Journal of Computer Science

Research Article

Identification of Influential Nodes With Topological Structure Via GRAPH Neural Network (GNN) Approach in Social Media Networks

Rajnish Kumar[1], Laxmi Ahuja[2], Suman Mann[3] and Sanmukh Kaur[4]

[1]Department of Computer Science, Amity University Noida, India
[2]Departmant of Information Technology, Amity University Noida, India
[3]Department of Computer Science, Panipat Institute of Engineering and Technology, Smalkha, India
[4]Departmant of Electronics and Communication Engineering, Amity University Noida, India

Article history
Received: 16-04-2025
Revised: 07-05-2025
Accepted: 21-10-2025

Corresponding Author:
Rajnish Kumar
Department of Computer Science, Amity University Noida, India
Email: rajnishghose@gmail.com

Abstract: The amount of data in social networks is vast these days, and constantly changing, making influential node identification crucial. The existing topologies are constantly changing due to the evolving behavior of the applied dataset. Node and leaf topology or feature-based value form the basis for machine learning and centrality computation. Hence, influential node value determination is based on the node attribute and network topologies. In the context of a large dataset, working towards the identification of the most influential node in the network, the Graph Convolutional Network (GCN) is the most effective and trusted approach. In the current research paper, the GCN has been projected as the most effective approach towards the identification of the node that is most influential in the graph-based dataset. The graph-based datasets are very large. A deep learning framework with the help of structural centrality via GCN, known as DeepInfNode, has been developed for the identification of the most influential Node. The Susceptible-Infected-Recovered (SIR) model is developed to identify the infection rate. This infection rate is further divided into three categories: Susceptible, Infected, and Recovered. The current approach uses the SIR model to collect contextual information to develop node representations. Higher values of F1 and AUC (Area under Curve) and F1 is visible when the suggested model is used. This has been discussed and explained in the experimental section. The observations prove that the above-mentioned strategy is precise and effective. It also suggests a potential new linkage within the network. An accuracy of up to 98% is achieved on all publicly available standard graphs available from Kaggle for different domains and datasets like Facebook, credit card fraud detection, Twitter, and Disease prediction using machine learning. Implementation of the proposed DeepInfNode works effectively and accurately for different domains. Additionally, performance improvement is confirmed during data processing and experimental analysis with the use of the DeepInfNode framework.

Keywords: Susceptible Infected Recovered, SIR, Deep Learning, Influential Nodes, Graph, Topological Structure

Introduction

The network available in the datasets is either Static or dynamic. For the case of static datasets, the traditional centrality measurement techniques fulfil the requirements. But when talking about dynamic datasets, the traditional centrality measurement techniques do not satisfy the requirement because of the dynamicity or continuous shift in priority and influence. The traditional centrality measure comes with the following limitations:

- Static snapshot assumptions: The traditional centrality measure always expects that the snapshot of data should be static and there should not be any

SCIENCE Publications

shift in priority or influence, which is not the correct expectation

- Lack of temporal sensitivity: Capturing the temporary shift is not possible in the case of traditional centrality measurement
- No support for cascading effects or influence: Traditional centrality measures generally consider only the immediate neighbor, not the distant neighbors of delayed influence, resulting in an incorrect outcome
- Focus: The traditional centrality measure focuses on topology, not the behaviors that finally result in incorrect observation

