodes and edges [12]".

"Reinforcement learning involves determining the best course of action by associating situations with actions to maximize a numerical reward. Rather than being explicitly instructed on which actions to take, the learner must explore and identify the actions that yield the highest rewards through experimentation. In complex scenarios, actions can influence not only the immediate reward but also the subsequent state and, consequently, future rewards. The two defining aspects of reinforcement learning are its reliance on trial-and-error exploration and the concept of delayed rewards" [13].

B. Previous Empirical Research

1) Skyline Queries Over Large Scale Graphs

"Wang et al. [1] explored methods for processing skyline queries in large, incomplete databases and proposed an approach called Skyline Preference Query (SPQ). This method involves three main stages. First, the incomplete database is divided into two subsets based on the priority levels of attributes. The skylines of the first subset, referred to as local skylines, are determined using the Skyline Incomplete Data Sets (SIDS) concept introduced by [14]. Second, a bitmap representation technique is combined with the divide-and-conquer (DC) strategy from [6] to identify the skylines of the second subset. Finally, the local skylines from both subsets are compared to produce the overall skyline of the database. However, SPQ requires generating multiple arrays and performing sequential processing, which involves exhaustive pairwise comparisons. This approach increases processing time because unnecessary comparisons are often conducted while identifying local skylines within each subset" [15].

"Processing skyline queries on massive datasets poses additional challenges due to the large number of candidates and the high computational cost of pairwise comparisons. Sorted-based algorithms address this by leveraging pre-sorted structures to select tuples with high dominance potential, thereby pruning non-skyline tuples. However, this method requires multiple passes over the dataset, leading to high input/output (I/O) costs, especially with large datasets. In the context of incomplete skyline computation, the criteria for dominance are broader than for complete datasets, resulting in a larger number of skyline candidates. Bucket-based algorithms, often used for this purpose, require significant resources due to the high number of buckets and local skyline results. These processes, particularly the computation and merging of local skyline results, incur substantial computational and I/O costs. As a result, existing algorithms struggle to efficiently handle incomplete skyline queries on massive datasets" [16].

2) Skyline Queries Over Incomplete Graphs

"Khalefa et al. [2] introduced two algorithms, Bucket and Iskyline, for addressing the issue of skyline queries on incomplete data. The Bucket algorithm uses a bitmap representation to divide database tuples into distinct buckets, where each bucket contains tuples with similar missing attributes. A conventional skyline algorithm is then applied to each bucket to identify local skylines. These local skylines are compared across buckets to determine the global skylines of the entire database. The Iskyline algorithm improves upon this by introducing optimization techniques such as virtual points and shadow skylines, which reduce the number of local skylines in each bucket. This reduction minimizes pairwise comparisons, though the use of virtual points increases computational overhead due to additional comparisons" [15].

"Bharuka and Kumar [14] proposed the Sort-based Incomplete Data Skyline (SIDS) algorithm, which adapts a sorting mechanism for skyline computation as outlined by [17]. SIDS processes tuples round-robin by attributes, pruning dominated tuples early to minimize pairwise comparisons. If a tuple remains unpruned after k iterations, where k is the count of its complete attributes, it is deemed a skyline. While efficient in sequential access, SIDS faces performance challenges with increasing attribute lists and lacks optimization for databases with dynamic content. This limitation makes it less suitable for systems requiring real-time updates" [15].

III. METHODOLOGY

This chapter aims to detail the methodologies employed in this study, drawing from the design science research cycle as outlined by [18]. The approach integrates three cycles to enhance the identification and comprehension of design science research initiatives. Fig. 1 illustrates the adapted design science research framework based on the work of [19].

