Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 46 No. 7
July 2025

An Optimization of Data Sourcing Services using Data Gateway

Vijay Shyam Dahake

ORCID ID: 0000-0002-5365-3684

MIT College of Management, MIT ADT University, Pune 412201

India

Abstract

Introduction: This article presents the design and testing of a GNN-based data gateway that can be used to build query-based data mapping and categorisation in corporate settings. By using the Graph Neural Network's (GNN) capacity to record complex interactions between entities inside datasets, the proposed method aims to increase data retrieval speed and accuracy.

Objectives: The study aims to improve commercial applications' decision-making processes by optimising data sourcing services using a GNN-based data gateway, boosting query-based data mapping's accuracy and efficiency.

Methods: Five data classes are used to evaluate the GNN model: MongoDB, Sales, Product, Employee, and Customer. Its performance is evaluated using important measures including F1-score, accuracy, precision, and recall in comparison to more conventional machine learning models, especially Random Forest (RF) and Logistic Regression (LR).

Results: Across all assessment measures, the GNN outperformed the LR and RF models. With a total accuracy of 0.99, it demonstrated exceptional recall and precision. The reason for its exceptional performance is its capacity to record complex relationships between data nodes.

Conclusions: These findings demonstrate the GNN-based data gateway's scalability and dependability as a potential strategy for contemporary data-driven commercial applications. In a variety of sectors, it has significant potential for improving query-based data mapping, facilitating improved decision-making, and improving operational efficiency.

Keywords: Graph Neural Network (GNN), Data Gateway, Query-Based Data Mapping, Classification, Business Data, Data Retrieval, Scalable Solution.

1. Introduction

Large enterprises depend largely on on-premise data gathering; yet, legal constraints usually preclude a complete transition to the cloud. As a consequence, many businesses maintain a master copy of their data on-site and only project pieces to the cloud [1]. However, this hybrid model raises issues with data accessibility, synchronization, and compliance management, necessitating new solutions to enable smooth data operations.

Traditional data source solutions' inflexible design makes them difficult to scale, maintain, and coordinate as data management systems change. These solutions frequently require user applications to connect to numerous databases, which leads to inefficiencies and increased expenses for operations [2]. To address these problems, cloud-based solutions such as Data as a Service (DaaS) have been created. DaaS enables flexible data consumption, but it has costs, data longevity issues, and difficulties adapting to changing business needs [3]. Moreover, inefficiencies are exacerbated by the necessity to modify multiple interconnected systems due to frequent changes in data sources [4].

In order to address these issues, Graph Neural Networks (GNNs) have become a powerful tool for modern data management, providing a scalable and adaptable approach to dynamic data routing independent of permanent storage [5]. GNN-based Data Gateways centralize data access and guarantee that changes are efficiently managed in one location, in contrast to traditional systems that

depend on static data mappings. In addition to improving overall data retrieval performance, this significantly reduces system disturbances [6]. Through the use of graph-based learning, GNNs improve query-based data retrieval, eliminate redundant bug reporting efforts, and speed up update testing [7]. Because GNNs can dynamically adjust to fit changes in data relationships, they are the ideal choice for businesses seeking scalable, flexible, and secure data management systems.

To overcome the limitations of conventional data service designs, this work proposes a GNN-powered Data Gateway [8]. Integrating GNNs into data sourcing plans helps companies to save running costs and guarantee flawless integration of new data sources and tools. For companies managing complicated, changing data environments, the suggested strategy provides a strong, future-ready answer [9].

2. Literature Review

[10] The project integrates big data, advanced analytics, and artificial intelligence with a cloud platform that serves on-demand services to information-driven organizations. Traditional and Big Data analysis tools turn data from multiple sources into useful knowledge. An everything-as-a-service strategy makes it available to organisations, people, objects, and systems worldwide at any time. As seen by the triumphs and failures of businesses during the coronavirus pandemic, digital transformation is crucial. This article presents a digital transformation architecture for modern corporations.

[11] This study aims to optimize traffic lights through the development of a centralized control system via a distinctive wireless communication network. Standard urban intersections were examined to demonstrate the effectiveness of the system. Following the implementation of direct control methods, network traffic lights possess comprehensive authority over anomalous occurrences, including road closures due to accidents or public events. Safety standards were established to apprise central management of the status of the traffic signal lights. An operational phases timing diagram was created for every traffic light using a logic analyser, allowing system validation based on similarities between theoretical and practical timing diagrams.

[12] This study explores many applications of graph neural network (GNN) models, including text, entities, and relations. In order to comprehend dependencies, GNNs help nodes in a graph communicate information. This study examines GNN models, including graph convolutional networks (GCNs), graph attention networks (GATs), and graph SAGE, along with their message-passing methods, benefits, and drawbacks. It also looks at the many uses for GNNs, the datasets that are often used, and Python libraries that enable GNN models.

[13] This research looks at equipment model data standardization technologies and unified data interfaces based on the digital station information system to solve problems such different information islands, fault patterns, and low data values. Modelling is done using the SG-CIM model definition, which offers standardised data. The equipment information model, application and development standards, and unified basic software and hardware capabilities comprise the unified data interface, which enables equipment-centric panoramic data aggregation and exchange. This digital station information system's coordination and interaction may help the country advance the "double carbon" goal.