Because of these limitations, there is a requirement for a shift from the traditional model to a deep learning model with support for dynamic datasets, along with no limitation on dataset size and complexity. Deep learning models come up with a lot of approaches. Out of multiple approaches available for the abstraction and analysis of the available dataset, a graph is considered the easiest and most convenient way to understand the dataset. The data pattern is consistent and considered intact for real-time series or normal data. Analyzing the data available in the graph helps us find the aspects of the data and draw necessary and useful information from it. The network data or online social media data consists of the root node, intermediate node, and leaf data distribution (Ayman et al., 2022). It is very tough and crucial to find the most relevant node out of all available nodes in the network. The few aspects, like the prominent online communities, the most influential person, measuring the most influential scientific impact, financial risk associated, and forecasting professional growth, are a few use cases in the influential node identification in network analysis and research. As per historical data available, to identify the node that is most influential in the network, the Centrality methodology and machine learning algorithms have been used extensively. These methods quantify the significance of each node from the perspective of node attributes and physical structure. As per the studies available, in the centrality-based technique, the functions are undervalued, whereas the structures are overvalued (Cai et al., 2018). For the case of an influence situation, the functional relevance of any node is not fully determined by the presence of the algorithm, which uses the information available within the solitary framework to establish node importance. Machine learning approaches overuse feature engineering. Feature selection has a lot of encouragement for the effectiveness of current machine learning algorithms. For any node in the network, the overall influence is determined not only by its characteristics but also by the association between nodes and neighborhood nodes as well. The influence of metrics, such as the centrality metrics, shows the statistical perspective of node influence on the graph. It also represents how the presence or absence of that influential node impacts the graph structure and architecture (Chen et al., 2019). The available connected node has more impact on the overall network as compared with the only available nodes in the network. The node influence matrix shows the graph structure changes because of the presence of a highly influential node in the network. Additionally, it shows how the structure will change if the same node is not present in the network. This data is shown statistically for the centrality measure. Communication takes place among the nodes present in the network. The node that can communicate the message rapidly to the adjacent node is considered the shortest node. The shortest path possible is the traversal path between these two nodes. Another centrality measure, k-shell decomposition, shows the quantitative measurement of structural centrality in the network. The K-Core reflects a subgraph in the network in which the participating vertex has a degree either greater than or equal to the actual K-Value. In order to determine the centrality metric, the nodes that have a degree value less than K-Value are selected. This is done using K-shell decomposition, and the activity continues till the optimum centrality metric is obtained. The network is analyzed again to ensure no node in the network has a degree less than the k-value. If any nodes are found, these nodes are also eliminated from the network. Finally, there remain k nodes in the network, and finalized nodes create the sub-graph (Guo et al., 2020).

Out of multiple deep learning models, Graph Conventional Network (GCN) is another deep learning model that has gained a lot of popularity and implementation these days, having the capability of accessing both characteristics and relationships between the nodes. GCN has got a lot of implementations throughout the globe and into multiple disciplines like online social media, biochemistry, clinical research, and Natural Language Processing (NLP). Since the GCN framework can understand and analyze graph standard data, the practical implementation has taken place in a lot of areas, working in different domains. Additionally, GCN can resolve a lot of complex network problems, along with simple networks. Node aggregation is the base of GCN, which works through graph edges. The characteristics are drawn from the neighboring node lying between the path of one node and another in a multi-layer GCN. The neighboring nodes are the nodes that are used to connect the two main nodes in the multilayer network (Ibnoulouafi and El Haziti, 2018). Finally, the node description is summarized by the aggregation of structural information of the immediately surrounding nodes. Multiple authors working on the same domain have suggested that GCN is the best option available to understand the graph topology, as it works on the selective feature aggregation approach. The shortest path is the approach for selective feature aggregation in the network.

The main reason behind using GCN is that each node feature aggregation is done multiple times by traversing between two nodes, covering the shortest distance between the nodes. In the following steps, the value obtained as information of the node's aggregation is used as training data for the scoring function. Centrality measure score is connected by a scoring function using ranking-based loss. As per the hypothesis, 15% of the nodes in the network are considered as significant or influential nodes; the remaining 85% nodes are kept under the category of non-influential nodes in the network. DeepInfNode, a deep learning framework implemented for the identification of influential nodes, has been proposed utilizing a Graph Convolutional Network (GCN). It can analyze node properties as well as the shortest distance between two corresponding nodes. Additionally, it uses Breadth-First Search (BFS) to understand and get insight into hidden predictive signals. (Jena et al., 2022). The computation of the infection rate is done with the help of underlying data derived from the SIR (Susceptible Infected Recovered) model. This activity is done before using the anticipated signals in the task learning layers.

A detailed comparative study has been done for the proposed framework with a lot of established traditional methodologies currently available, including machine learning based algorithms and centrality measures for its accuracy, usability, and effectiveness. As per the data received post comparative analysis, it is fair and worth saying that DeepInfNode is one of the best available deep learning techniques and has the capability to improve predictive analysis dynamically.