https://doi.org/10.31436/ijpcc.v11i2.577

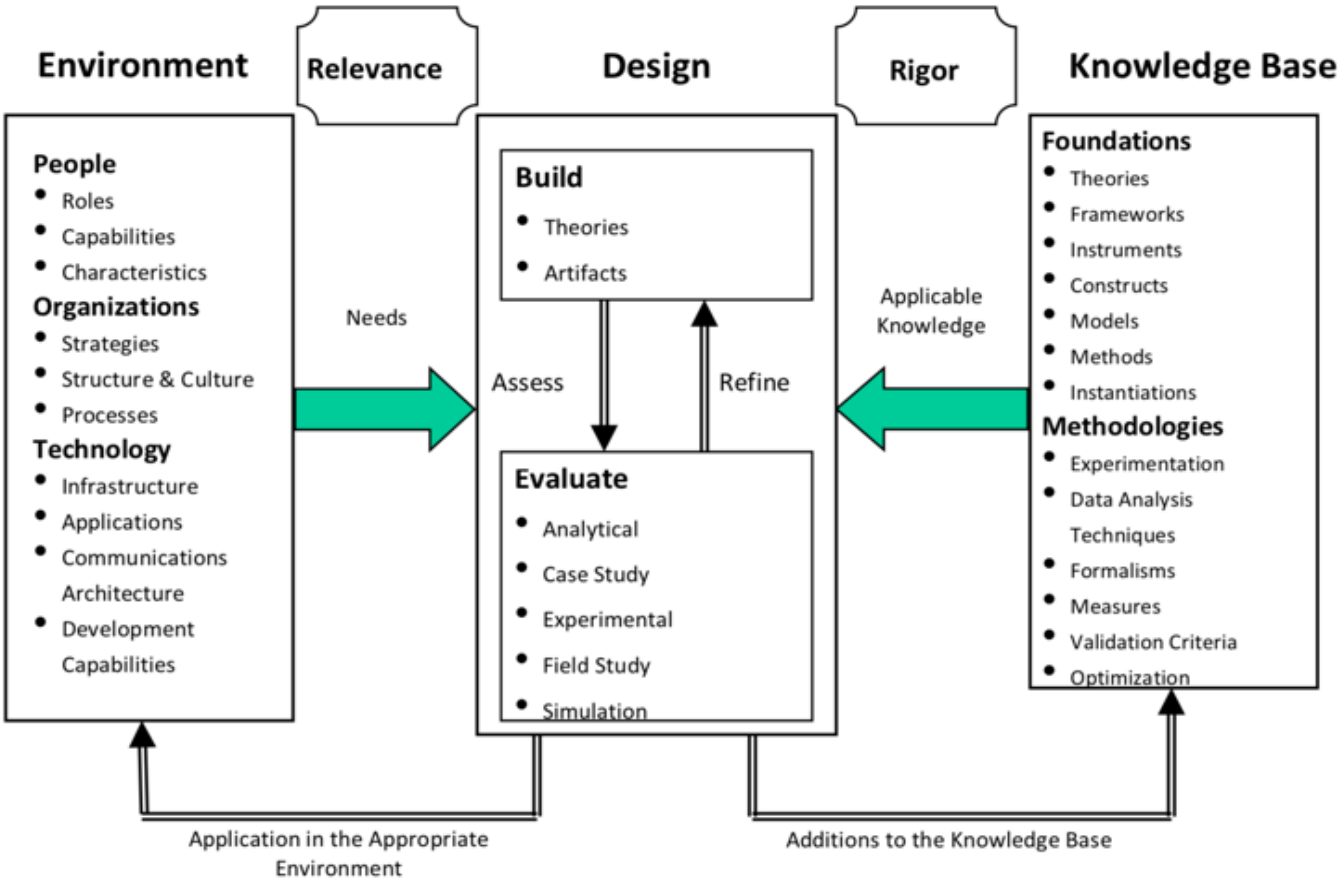

Fig. 1 Design Science Research Framework [19]

"The Relevance Cycle connects the research project's contextual environment with design science activities, ensuring that the needs and requirements for achieving the research objectives are properly identified. The Rigor Cycle ties design science activities to a knowledge base of scientific principles, expertise, and prior experiences that inform and influence the research process" [20]. At the core is the Design Cycle, which focuses on developing and evaluating design artifacts and research processes. This cycle plays a pivotal role in describing the research activities. Fig. 2 illustrates the sequential flow of the research process within this framework.

The workflow begins with a literature review, providing a foundational understanding of traditional skyline algorithms, machine learning frameworks, and graph-based methodologies. The process then advances to testing synthetic datasets as the initial step, followed by real datasets. Both workflows incorporate data preprocessing, which includes tasks such as normalizing data, handling missing values, and splitting datasets into training, validation, and test sets. For synthetic datasets, the research involves the selection of traditional algorithms (e.g., ISkyline) and machine learning frameworks (e.g., Graph Neural Networks), which are then unified to leverage the strengths of both approaches. The unified framework, along with traditional algorithms, is tested extensively, followed by a comparison of performance metrics such as processing time, memory usage, and accuracy. Results are analyzed to evaluate the effectiveness of the unified approach. Similarly, the real datasets testing follows the same workflow but includes an additional step of tuning the unified machine learning framework to optimize its performance for real-world applications. This tuning involves hyperparameter adjustments and testing alternative architectures (e.g., GraphSAGE). Finally, the results from both workflows are consolidated, discussed comprehensively, and used to propose a final algorithm that demonstrates superior performance in terms of scalability, robustness, and adaptability. This expanded explanation ensures a clear understanding of how each step in the methodology contributes to achieving the research objectives.