[14] suggested a subscription-based data-sharing mechanism based on DaaS and blockchain concepts." Customers may use this idea to pay according to the subscription plan for data access and subscribe to a DP for a certain period of time. Since the DP makes money gradually, it has a larger profit margin than selling data right away. The outcomes of the simulation demonstrate how useful and effective the proposed model is.

[15] Compared REST and GraphQL data fetching approaches for a web application called Bakery Service. The experiment involved creating separate API implementations and measuring data fetching performance. Results showed REST with field-level filtering outperformed GraphQL in most test situations, overcoming GraphQL in data utilization. GraphQL may simplify the API and develop further capabilities. According to this study, GraphQL can be a cutting-edge data retrieval solution for

contemporary data-intensive applications. GraphQL could be a simplified API for further development. This study suggests that GraphQL may be an advanced fetching solution for contemporary applications that need intense data.

Besides, [16] will test and compare the efficacy of REST and GraphQL API services for real-world scenarios with complicated databases and plenty of query interactions. In this situation, a fair assessment utilising QoS metrics like response time, throughput, CPU load, and memory utilisation would be made possible by provisioning client requests and service answers along parallel execution paths. This means that REST is suited for frequent, periodic data access, while GraphQL is suitable for dynamic data needs and proper resource utilization.

Customizable manufacturing services are available on demand via cloud manufacturing. Nevertheless, it is still challenging to integrate the cloud manufacturing platform with field-level production data. This study looks at how cloud manufacturing systems use Industrial Internet of Things (IIoT) technolog [17]. For effective cloud platform query and storage, the suggested service-oriented plug-and-play (PnP) IIoT gateway system gathers critical data about industrial equipment at the field level.

[18] classified the key success variables for data integration in a hybrid cloud architecture into three categories: environment, organisation, and technology. By showing the value of top management support, security, and government regulation in organisation, technology, and the environment, the results contribute to the knowledge of cloud computing and data integration.

Research Gap

The literature review underscores several developments in data-sharing methods, API performance, and cloud manufacturing systems; a research gap exists in integrating these ideas for optimum data source and access. Although blockchain-based subscription models and IoT gateway solutions present promising frameworks for efficient data sharing and integration, there is an insufficient investigation into the potential synergy between these technologies and contemporary API services such as REST and GraphQL to improve data sourcing platforms' performance and resource efficiency. Moreover, although studies assess the efficacy of REST and GraphQL, they fail to consider the optimisation of these APIs inside cloud-based or hybrid cloud architectures, particularly for intricate, high-volume data searches and fluctuating data requirements. Additionally, these challenges indicate the need for more studies examining the amalgamation of data-sharing methods, API selection, and cloud technologies to enhance data source services.

3. BACKGROUND

Graph Neural Network

A relatively recent type of neural network called graph neural networks (GNNs) is designed to handle the complex interactions seen in data sources that are graph-structured. Inference efficiency may be increased by using GNNs to encode node and edge information from the graph with more flexibility and a wider representation space. G ¼ (V, E) is the formal representation of a graph, where V is the node set and E is the connection between the nodes. Every node in this drought prediction job corresponds to a grid point [19]. The matrix A €[{0, 1}$^{N\times N}$] represents E as a symmetric adjacency matrix, where A_{ij} = 0 else and A_{ij} = 1 if two grid points v_i, v_j € V are adjacent [20]. There are a total of N grid points. Every node has a corresponding value of $x_{c,t}$ for every month.

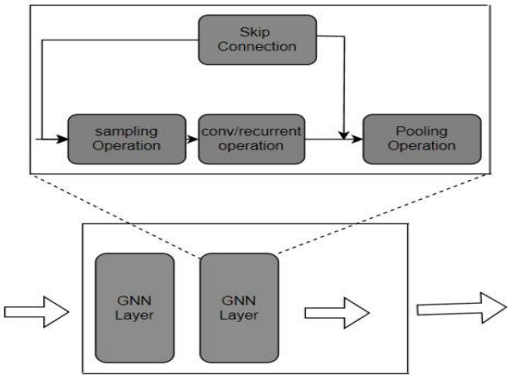

Figure 1 GraphSAGE GNN Architecture [21]

GraphSAGE depicted in Figure.1 is a popular GNN model that uses neighbourhood aggregation to train node embeddings using node feature information. In contrast to existing approaches that depend on matrix factorisation and normalisation,

GraphSAGE collects information from a node's immediate vicinity and requires less computing power [22]. Because it can incorporate variables from various hop counts and search depths, the model is extremely versatile and improves generalization. A useful technique for predicting drought is GraphsAGE since the adjacency matrix is sparse because most grid points have few nearby points[23]. Formally, for GraphSAGE's -th layer.