Related Work

Multiple methods and approaches are currently available for the most influential node identification or the most significant nodes in the complex network. The network can be very large in size or small, including propagation probability dynamics, information entropy, and many others. Technically, all these techniques, as mentioned above, fall broadly under two categories:

a) Structured centrality approaches
b) Supervised learning-based approach

The approach that has become most popular and usable nowadays among academicians, researchers, and practitioners because of its wide acceptance in the deep learning vertical and community is the Graph Convolutional Network (GCN). Considering the above fact and usability, the current work becomes worth experimenting with and establishing the research approach. This approach, which has been discussed in the current paper, is based on a Machine Learning (ML) approach, centrality measurement, and Graph Convolution Network (GCN).

Machine Learning Approach

During the literature review, it has been seen that various studies have been done in the field of implementation of machine learning, focusing on extracting and selecting features helpful in improving the overall performance. The two most popularly used approaches to implement ML algorithms are:

a) Logistic Regression (LR)
b) Support Vector Machine (SVM)

Identification of critical nodes is easy through machine learning algorithms in a wide range of verticals and associated domains, enabling the scope of future research in the implementation of machine learning algorithms. Machine learning algorithms work on the training and testing data approach. The way the algorithm is trained with the training data is the same way the algorithm will work in the testing data, instead of finding the links between the nodes in the network (Khan and Haroon 2022). The structure of the community and influential difference distribution, along with influence maximization, is the basis of most influential node identification in the network.

Being the structural process, the first step identifies the communities available within the provided network. The communities are the interconnected, closely associated nodes in the network. The second step involves a greedy search within the network for the identification of the nodes with maximum influence in the network. The nodes having similar characteristics and interests are kept in a single network. These are the rank-based communities created within the network.

The following is the sequential activity done in the overall process:

a) Prioritization of the effect on the network based on similar interests within nodes
b) Evaluation of content distribution
c) Come up with the ranking model

The rank is drawn based on the direct and Indirect interconnection between different nodes available in the network based on metric parameters, reachable interest group, and the nodes that are reachable in the network. In addition to the close centrality, outliers are also identified based on the same parameters that are implemented for the interconnection (Kumar and Panda, 2020). The pattern and trends are drawn based on customer information available, their interest, feedback from other nodes, and recommendations from adjoining nodes in the social media network.

Centrality Measure Approach

In order to develop a centrality-based approach, the researchers approach the graph method. They take into consideration flexibility and performance. This approach is based on the topological structure of the network.

The centrality measure approach is divided into four categories (Kumar et al., 2022):

a) Distance-based approach implementing closeness centrality and betweenness centrality, for which the distance between the nodes is considered as the base
b) Degree centrality, for which the neighbor relationship is considered the base. The number of neighbors is computed to identify the degree centrality
c) Eigenvalue and eigen vector implementing iteration-based centrality
d) Page rank centrality, assessing the significance of nodes in the network
e) Gravity and density centrality, having the base of global measures and geodesic distance

Graph Convolutional Network (GCN)

The GCN-based approach is one of the most popular methods in the research community. Since this method is in use and a lot of research work is being done in this area, we have various GCN-based approaches, which are either well-established or in the development phase. This methodology works on a very large network by identifying important nodes in the network, and ranking is done based on their relative importance.

Two categorical classifications are available, keeping the operation as the base:

a) Spatial domain convolutional
b) Spectral domain convolutional

In the spatial approach, the aggregation of information is done based on the neighboring nodes in the graph, whereas in the spectral approach, eigenvalues and eigenvectors are computed to find the density of the nodes in the network (Lü et al., 2016).In the Spatial approach, the nodes in the network are interconnected. In Spectral domain convolutional, the graph Laplacian matrix is used to transform the transformation of graph data into the spectral domain. Further, the convolution operation is done by multiplying with a filter in the frequency domain. Iteratively aggregating neighbor information is used to get both attributes of any node and the corresponding relationship with the neighboring nodes. A Ranked-Based Convolutional Neural Network (RCNN) is developed for locating the most probable critical nodes. These are super-spreaders in the complex network. RCNN is the most efficient iterative technique for identifying the most influential node. A feature matrix is created for every node in the network using the RCNN technique. Accordingly, the algorithm is trained, and a prediction is made for the rest of the nodes in the network using a Convolutional Neural Network (CNN) (Maurya et al., 2021).