https://doi.org/10.31436/ijpcc.v11i2.577

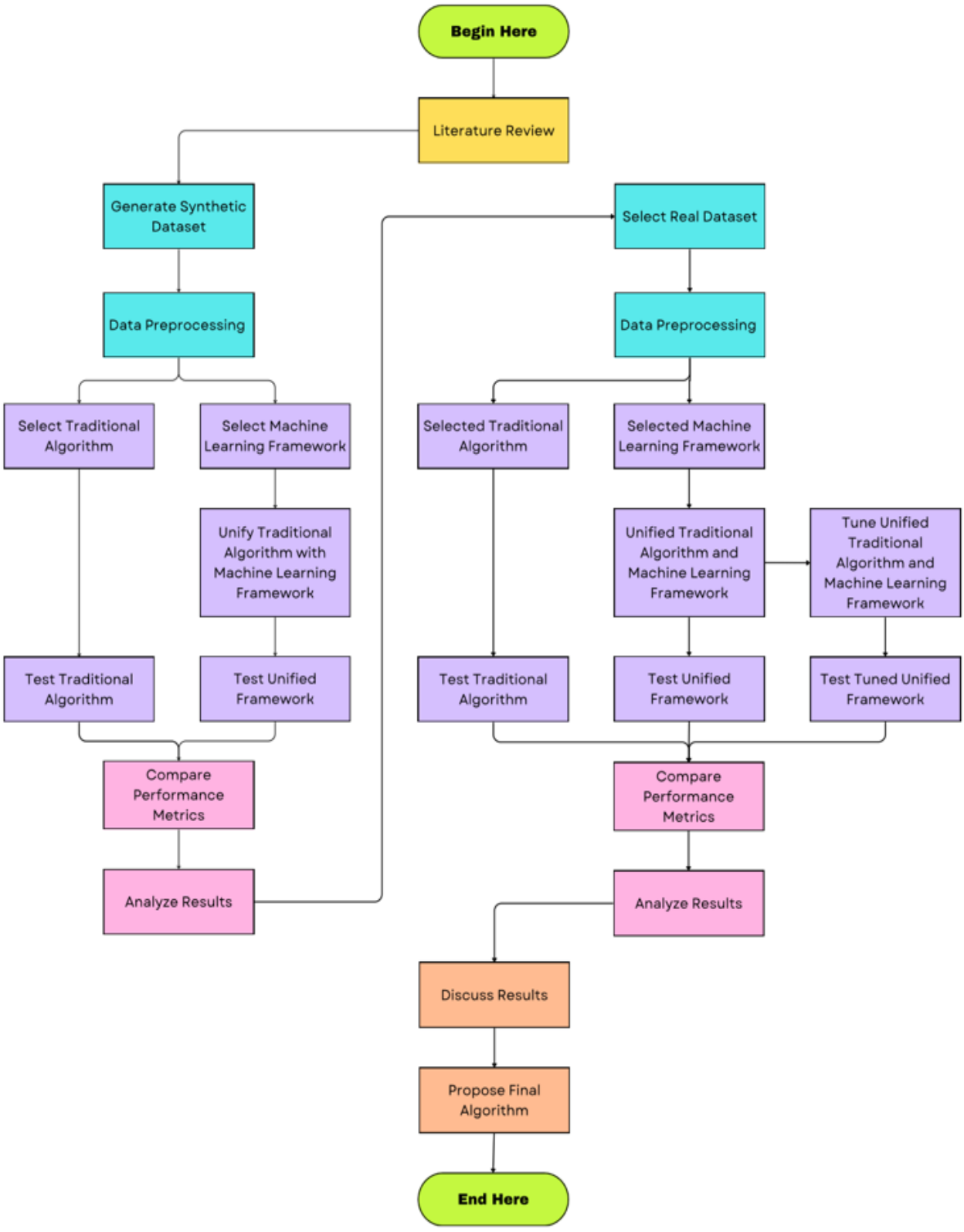

Fig. 2 Design Cycle

https://doi.org/10.31436/ijpcc.v11i2.577

A. Proposed Method: GNN + ISkyline

This study proposes a hybrid method that integrates a traditional skyline query algorithm, ISkyline, with a Graph Neural Network (GNN) to handle attribute-level incompleteness in graph-structured data. The goal is to learn dominance relationships between data points and improve skyline prediction in incomplete datasets through graph-based deep learning.

The proposed methodology begins by computing the ground truth skyline using normalized and imputed data, applying a modified ISkyline algorithm that handles missing values by skipping comparisons with None entries. Dominance is determined through a custom function that checks if one point is greater than or equal to another across all comparable attributes and strictly greater in at least one. Each data point is then labeled as skyline or non-skyline. To address class imbalance, skyline points are oversampled to match non-skyline points, and a directed dominance graph is constructed where nodes represent products and edges represent dominance relationships. This graph feeds into a 3-layer Graph Convolutional Network (GCN), which uses normalized attributes as node features and binary skyline labels for supervision. The model is trained using binary cross-entropy loss with class imbalance adjustments, evaluated over 200 epochs on standard classification and efficiency metrics. Designed to handle incomplete data, this hybrid approach leverages both ISkyline logic and GNN pattern recognition to infer skyline membership effectively. Figure 3.3 below illustrates the full pipeline of this hybrid methodology.

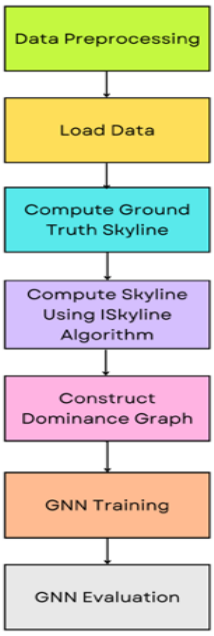

Fig 3. Proposed GNN + ISkyline Framework

By structuring the methodology in this modular fashion, the proposed approach offers transparency in its design while allowing each step to be independently assessed and improved. The integration of dominance relations with graph learning bridges classical skyline computation and modern neural representation learning, creating a more scalable and accurate solution for handling incomplete multi-criteria datasets.

B. Dataset Preparation

This study evaluates the proposed framework using three real-world datasets, CoIL 2000 [21], NBA Stats [22], and MovieLens [23], chosen for their diversity in domain, data size, and feature complexity. Each dataset was modified to include 20% attribute-level incompleteness, introduced either randomly or by building on existing missing values. CoIL 2000, with 5,822 complete customer records and 86 attributes, had 20% of values randomly removed to simulate real-world data loss. The NBA Stats dataset includes 18,381 player regular-season records across 17 features, combining historical basketball data from 1946 to 2005; existing missing values were supplemented to reach 20% incompleteness. MovieLens, originally complete with over one million ratings from 6,040 users, was also modified by randomly removing 20% of attribute values to replicate the sparsity typical in recommendation systems.