$$a_{c,t}^{(l)} = gl\left(\left\{z_{c',t}^{(l-1)}, \forall c' \in N(C)\right\}\right) \quad (1)$$

$$z_{c,t}^{(l)} = \sigma\left(W^{(l)} \cdot \left(z_{c',t}^{(l-1)}, a_{c,t}^{(l)}\right)\right) \quad (2)$$

where $z_{c,t}^{(0)} = x_{c,t}$, and $l \in \{0,1,...,L\}.N(c) = \left\{c', \forall A_{c,c'} = 1\right\}$ is the group of grid points around c. Bygl(.) represents the function that aggregates the n-th layer; this function may be pooling, mean function, or graph convolution. They found via study that aggregation works well in GraphSAGE [24]. To get the aggregated feature, the average of the features of each node's neighbours is taken. By doing this, the node can gather the general characteristics of the nodes around it. Mean aggregation is a straightforward yet effective method for discovering the graph's spatial structure[25]. The combined embedding from the adjacent grid points is denoted by $a_{c,t}^{(l)}$. Before the transformation using W(l), we concatenate the from the last layer with the. A non-linear function is (.)

4. Methods

The Proposed methodology as shown in Figure.2 employs Graph Neural Networks (GNN), a to enhance data mapping and processing across many business domains, including HR, Finance, Sales, and Marketing. GNNs model the data as graphs, with nodes standing for entities and edges for connections, enabling the study of complex interdependencies between data points. This method enables a centralized data service architecture that connects several databases, including SQL, Oracle, and MongoDB, while improving query-based data retrieval. The technology allows businesses to query data based on specified criteria, resulting in faster, more accurate decision-making while preserving scalability and agility when new data is added.

Data Pre-processing

Various data preprocessing techniques are implemented to guarantee that the GNN receives high-quality input data. Interpolation is employed to address missing values in numerical fields, while mode imputation is employed for categorical fields. Min-Max scaling is implemented to normalize continuous variables, thereby guaranteeing that they remain within a consistent range and preventing feature dominance and numerical stability. Furthermore, categorical features are incorporated to ensure compatibility with the GNN model.

Feature Engineering

The GNN is optimized through feature engineering, which involves the transformation of raw data into meaningful representations. Node attributes were encoded using numerical vectors, while the strength and importance of interactions were encoded using edge weights. Hidden patterns were extracted using GraphSAGE-based embeddings, and stability was guaranteed through Min-Max scaling. One-hot encoding and embedding layers were employed to convert categorical attributes. The GNN model's predictive accuracy, scalability, and efficient query processing were all improved by these engineered features.

Query Submission

Once the database is filled up, the system starts processing business queries. A business user coming from HR, Finance, Marketing, etc., will send a data query like:

"Which products do the most customers buy in Region X?

"Which employees are tied to best-selling items?".

These queries are typically submitted in an SQL-like format or as business-specific queries, such as "Top products sold during the past month." The query then goes through the GNN based Data Gateway, which processes it based on the graph database's relationship.

Query Processing using GNN-Based Data Mapping

Once the query has been received, the GNN-based Data Gateway uses a learned Graph Neural Network (GNN) model to understand and map the query onto appropriate data nodes and relationships.

Step-by-step GNN processing:

The GNN model checks out the graph structure to establish the relevant entities (Customer, Product, Employee, etc.) and their relations (purchased, managed, etc.) with them.

The GNN model is trained using historical data, such as prior sales records, personnel information, and client details. The model learns how these data entities and their connections interact, which allows it to link query context to the appropriate data nodes.

Mathematical Representation: The process may be characterised as an iterative message-passing mechanism in which nodes update their feature vectors depending on information from neighbours.

$$h_i^{(k)} = \sigma\left(W^{(k)} h_i^{(k-1)} + \sum_{j \in N(i)} A_{ij} h_j^{(k-1)} \right)$$

$H_i^{(k)}$ represents the feature vector of node i at the k-th layer.

σ is a function of activation (like ReLU).

$W^{(k)}$ is a function of activation (like ReLU).

A_{ij} The adjacency matrix element represents the connection (edges) between nodes i and j.

The GNN propagates information throughout the network via this message-passing mechanism, learning the intricate interdependencies between entities (for example, sales patterns connected to customer demographics).

Database Connection and Data Retrieval

After processing the query, the system uses the GNN model's mappings to identify the appropriate data sources (Oracle, SQL, MongoDB, Salesforce). The system uses connectors to interconnect numerous databases, enabling it to get data from diverse sources.

The connections include the following:

SQL Connector: Handles relational database queries.

Oracle Connector: retrieves business data from Oracle databases.

Salesforce Connector: Provides access to CRM data, such as customer profiles and sales history.

MongoDB Connector: This tool works with NoSQL data, including unstructured or semi-structured data such as customer feedback or product evaluations.

These connections enable the system to easily access and obtain the necessary data depending on the query mappings defined by the GNN model.

Response Generation

The last stage in the approach is to generate the answer. Based on the converted and integrated data, the system combines the query results in a structured manner like this:

A report on product sales by area.

A list of personnel in charge of top-selling items.