For the potential influential node identification, the similarity-based Graph Neural Network (SGNN) is used. SGNN is considered one of the best influence maximization methods available for complex networks. Struc2vec is a defined framework that generates the representation of node vectors, preserving the structural identity of any graph. To find the possible impact on the overall network, the Struc2vec framework is combined with a similarity-based Graph Neural Network (SGNN), along with graph neural network-based regression. Further, to identify the spreading influential nodes, a multi-channel RCNN algorithm (M-RCNN) based on GCN has been created. During the model training, we utilize different matrices such as macro-level, micro-level, and community-level structural information. The weights are enabled for each of the nodes. Neural network techniques are incorporated with attention techniques. The underlying representation is drawn based on the node’s behavior and neighbors. Each node is provided with a unique weight using GCN’s fundamental aggregation function. It’s the GAT layer’s attention coefficients that help determine the weight of each node.

Additionally, deep learning methodology and a knowledge graph are combined in the proposed approach. User interaction is depicted through an effective knowledge graph. User behavior is also studied through an unsupervised deep learning model using an autoencoder. The proposed model is influenced by a Graph Convolutional layer. User relationship and user attributes are studied in the model (Okamoto et al., 2008). The proposed model is well-equipped to handle any type of dataset, including an unlabeled dataset. Load-bearing nodes are enabled in each layer of the neural network. Proper tuning for perception weights has been done at the training model level. The ultimate intent of the complete research work is to design a neural network model with high precision.

The following is a summary of the literature review.

Zhao et al. (2019) used the graph convolutional network approach with the intent of identifying influential nodes in any network with the method InfGCN and GCN technique, with an accuracy of 97%.

Zhang et al. (2017) used the Heuristic algorithm and greedy algorithm with the intent to discover the influential nodes in a social network with the influential maximization method using community structure and influence distribution, with an accuracy of 61%.

Gou et al. (2022) used the Neighbors’ Degree, k-core, and GCN-based approach called RCNN algorithm for influential node identification using Different levels of structural features. The M-RCNN approach was used for the identification of the most influential nodes with an accuracy of 9.25%.

Xiang et al. (2021) used the method Graph Attention Networks, Common neighbor, and centrality measures with the intent of finding infected influential neighbors with an accuracy percentage of 83.5%.

Tran et al. (2015) used the Variational Graph Autoencoder methods for detecting emerging influencers with an accurate percentage of 91.5%.

Zhang et al. (2019) used the GCN approach and heuristic method.

Ayman et al. (2022) used Centrality Measures and Machine learning techniques for the influential node identification using dynamic GCN and UltRank method, respectively, with the accuracy percentage of 91%.

Background

A complex network is a graph G with a sophisticated structure, consisting of a pair of discrete sets. (V, E) €G.

Where V is no of sets of components identified as vertices or nodes, and E is known as edges or arcs.

In a complex network, some nodes are more significant in the network as compared to others. The significance and impact of the significant nodes are higher than those of the nodes having lesser significance. Public figures and leaders have a higher fan following than normal people and have a higher impact on the overall social network. The high-impact nodes are the root nodes in the network. Since centrality is dependent on context, centrality is defined in terms of total hits or requests, or total communication happening from that node. In addition to centrality, other matrices have been drawn as part of the process, focusing on different ideas (Ou et al., 2022).

Degree Centrality

The best way to calculate the centrality is by summing the number of connections between nodes. The adjacency matrix(A) and Degree(vi) are used to visualize the network topology mathematically according to Equation 1:

$$Deg(v_i) = \sum_{J=1}^{n} A_{ij} \tag{1}$$

Here, A_{ij} = 1 if two nodes *a* and *b* are adjacent to each other, else A_{ij} = 0:

Betweenness Centrality (BC)

Statistical information is used to determine key nodes in betweenness centrality. Betweenness centrality can be defined with Equation 2 as follows:

$$BC(v) = \sum_{i \neq J \in v} \frac{\hat{\delta}_{iJ}(v)}{\delta_{iJ}} \tag{2}$$

Where $\delta_i j$ denotes the path between the nodes $i \in V(G)$ and $j \in V(G)$.