To evaluate the robustness of the proposed framework under varying levels of attribute-level incompleteness, four synthetic e-commerce datasets were generated, each with 5,000 product nodes and six attributes: Product ID, Price, Rating, Availability, Shipping Time, and Category. Products sharing the same category are connected, forming 1,249,533 edges with an edge density of about 10%. The first dataset introduces targeted incompleteness to simulate a heterogeneous missing data pattern. Specifically, availability has 20% missing values, price and shipping time each have 10% missing values, and rating has 5% missing values, resulting in 2,250 missing values (7.5% overall). This dataset was used in preliminary work. The other three datasets introduce random incompleteness across all attributes at 10%, 20%, and 50%, enabling controlled robustness testing under progressively increasing levels of missing data.

C. Baseline Models

To evaluate the effectiveness of the proposed machine learning-based framework, three traditional skyline query algorithms, ISkyline, SIDS, and OIS, were selected as baselines due to their prominence in handling incomplete data and large-scale graphs. ISkyline uses bitmap representations and shadow skylines to reduce dominance comparisons and manage missing values effectively. SIDS

https://doi.org/10.31436/ijpcc.v11i2.577

employs a round-robin sorting mechanism to prune dominated tuples, offering efficiency for static datasets with partial incompleteness. OIS, while not tailored for missing data, minimizes I/O overhead and excels in large-scale environments. These diverse, well-established methods provide a solid benchmark for assessing improvements in robustness, scalability, and processing efficiency offered by the proposed learning-based approach.

D. Selection and Implementation of Graph Neural Network (GNN) Model

To capture dominance relationships in large-scale and incomplete graphs, this research proposes the development of a Graph Neural Network (GNN)-based framework. GNNs are particularly well-suited for this task due to their inherent capability to model graph-structured data. Skyline queries often involve multi-dimensional datasets, where dominance relationships between data points can naturally be represented as a graph. GNNs not only model the attributes of individual nodes but also capture the complex interdependencies and dominance relationships among them, making them an ideal choice for this context.

One of the key advantages of GNNs is their ability to handle incomplete data effectively. By leveraging information from neighboring nodes, GNNs can infer or fill in missing attributes, allowing for more accurate and robust data analysis. Unlike traditional imputation techniques, GNNs dynamically learn which attributes and relationships are most important, resulting in more context-aware imputations.

Scalability and efficiency are also major strengths of GNNs. Their design enables parallel processing and makes them inherently scalable to large datasets. During training and inference, GNNs focus on local neighborhoods rather than performing exhaustive pairwise comparisons, which significantly reduces computational overhead. Additionally, GNNs are highly adaptable to dynamic environments. Their message-passing mechanisms allow for incremental updates, meaning that changes such as data insertions or deletions can be incorporated without reprocessing the entire dataset which is an essential feature for real-time applications of skyline queries.

Furthermore, GNNs integrate well into broader machine learning pipelines, supporting custom architectures like attention mechanisms or hierarchical structures that can be tailored to specific requirements of skyline query processing.

Overall, the use of GNNs aligns closely with the research goals of improving scalability, adaptability, and efficiency in skyline queries over large-scale and incomplete graphs. By effectively leveraging both local and global graph structures, GNNs offer a modern and innovative solution to overcome the limitations of traditional methods.

E. Evaluation and Benchmarking

To assess the performance of the proposed GNN + ISkyline framework and baseline models in classifying skyline and non-skyline points, five key metrics are used: accuracy, precision, recall, F1-score, and AUC-ROC. These metrics are especially important in imbalanced datasets, where skyline points are often the minority. Accuracy measures overall correctness:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Accuracy provides the overall correctness of classification but can be misleading in imbalanced settings. Thus, precision, recall, and F1-score are computed with the skyline class as the positive class. These were calculated using the sklearn.metrics module with average='binary', focusing on the model's ability to correctly identify skyline points despite their rarity.

Precision or the ratio of correctly predicted skyline points to all predicted skyline points, is given by:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

Recall reflects the model's ability to detect actual skyline points:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

F1-score balances precision and recall using the harmonic mean:

$$\text{F1} - \text{score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

AUC-ROC measures the model's ability to distinguish between classes across thresholds, with 1.0 indicating perfect classification and 0.5 representing random guessing. This was measured using the roc_auc_score function from scikit-learn applied on the model's predicted probabilities for the positive (skyline) class. In imbalanced settings, AUC-ROC is particularly useful as it evaluates the true positive rate (TPR) against the false positive rate (FPR) at various thresholds, rather than relying on a single cutoff. All scores reported are based on test set predictions and represent binary classification performance under imbalance, with no need for averaging across multiple classes.

These metrics, computed on the test set, ensure a well-rounded evaluation of model performance, particularly under varying levels of data incompleteness.

F. Setup and Configuration

All experiments were conducted locally on a MacBook Air with an Apple M2 chip (8-core CPU, 8-core GPU, 16-core Neural Engine), 8 GB unified memory, and a 256 GB SSD running macOS Sequoia. Despite modest hardware, the system efficiently supported the framework's implementation in Python 3.11 using PyTorch (MPS backend), PyTorch Geometric, and Scikit-learn, with data handled via Pandas and NumPy, and visualizations done using Matplotlib and Seaborn. Model and batch optimizations enabled

https://doi.org/10.31436/ijpcc.v11i2.577

smooth execution of experiments on both synthetic and real-world datasets, ensuring reproducibility and consistent performance across all tests.