The answer is returned to the business user, ensuring the data is correct and presented user-friendly.

scalability and flexibility

The system is intended to grow effectively when more data sources or query types are introduced. To improve precision and adaptability to the shifting business requirements, a GNN model could be periodically trained on new data. With connection-established interfacing between multiple data sources, expansion can happen without significant penalties in rearranging current processes.

Cost Efficiency

The GNN paradigm centralises query processing and mapping, sharply reducing maintenance costs. When the database schema changes (for example, an additional field is added to the employee database), changes are made only in the Data Gateway, not in a consumer application. Centralising reduces the development time and effort to maintain multiple connections and

interfaces between different business systems, providing substantial cost savings.

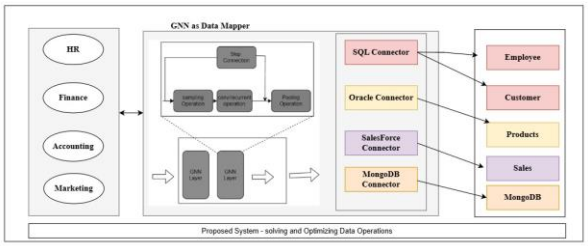

Figure 2 Proposed System of the study

Model Training Parameters and model configuration

Effective model learning and generalization are ensured by using 80% of the dataset for training and 20% for testing. The following hyperparameters are used to train the GNN model:

Learning rate: 0.001 (tuned using grid search for stability)

Optimizer: Adam (to speed up convergence and avoid vanishing gradients)

Loss Function: Cross-entropy loss (to improve classification performance).

Layers: Three GNN layers (to capture multi-hop dependencies in the graph).

Dropout rate: 0.3 (to avoid overfitting).

Batch size: 128 (to balance training efficiency and model generalization).

Model Evaluation

The ROC curve and the area under the curve (AUC) were used to evaluate the performance of the suggested GNN model. To assess the model's capacity to correctly categorize the query categories, significant classification metrics such as accuracy, precision, recall, and F1-score were also examined. Better model performance is indicated by higher AUC values. The True Positive Rate (TPR) and False Positive Rate (FPR) at various thresholds are contrasted by the ROC curve. Accuracy and recall are balanced in the F1-score, and these metrics offer an extensive overview of the model's classification performance.

Accuracy: Accuracy (Acc), which is the ratio of correctly classified samples to all samples, is one of the most widely used metrics for classification performance.

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

where P and N stand for the number of positive and negative samples, respectively.

Precision: Precision is known as the ratio of True Positive elements to the total number of expected positives (column total). True Positive components are positive, whereas False Positive components are negative even though the model labels them as such [26].

$$\Pr ecision = \frac{TP}{TP + FP}$$

Recall: The proportion of True Positive components corresponding to the total number of favorably categorized units (row sum of actual positives) is known as recall. For example, things that the model classifies as negative but are really positive are referred to as false negatives.

$$\mathrm{Re}call = \frac{TP}{TP + FN}$$

F1-Score: Using F1 Score, which measures the usefulness of imbalanced datasets, Precision and Recall are important.

$$F1 - Score = \frac{(P * R)}{(P + R)}$$

Roc-Curve: The Receiver Operating Characteristic (ROC) curve displays a classifier's diagnostic performance across thresholds. At different threshold levels, it shows the True Positive Rate (TPR) (recall or sensitivity) in relation to the False Positive Rate (FPR) (1-specificity) . Often used to test binary classification models [27].

True Positive Rate:

$$TPR = \frac{TP}{TP + FN}$$

Where,

TP=True Positive.

FN=False Negative.

False Positive Rate:

$$FPR = \frac{FP}{FP + TN}$$

Where,

FP=False Positive.

TN=True Negative.

5. Results And Discussion

Results

This chapter gives the proposed GNN-based Data Gateway classification results in mapping queries into the right datasets. The system's rating is based on the ability to execute queries, identify data nodes, and gain correct results from the link databases. The F1-score, precision, recall, and total accuracy metrics demonstrate the GNN's classification performance. In order to guarantee scalability, flexibility, and reliable categorization results for real-world business scenarios, the system's usability when handling various queries across various domains is also examined.