In addition to this, the shortest path is represented as ˆ $\delta ij(v)$ between nodes *i* and *j*, passing through node *v*.

Density Centrality (DNC)

The combination of density and centrality is the basis for the creation of a two-dimensional strategic diagram. In any plot, interaction strength between the nodes, i.e., centrality, is denoted on the x-axis, whereas the internal coherence or density is denoted on the y-axis. The density centrality is denoted by Equation 3 below:

$$DNC(v) = \sum_{J \in v_i} \frac{Deg(v)}{\pi dis_{ij}^2} \tag{3}$$

K-Shell Measure

It is a subgraph that contains at least k degrees for every vertex, known as a k-core. To get the centrality value, all nodes having a degree less than k are removed. This decomposition of the k-shell provides a centrality value. With the removal of any node, the corresponding influential node value also changes for all the nodes connected to the removed node. The influential node value is directly proportional to the k-shell value. When the k-shell value increases, the influential node value also increases, and vice versa. The distance is inversely proportional to the impact between any two nodes in the network and is denoted by Equation 4 below:

$$Gs_c(v_i) = \sum_{i=j} \frac{k_s(ij)}{d_i s_{(ij)}} \tag{4}$$

Graph Neural Network (GNN)

The Graph Neural Network (GNN) method is one of the available methods for evaluating Graph-Structured Data. Within GNN, there are a lot of methods available for processing graph-structured data. Aggregation of node and edge characteristics is done through the graph structure in all the GNN models. Proper training is done for the neural network for sharing an edge, prediction of node labels, and other factors.

Message passing Neural Network (MPNN) is a single place where all preceding models are generalized. MPNN is a defined architecture (Qiu et al., 2018). A Graph Neural Network (GNN) is designed in such a way that it can train against the loss function for a graph G along with a feature information matrix. It applies to all nodes and edges. The node feature vector is the way to express the graph. A variety of factors are available to find the number of layers in GNN, including graph size, work in hand, and underlying properties of the graph. It's a node in the GNN that starts the message passing stage by doing the aggregation of the properties of the node and its surroundings. The feature vector of a node is modified when any node and its neighboring nodes modify their feature vector, and the same activity is done for all the layers in the Graph Neural Network. It is expressed with the statistical formula as mentioned in Equations 5 and 6.

Equations 5 and 6 are for aggregation, and combination is defined as follows:

$$A_v^k = Aggregate^k(\omega_u^{k-1}: u\epsilon N_v) \tag{5}$$

$$\omega_k^v = Combine^k(\omega_u^{k-1}, A_v^{k-1}) \tag{6}$$

In Equation 5, the aggregation is done for the kth layer's feature vector. The aggregate function denotes the average, max pooling, or sum of the feature vector, and it is dependent on the model. In Equation 6, a combination is done for the aggregation and node feature vectors (Rodrigues, 2019). A node with cumulative feature data is present for each layer. The Rectified Linear Unit (RELU) is the next step, where the mapping of aggregated characteristics is done with a trainable weight matrix. The outcome of the previous layer is considered the feed-in for the next layer. The iteration continues, and after the kth iteration, all feature information is gathered for all nodes. In addition to feature information, the structural information of all neighbors is also gathered for the final layer's node (Sandhya et al., 2020).

This is done for all distances in the network. It is the principal outline as above that defines the structure of the suggested GNN framework. For simplicity, it is suggested to reduce the weight from the GNN model layer along with non-linearity. The simplified model and GNN model work in the same lines in the classification task. The gain in terms of performance of the model, along with non-linearity among all the layers of the network, is seen in the proposed architecture diagram (Figure 2). The message passing technique is the base technique used to gather feature data in the Graph Neural Network approach for each node from its neighbors. It is done for all the nodes in the network. It is the node's feature data that travels all the links available in the graph using an aggregation approach. K-core centrality and betweenness centrality are used to compute the shortest distance between two nodes using Breadth-first search (BFS). Our proposed message passing scheme works in the same lines for the flow of feature information.