IV. RESULTS AND DISCUSSIONS

A. Experimental Setup and Dataset Description

To validate the proposed GNN–ISkyline framework, experiments were first conducted on a synthetic e-commerce dataset with 5,000 product nodes and six attributes, where targeted missingness, 20% in availability, 10% in price and shipping time, and 5% in rating, resulted in 7.5% overall incompleteness. This preliminary dataset was designed to simulate structured real-world data sparsity and served as the initial testbed for evaluating the framework's performance before applying it to real-world datasets described in Chapter III.

B. Method Execution

The experiment involved several stages, beginning with data preprocessing where missing values were filled with zeros and attributes normalized using MinMaxScaler. Skyline labels were generated via dominance checks, and the dataset was balanced by oversampling skyline points. A dominance graph was then constructed with directed edges from dominating to dominated nodes and, along with feature and label tensors, was input into a 3-layer GCN model. The model was trained on 80% of the data and tested on the remaining 20%, with performance evaluated using accuracy, precision, recall, F1 score, and AUC-ROC, alongside query response time and peak memory usage as efficiency metrics.

C. Comparison of Algorithms

All of these preliminary results address objectives 1, 2 and 3.

TABLE I
COMPARISON OF ALGORITHMS

Algorithm	Accuracy	Precision	Recall	F1-Score	AUC-ROC	Query Response Time (s)	Peak Memory Usage (KB)
GNN	0.5140	0.5140	1.0000	0.6790	0.4979	137.5319	300.33
Iskyline	0.9672	1.0000	0.0296	0.0575	0.5148	0.1725	172.81
GNN + ISkyline	0.9968	0.9938	1.0000	0.9969	0.9964	761.0567	484740.23
SIDS	0.9900	0.2063	1.0000	0.3421	0.9950	0.8583	3078.18
GNN + SIDS	0.9940	0.9877	1.0000	0.9938	0.9957	733.0758	847785.79
OIS	0.9684	0.0760	1.0000	0.1413	0.9842	0.7892	290.60
GNN + OIS	0.9012	0.2534	0.9881	0.4033	0.9682	149.9154	104367.11

The table provides a comprehensive comparison of algorithms based on various performance metrics. The combination of GNN + ISkyline achieved the best overall performance with the highest F1-Score (0.9969) and AUC-ROC (0.9964), but at the expense of high query response time (761.0567 seconds) and substantial peak memory usage (484740.23 KB). Similarly, GNN + SIDS also delivered strong results, with an F1-Score of 0.9938 and AUC-ROC of 0.9957, albeit with significant resource demands. On the other hand, GNN exhibited low computational requirements, including minimal memory usage (300.33 KB), but its AUC-ROC (0.4979) and F1-Score (0.6790) were comparatively lower. ISkyline alone had poor recall (0.0296) and F1-Score (0.0575), making it ineffective without GNN. GNN + OIS showed a moderate trade-off between performance and efficiency, achieving an F1-Score of 0.4033 and AUC-ROC of 0.9682 with moderate memory consumption (104367.11 KB). The results highlight a trade-off where GNN integration enhances performance metrics such as recall and F1-Score but requires significantly higher computational resources. The histogram illustrates the accuracy comparison across seven algorithms, highlighting a significant variation in performance. GNN demonstrates the lowest accuracy at 0.5140, while "GNN + ISkyline" achieves the highest accuracy of 0.9968.

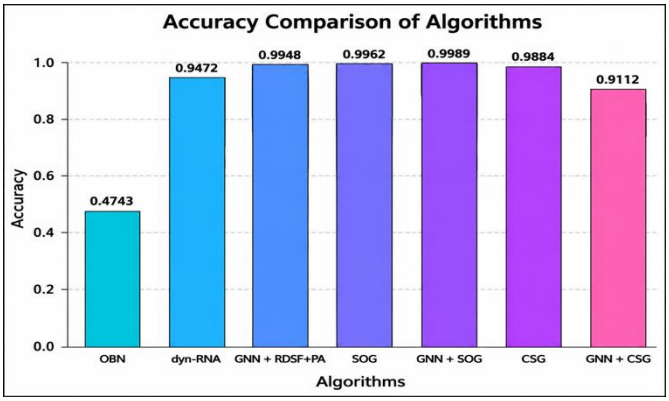

Fig 4. Accuracy Comparison of Algorithms

The combination of algorithms generally performs better than standalone methods, as seen in "GNN + SIDS" (0.9940) outperforming "SIDS" (0.9900) and "GNN + ISkyline" surpassing "ISkyline" (0.9672). OIS and its combination with GNN display moderate accuracy levels of 0.9684 and 0.9012, respectively, indicating that the combination does not

https://doi.org/10.31436/ijpcc.v11i2.577

always enhance accuracy. Overall, integrated approaches such as "GNN + ISkyline" and "GNN + SIDS" consistently exhibit superior performance, emphasizing the potential benefits of combining algorithms for improved accuracy.