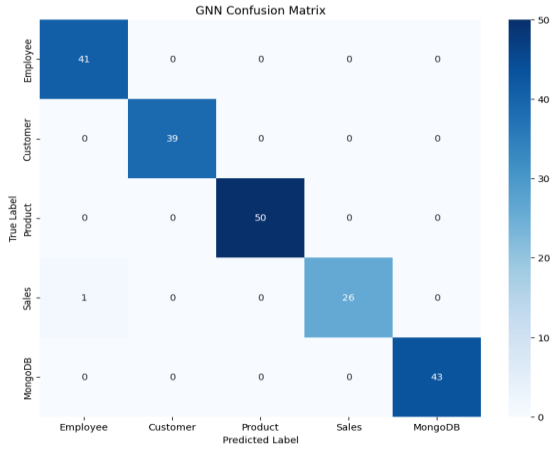

Figure 3 GNN Confusion matrix

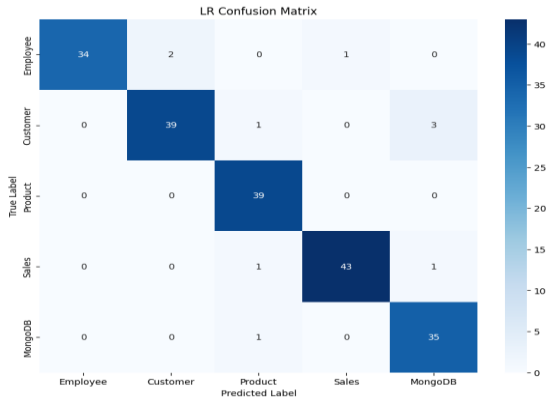

Figure 4 LR Confusion Matrix

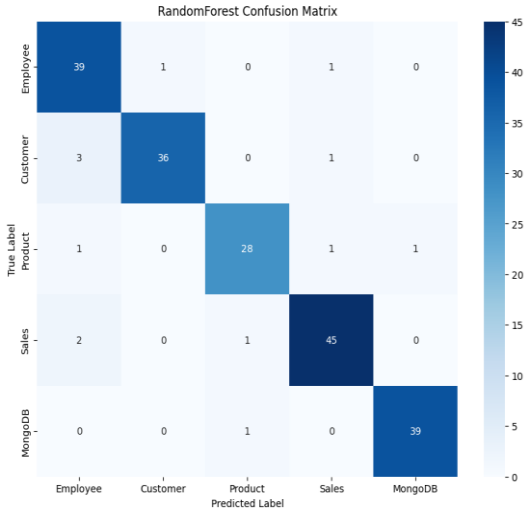

Figure 5 Random-Forest Confusion Matrix

Employee, Customer, Product, Sales, and MongoDB are the five classes represented in the confusion matrices of the proposed GNN, logistic regression (LR), and random forest (RF) models, which are displayed in Figures 3 to 5. In the Employee class, the proposed GNN model correctly identified 41 out of 42 queries, outperforming both LR and RF, which correctly classified 34 and 39 questions, respectively. GNN successfully classified all 39 inquiries in the Customer class, whereas LR and RF correctly identified 39 and 36 questions, respectively. Results for the Product class were similar, with GNN correctly classifying all 50 queries and LR and RF correctly classifying 39 and 28, respectively. In the sales class, LR and RF correctly classified 43 and 45 questions, respectively, with a few misclassifications, while GNN correctly identified 26 of 27 questions with only one misclassification. Last but not least, GNN successfully recognized all forty-three queries in the MongoDB class, whereas LR and RF correctly classified 35 and 39 queries, respectively. Overall, the proposed GNN model demonstrated improved accuracy and fewer misclassifications, particularly for complex classes like Product and MongoDB, where it achieved perfect classification. These results show how the GNN model is the most dependable technique for query-based data mapping and classification because it can capture intricate data linkages.

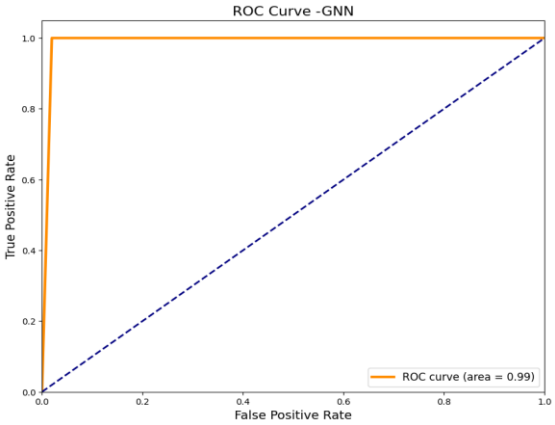

Figure 6 GNN ROC-Curve

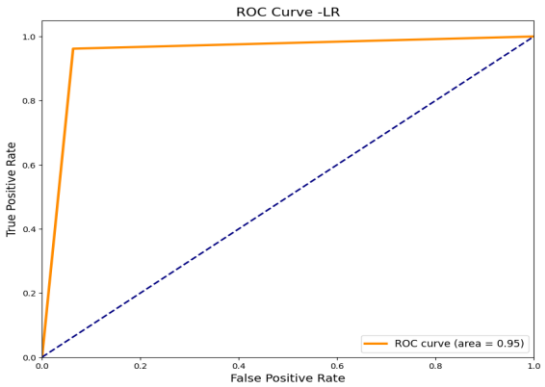

Figure 7 LR ROC-Curve

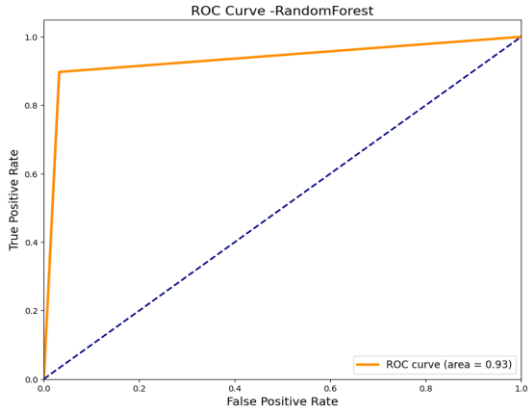

Figure 8 Random-Forest ROC-Curve

Figures 6 to 8 show the performance of each model using the AUC under the ROC curve. The higher the AUC, the better the classification ability. The study compares the three models, LR, and the proposed based on their ability to discriminate between the five categories, customer, product, sales, and MongoDB.

The proposed GNN model achieved an AUC of 0.99, which implies its superior classification capability for queries belonging to all classes. On the other hand, the Logistic Regression model has an AUC of 0.95, which is fair but not that great compared to the previous two models. The worst performer of the three models is Random Forest, with an AUC equal to 0.93, meaning its misclassification rate was considerably high.

These results indicate the robustness of the proposed GNN model, as it continuously outperforms the rest in distinguishing between the complex relationships of the data, making it the best option for query-based data mapping.

Table 1 evaluation metrics

Model	Accuracy	Precision	Recall	F1-Score	Improvement in Recall (%)
GNN (Before Handling)	0.96	0.95	0.89	0.92	—
GNN (After Handling)	0.99	0.981	0.978	0.99	8.80%
LR (Before Handling)	0.92	0.91	0.86	0.88	—
LR (After Handling)	0.95	0.9523	0.95	0.9512	9.00%
RF (Before Handling)	0.91	0.9	0.85	0.87	—
RF (After Handling)	0.9398	0.93	0.9318	0.9398	9.60%

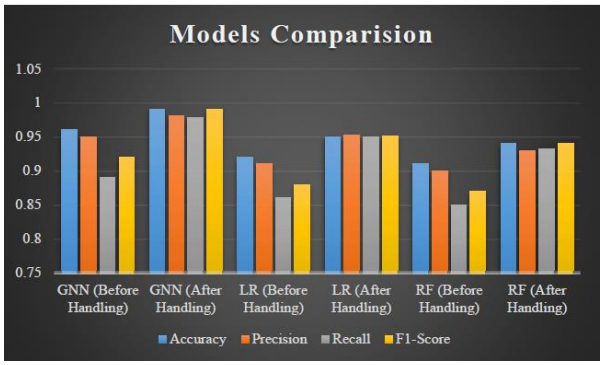

Figure 9 Models Comparison

From the above Table 1 and Figure 9, it is clear that the suggested model significantly improved performance across all assessment criteria by precisely classifying the domains before and after resolving class imbalance. Particularly in recall, which improved across all models, the use of methods like SMote and class-weight tweaks produced significant results. While Random Forest (RF) and Logistic Regression (LR) had recalls of 9.0% and 9.6%, respectively, the recall of GNN increased from 0.890 to 0.978 +/-8.8%. These improvements show that the models started to detect minority class events more accurately, hence lowering misclassification errors.

Accuracy increased across all models as well; GNN reached 0.99, LR rose to 0.95, and RF improved to 0.9398, thus verifying that managing class imbalance favorably affected general model dependability. Additionally, showing increases were precision and F1-score, which guarantee that the models minimized false positives and appropriately identified positive instances. GNN showed the most notable performance increase across the models as it proved better to generalize over unbalanced datasets.

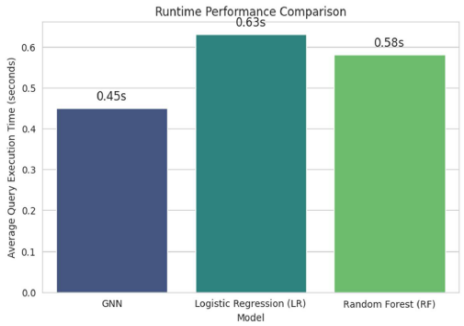

Figure 10 Comparison of Runtime performance of proposed GNN model with existing models

Runtime performance assesses the real execution time needed for each model to handle the dataset is shown in Figure 10. This investigation compares the training durations of Graph Neural Network (GNN), Logistic Regression (LR), and Random Forest (RF). GNN has the fastest execution time of 0.45 seconds, because to its O(N + E) complexity, which allows for rapid message flow between nodes without costly sorting operations. Logistic Regression, on the other hand, took 0.63 seconds because it requires matrix transformations and repeated gradient descent, both of which are computationally intensive. Random Forest outperformed LR, with an execution time of 0.58 seconds, since it generates many decision trees in parallel while still requiring recursive sorting for node splits. The findings reveal that GNN is the most efficient model for dealing with complicated data structures, whereas LR and RF, albeit having comparable O(N log N) complexity, have modest runtime differences owing to various optimization tactics.