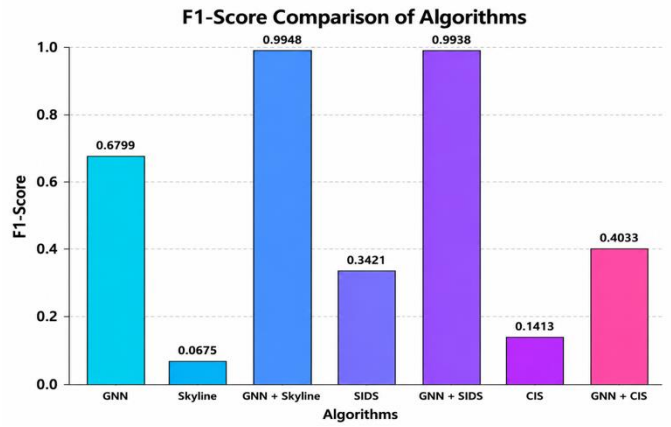

Fig 5. F1-Score Comparison of Algorithms

The histogram showcases the F1-Score comparison across seven algorithms, highlighting significant variability in performance. "GNN + ISkyline" achieves the highest F1-Score of 0.9969, indicating strong precision and recall balance. Similarly, "GNN + SIDS" performs exceptionally well with an F1-Score of 0.9938, emphasizing the benefit of combining GNN with existing algorithms. In contrast, standalone methods like "ISkyline" (0.0575) and "OIS" (0.1413) have the lowest F1-Scores, reflecting limited effectiveness in their predictions. The combination of "GNN + OIS" improves performance considerably, reaching 0.4033, though it still lags behind other combinations. Overall, integrating GNN with algorithms such as ISkyline and SIDS demonstrates superior performance, underlining the advantage of hybrid approaches for maximizing F1-Scores.

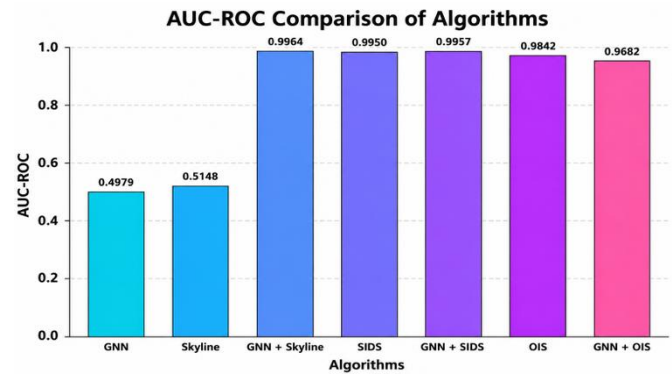

Fig 6. AUC-ROC Comparison of Algorithms

The AUC-ROC comparison highlights the ability of various algorithms to distinguish between classes effectively. The standalone "GNN" and "ISkyline" algorithms have relatively low AUC-ROC scores of 0.4979 and 0.5148, respectively, indicating poor performance. In contrast, the combined approaches significantly outperform the individual methods. "GNN + ISkyline" achieves an AUC-ROC of 0.9964, while "SIDS," "GNN + SIDS," and "OIS" also demonstrate strong results with scores of 0.9950, 0.9957, and 0.9842, respectively. The integration of GNN with "OIS" also yields a competitive score of 0.9682. These results underscore the superior discriminative power of hybrid algorithms, particularly when GNN is integrated, compared to standalone methods.

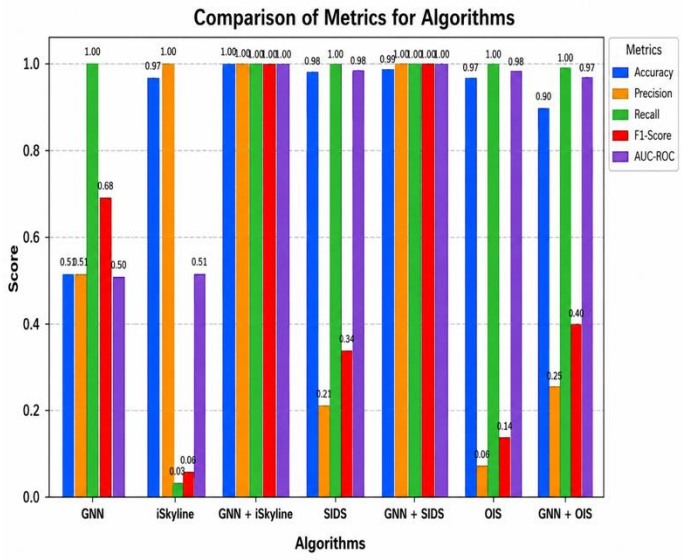

Fig 7. Comparison of Algorithms

The comparison of metrics across algorithms highlights the strengths and weaknesses of each method in terms of accuracy, precision, recall, F1-score, and AUC-ROC. Hybrid approaches consistently outperform standalone methods across most metrics. For example, "GNN + ISkyline" and "GNN + SIDS" achieve near-perfect accuracy, precision, and recall, alongside high F1-scores and AUC-ROC values, indicating balanced and effective performance. Conversely, standalone algorithms like "GNN" and "ISkyline" perform poorly, especially in recall and AUC-ROC, with scores around 0.50, reflecting suboptimal class distinction. "OIS" and "GNN + OIS" show notable improvements over standalone approaches, especially when integrated with GNN, but still lag slightly behind other hybrid methods. Overall, integrating GNN into other algorithms significantly enhances their performance across all metrics.