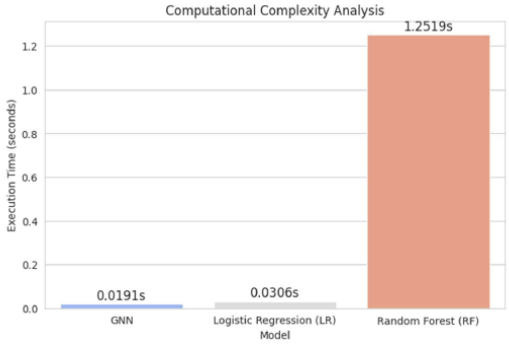

Figure 11 Comparison of Computational Complexity of Proposed GNN with existing models

Computational complexity guides the growth in execution time of a model as the size of the dataset rises as shown in Figure 11. Graph Neural Network (GNN), Random Forest (RF), and Logistic Regression (LR) were assessed in this study based on their theoretical complexity and real runtime performance. GNN operates in 0.0191 seconds with a complexity of O(N + E), where N is the number of nodes and E is the number of edges. This architecture reduces needless computations by ensuring scalability and facilitating efficient message transfer between connected nodess. By way of execution durations of 0.0306 seconds and 1.2519 seconds respectively, Logistic Regression

and Random Forest both show O(N log N) complexity. While Random Forest builds multiple decision trees, where each tree recursively splits the data using sorting operations, causing significant execution slowdowns, logistic regression depends on matrix transformations and iterative gradient descent, which increases computational cost as data size grows. The findings validate that GNN is the quickest model, scaling well with graph-based data, while LR and RF face increased computational cost owing to sorting activities

Discussion

The resulting display reveals that the presented GNN model is more appropriate for query-based data mapping and classification problems. The GNN performed better than the current models, LR and RF, in every metric that was looked at, including accuracy, precision, recall, and F1-score. The GNN outperformed LR (0.95) and RF (0.9398) in handling complex connections in data, demonstrating an accuracy of 0.99. Increased accuracy scores (0.981) and recall ratings (0.978) demonstrate the model's capacity to lower false positives and false negatives, resulting in balanced and reliable classifications.

The confusion matrix study showed how the GNN achieved almost perfect or perfect classification performance in the challenging classes Product and MongoDB, while all other models, particularly RF, faced higher misclassification rates. The GNN's brilliant results are due to its ability to effectively capture and describe complex linkages between entities, resulting in its resilience in a different kind of categorization context.

Additionally, ROC curve analysis showed that GNN was more discriminatory. Its AUC was recorded to be 0.99, and it clearly outperformed LR by 0.95 and RF by 0.93. Therefore, this signifies the GNN's capability to identify between classes even in complicated and overlapping distributions.

The results demonstrate the versatility and applicability of GNN models in real-world applications, which require features such as scalability, adaptability, and accuracy. By leveraging the GNN's capability to model vast data linkages, the system provides a secure solution for query-based data mapping, thereby making it optimum for various business domains. These findings validate the proposed GNN model as the best strategy for such jobs and suggest that it can improve decision-making and operational efficiency in data-driven systems.

6. Data Availability

Coding has been done in Python language to test the use cases which covers -3 Models, that are used for comparison (GNN, Random Forest and LR).

Original Coding file, Test Data, Results Generated, Graphs and all supporting files can be made available based on request from the Journal Editor or any competent authority on a request basis.

7. Conclusion

The proposed GNN-based Data Gateway is introduced in this paper. The merits and demerits are discussed with its evaluation of query-based data mapping and classification applications. The performance results are really rather clear when compared to other well-known algorithms such as Random Forest and Logistic Regression. Recall of 0.978, accuracy of 0.99, precision of 0.981, and F1-score of 0.99 all show how well the GNN model can handle complex and interrelated data connections. The comparison made with the LR and RF depicted that although these models were doing well, those results did not appear as fruitful in identifying complicated relationships within data; therefore, resultant accuracy was lesser, and misclassification rates were larger.

Its applicability for classifying queries from many business domains like Employee, Customer, Product, Sales, and MongoDB makes GNN valuable in real-life situations. In addition, the performance on the ROC curve with a brilliant AUC of 0.99 can distinguish between classes. The findings show that the GNN-based approach is efficient for query processing optimization and thus forms a very reliable tool for enterprises dealing with massive and complex datasets that must be categorised accurately and quickly. This study reveals that GNNs can transform the data mapping system as they provide a scalable and adaptive solution that can cater to the changing demands of businesses. This proposed model outperforms current techniques and lays a robust platform for further improvements of data-driven decision-making systems in virtually every sector.

8. Acknowledgements

I would like to express my sincere gratitude to my research guide (Dr. Padmakar Shahare) and Research coordinator (Dr. Pralhad Tipole) of MIT ADT University for their support and primary review to this work.

9. Statements and Declarations

Funding Information: This research is self-sponsored, hence no funding acquired.

Conflict of Interest Statement: The author declares that there are no conflicts of interest associated with this research.

Competing Interests: The paper is made for the efficient use of data with minimum complexity in field of software, however there is no financial or non-financial interests that are directly or indirectly related to the work submitted for publication.

Refrences

[1] K. Perakis *et al.*, "Data sources and gateways: Design and open specification," *Acta Inform. Medica*, vol. 27, no. 5, p. 341, 2019.

[2] L. M. Abdulrahman *et al.*, "A state of art for smart gateways issues and modification," *Asian J. Res. Comput. Sci.*, vol. 7, no. 4, pp. 1–13, 2021.

[3] M. F. Radhiyan, D. Khairani, and H. B. Suseno, "Analysis and design of microservices architecture with graphlql as an api gateway for higher education information system," in *2022 International Conference on Science and Technology (ICOSTECH)*, IEEE, 2022, pp. 1–7.

[4] C. Liu, Z. Su, X. Xu, and Y. Lu, "Service-oriented industrial internet of things gateway for cloud manufacturing," *Robot. Comput. Integr. Manuf.*, vol. 73, p. 102217, 2022.

[5] C. An, M. Wang, and L. Lv, "Research on wireless acquisition and analysis system of gateway table data based on improved ACO algorithm," in *E3S Web of Conferences*, EDP Sciences, 2024, p. 2008.

[6] F. Hasić, J. De Smedt, S. vanden Broucke, and E. Serral, "Decision as a service (DaaS): a service-oriented architecture approach for decisions in processes," *IEEE Trans. Serv. Comput.*, vol. 15, no. 2, pp. 904–917, 2020.

[7] K. Hsu, "Big data analysis and optimization and platform components," *J. King Saud Univ.*, vol. 34, no. 4, p. 101945, 2022.

[8] R. Kufakunesu, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on adaptive data rate optimization in lorawan: Recent solutions and major challenges," *Sensors*, vol. 20, no. 18, p. 5044, 2020.

[9] L. Shimei, Z. Jianhong, L. Enfeng, and H. Gang, "Design of industrial internet of things gateway with multi-source data processing," in *2020 International Conference on Computer Engineering and Application (ICCEA)*, IEEE, 2020, pp. 232–236.

[10] S. Wan, S. Ding, and C. Chen, "Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicles," *Pattern Recognit.*, vol. 121, p. 108146, 2022.

[11] L. F. P. De Oliveira, L. T. Manera, and P. D. G. Da Luz, "Development of a smart traffic light control system with real-time monitoring," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3384–3393, 2020.

[12] B. Khemani, S. Patil, K. Kotecha, and S. Tanwar, "A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions," *J. Big Data*, vol. 11, Jan. 2024, doi: 10.1186/s40537-023-00876-4.

[13] W. Xiaojin, S. Shucai, X. Yehua, J. Tao, and L. Hongkun, "Research on data standardization and unified data interface based on digital station system," in *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, IEEE, 2022, pp. 1372–1376.

[14] F. A. Al-Zahrani, "Subscription-based data-sharing model using blockchain and data as a service," *Ieee Access*, vol. 8, pp. 115966–115981, 2020.

[15] M. T. Baldassarre, D. Caivano, S. Romano, and G. Scanniello, "Software Models for Source Code Maintainability: A Systematic Literature Review," in *2019 45th Euromicro Conference*

on Software Engineering and Advanced Applications (SEAA), 2019, pp. 252–259. doi: 10.1109/SEAA.2019.00047.

[16] R. Siddalingappa and S. Kanagaraj, "Anomaly Detection on Medical Images using Autoencoder and Convolutional Neural Network," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 7, pp. 148–156, 2021, doi: 10.14569/IJACSA.2021.0120717.

[17] L. Gualtieri, E. Rauch, and R. Vidoni, "Emerging research fields in safety and ergonomics in industrial collaborative robotics: A systematic literature review," *Robot. Comput. Integr. Manuf.*, vol. 67, p. 101998, 2021, doi: https://doi.org/10.1016/j.rcim.2020.101998.

[18] A. P. Pljonkin, "Vulnerability of the synchronization process in the quantum key distribution system," *Int. J. Cloud Appl. Comput.*, vol. 9, no. 1, pp. 50–58, 2019.

[19] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?," *arXiv Prepr. arXiv2105.14491*, 2021.

[20] J. Yu, T. Ma, L. Jia, H. Rong, Y. Su, and M. M. A. Wahab, "Multivariate spatio-temporal modeling of drought prediction using graph neural network," *J. hydroinformatics*, vol. 26, no. 1, pp. 107–124, 2024.

[21] G. Corso, H. Stark, S. Jegelka, T. Jaakkola, and R. Barzilay, "Graph neural networks," *Nat. Rev. Methods Prim.*, vol. 4, no. 1, p. 17, 2024.

[22] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021, doi: 10.1109/TNNLS.2020.2978386.

[23] Y. Li, D. Yu, Z. Liu, M. Zhang, X. Gong, and L. Zhao, "Graph neural network for spatiotemporal data: methods and applications," *arXiv Prepr. arXiv2306.00012*, 2023.

[24] R. Zhou, "Multi-scale dynamic spatiotemporal graph attention network for forecasting karst spring discharge," *J. Hydrol.*, vol. 659, p. 133289, 2025.

[25] P. Li, Y. Yu, D. Huang, Z. Wang, and A. Sharma, "Regional heatwave prediction using graph neural network and weather station data," *Geophys. Res. Lett.*, vol. 50, no. 7, p. e2023GL103405, 2023.

[26] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," *arXiv Prepr. arXiv2008.05756*, 2020.

[27] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.