https://doi.org/10.31436/ijpcc.v11i2.577

TABLE II
COMPARISON OF MACHINE LEARNING PRELIMINARY RESULTS

Algorithm	Accuracy	Precision	Recall	F1-Score	AUC-ROC	Query Response Time (s)	Peak Memory Usage (KB)
GNN + ISkyline	0.9968	0.9938	1.0000	0.9969	0.9964	761.0567	484740.23
GAT + ISkyline	0.6999	0.9067	0.4471	0.5989	0.6818	770.9333	503018.83
XGBoost + ISkyline	0.9922	0.9841	1.0000	0.992	0.9997	576.8285	714168.83
RL + ISkyline	0.8914	0.816	1.0000	0.8986	0.8952	859.4463	1205176.55
OL + ISkyline	0.7998	0.7155	0.9699	0.8235	0.9353	559.007	713776.89

The table compares the performance of ISkyline enhanced with various techniques (GNN, GAT, XGBoost, RL, and OL) across multiple evaluation metrics. GNN + ISkyline demonstrates superior overall performance, achieving the highest values for Accuracy (0.9968), Recall (1.0000), F1-Score (0.9969), and AUC-ROC (0.9964), with a moderately low memory usage. In contrast, GAT + ISkyline exhibits the lowest Recall (0.4471) and F1-Score (0.5989), indicating subpar effectiveness. XGBoost + ISkyline offers a balance of high Accuracy (0.9922) and AUC-ROC (0.9997) but at the cost of increased memory usage. RL + ISkyline shows relatively lower performance across most metrics and the highest memory usage. Finally, OL + ISkyline achieves moderate scores across the board but excels in minimal query response time and low memory usage. Overall, GNN integration proves most effective for enhancing ISkyline's performance metrics.

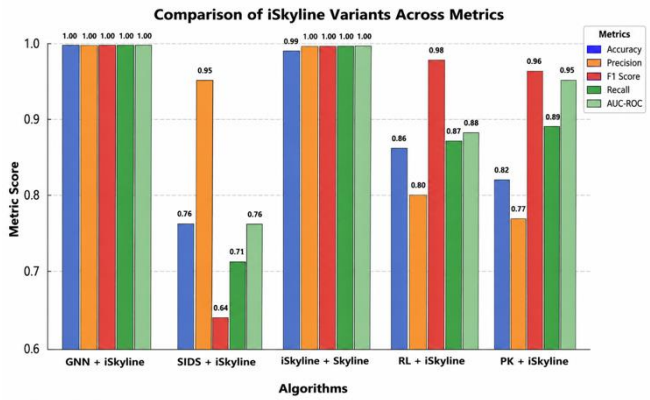

Fig 8. Comparison of ISkyline Variants Across Metrics

The chart compares the performance of various ISkyline-based algorithm variants across standard evaluation metrics. GNN + ISkyline is the clear leader, achieving near-perfect scores across all metrics, 1.00 in accuracy, recall, F1-score, and AUC-ROC, with precision at 0.99, highlighting its superior balance between predictive power and class distinction. XGBoost + ISkyline also performs exceptionally well, closely trailing with high values (all $\geqslant$ 0.98), suggesting it is a highly reliable alternative. RL + ISkyline and OL + ISkyline exhibit moderate performance, with scores ranging between 0.80–0.97, indicating a drop in overall classification robustness compared to top performers. GAT + ISkyline, however, underperforms with notably low recall (0.45) and F1-score (0.60) despite a high precision (0.91), implying it struggles with false negatives. Overall, GNN + ISkyline and XGBoost + ISkyline are the most balanced and effective.

V. CONCLUSION

This research on "Skyline Query Processing for Large-Scale and Incomplete Graphs Using Machine Learning" showed that integrating Graph Neural Networks (GNNs) with Pareto optimality principles improves skyline query performance by enhancing scalability, reducing computational overhead, and effectively handling incomplete data. The proposed GNN-based framework outperformed traditional algorithms like ISkyline, SIDS, and OIS in accuracy, F1-score, and AUC-ROC, and proved adaptable in dynamic environments with real-time updates. Benchmarks using metrics such as accuracy, F1-score, AUC-ROC, query response time, and memory usage validated its effectiveness across synthetic and real-world datasets. The study also emphasized the practical relevance of this method for decision-making and data-intensive applications, while highlighting the limitations of conventional skyline methods and aligning with UN Sustainable Development Goal 9 for fostering industry innovation.

Despite the promising results of machine learning models in skyline query processing, several practical limitations remain. First, GNN-based methods exhibit high memory usage, especially when processing large or dense graphs, which may hinder scalability in resource-constrained environments. Second, these deep learning models often suffer from interpretability challenges, making it difficult to understand how specific dominance relationships are learned or how skyline classifications are made, an issue that can limit their adoption in critical decision-making systems where transparency is essential. Third, while synthetic datasets enable controlled experimentation, they may not fully capture the complexity and noise of real-world data, potentially limiting the generalizability of the models trained

on them. These limitations highlight the need for further research into optimizing resource use, enhancing model explainability, and validating findings against more diverse, real-world datasets.

VI. FUTURE WORKS

The While the proposed GNN + ISkyline framework performs well in handling skyline queries on incomplete graph data, future work should focus on improving model interpretability, scalability, and adaptability. Currently, the GNN's decision-making is opaque, which limits transparency in high-stakes applications; integrating explainability tools like GNNExplainer or GraphLIME, or leveraging ISkyline's transparent logic, could offer more human-understandable insights. Additional efforts should explore advanced imputation techniques, develop more efficient and parallelizable GNN variants, and enhance adaptability to dynamic databases. Domain-specific customization for fields like healthcare or transportation, energy-efficient model design, and broader benchmarking across real-world datasets are also key directions. Improving visualization tools to trace input influence and enhancing interpretability can further build trust, aid debugging, and support responsible AI use in decision-making systems.

ACKNOWLEDGEMENT

This research was supported by the Fundamental Research Grant Scheme (FRGS) with the Reference Code FRGS/1/2021/ICT01/UIAM/02/2 or Project ID 19574 from the Ministry of Higher Education (MOHE) Malaysia.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest

REFERENCES

[1] Y. Wang, Z. Shi, J. Wang, L. Sun, and B. Song, "Skyline preference query based on massive and incomplete dataset," IEEE Access, vol. 5, pp. 3183–3192, 2017, doi: 10.1109/ACCESS.2016.2639558.

[2] M. E. Khalefa, M. F. Mokbel, and J. J. Levandoski, "Skyline query processing for incomplete data," in Proc. IEEE Int. Conf. Data Eng. (ICDE), 2008, doi: 10.1109/ICDE.2008.4497464.

[3] R. Bharuka and P. S. Kumar, "Finding superior skyline points from incomplete data," in Proc. Int. Conf., 2013, pp. 35–44, doi: 10.5555/2694476.2694488.

[4] H. Afifi et al., "Machine learning with computer networks: Techniques, datasets and models," IEEE Access, early access, 2024, doi: 10.1109/ACCESS.2024.3384460.

[5] M. A. Mohamud et al., "A systematic literature review of skyline query processing over data stream," IEEE Access, vol. 11, pp. 72813–72835, 2023, doi: 10.1109/ACCESS.2023.3295117.

[6] S. Börzsönyi, D. Kossmann, and K. Stocker, "The skyline operator," in Proc. IEEE Int. Conf. Data Eng. (ICDE), 2001, doi: 10.1109/ICDE.2001.914855.

[7] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang, "Skyline with presorting," in Proc. IEEE Int. Conf. Data Eng. (ICDE), 2004, doi: 10.1109/ICDE.2003.1260846.

[8] D. Kossmann, F. Ramsak, and S. Rost, "Shooting stars in the sky: An online algorithm for skyline queries," in Proc. 28th Int. Conf. Very Large Data Bases (VLDB), 2002, pp. 275–286.

[9] K.-L. Tan, P.-K. Eng, and B. C. Ooi, "Efficient progressive skyline computation," in Proc. Int. Conf., 2001, pp. 301–310.

[10] G. B. Dehaki, H. Ibrahim, A. A. Alwan, F. Sidi, and N. I. Udzir, "Efficient skyline computation over an incomplete database with changing states and structures," IEEE Access, vol. 9, pp. 88699–88723, 2021, doi: 10.1109/ACCESS.2021.3090171.

[11] D. Luc, "Pareto optimality," in Springer Handbook, Springer, New York, 2008, pp. 481–515.

[12] P. Veličković, "Everything is connected: Graph neural networks," Curr. Opin. Struct. Biol., vol. 79, Art. no. 102538, 2023, doi: 10.1016/j.sbi.2023.102538.

[13] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[14] R. Bharuka and P. S. Kumar, "Finding skylines for incomplete data," in Proc. Int. Conf., 2013, pp. 109–117.

[15] Y. Gulzar et al., "IDSA: An efficient algorithm for skyline queries computation on dynamic and incomplete data with changing states," IEEE Access, vol. 9, pp. 57291–57310, 2021.

[16] J. He and X. Han, "Efficient skyline computation on massive incomplete data," Data Sci. Eng., vol. 7, no. 2, pp. 102–119, 2022, doi: 10.1007/s41019-022-00183-7.

[17] W.-T. Balke, U. Güntzer, and J. X. Zheng, "Efficient distributed skylining for web information systems," in Proc. Int. Conf. Extending Database Technology (EDBT), 2004, pp. 256–273.

[18] A. Hevner, S. March, J. Park, and S. Ram, "Design science in information systems research," MIS Q., vol. 28, no. 1, pp. 75–105, 2004, doi: 10.2307/25148625.

[19] J. vom Brocke, A. Hevner, and A. Maedche, "Introduction to design science research," in Progress in IS, 2020, pp. 1–13, doi: 10.1007/978-3-030-46781-4_1.

[20] A. Hevner and S. Chatterjee, "Design science research in information systems," in Design Research in Information Systems, Springer, Boston, MA, USA, 2010, pp. 9–22.

[21] P. Putten, "Insurance company benchmark (COIL 2000) [Dataset]," UCI Machine Learning Repository, 2000. [Online]. Available: https://doi.org/10.24432/C5630S

[22] Basketball-Reference. [Online]. Available: https://www.basketball-reference.com

[23] GroupLens. [Online]. Available: https://grouplens.org/datasets/movielens.