Proposed Framework for the Identification of Influential Nodes

Topological structure and graph neural networks are the best ways to solve the influential node identification problem. There are other ways to solve the problem based on the local and global network structure. But most influential node identification is still considered a problem statement in the implementation of the graph neural network. The dataset is growing rapidly these days, with a diverse range of data types (Shashidhar et al., 2022). Deep learning model DeepInfNode has been discussed in detail in the current section, which is the core of the identification of influential nodes in a social network for a very large dataset.

Benefits of implementing a structured network topology:

- Defines influence pathways
- Centrality measures depend on Network topology
- Structure centralities enhance learning
- GNN improves accuracy
- Reduces computational cost

Structured Network topology helps in identifying the most influential node in the network, along with wide coverage. Centrality measurements are highly dependent on network topology. Structured centrality covers both local and global influential nodes in the network. Since the overage is both local and global, it provides more accurate data about the influential nodes in the network and can be seen in Table 5. Also, with the structured approach, the computational cost is also reduced, as can be seen in Table 6.

For any large dataset, the number of layers increases, and it is highly recommended to consider all nodes in the network in terms of structural centrality and neighbor networks. Additionally, the aggregation of their multi-hop feature vector is done. In the next step, the Graph Convolutional Network (GCN) layer is used to determine the feature of each node in the network (Tran et al., 2022). In the proposed framework, Structural centrality is combined with GCN, resulting in a lot of performance improvement.

It helps in performance improvement by

- Better node selection for training purposes: Influential nodes are prioritized during the training phase by structural centrality, leading to faster coverage and higher classification accuracy, which is generally missing in existing techniques
- Improved Embedding Quality: Peripheral and Central nodes treatment is the same in existing techniques, whereas in Structural centrality, the treatment is done as per their global importance, helping improve embedding quality.
- Enhanced generalization: Training structurally diverse nodes helps in reaching nodes that are not possible with existing techniques. This helps in richer representation.
- WGCN (Weighted GCN) using directional path traversal improves classification accuracy as compared with existing techniques

In this way, all nodes' feature information is obtained. More interconnected nodes might provide more reliable and detailed information across the graph. The outcome of Susceptible, Infected, Recovered (SIR) simulation experiments and the model's outcome are compared to perform a comparative analysis to deduce any likelihood loss (Ullah et al., 2021).

Design Neighbor Network

A network is developed with the nodes inside it, and these nodes are interconnected to each other. The node's influence is generated due to its value and the nodes around it. It's the local attribute that represents the node in the GCN model, where it is linked with its neighbors. These nodes are tied together solely to their neighboring network nodes or k-step network. To assess the impact of the node, its corresponding neighbor network must also be analyzed. The BFS tree is designed to get the node's neighbors, and then the overall network is designed using the same node and its neighbors. Designing deep learning models is difficult and challenging. The difficulty lies in changing the size of the neighboring network. The neighboring network changes with the change of a node in the network (Veličković et al., 2017).

To reduce the size issue:

Let the size of each node's neighbor =A
Step Neighbor to target node = i
Total number of nodes in the neighbor = ti

As part of the next step, the total number of nodes in the neighborhood is calculated. If the value of it is less than A, then the following step is performed. Nodes with higher betweenness centrality are kept, and nodes with lower betweenness centrality are removed for the said ith step. It is considered the criterion because it reflects the bridging role of nodes and is considered a more significant process of transmission of information.

Computational Process of DeepInfNode

The proposed DeepInfNode or Deep Influence Node model consists of four phases, utilizing the feature map of each node in the network:

a) Construction of the calculated balanced Laplacian equation of the network graph
b) Using the feature vector and the graph topological feature, the creation of GCN is derived from the node representation vector
c) Dropout technology is considered a regularization technique to deal with overfitting in neural networks. It is implemented for the first two fully connected layers. Afterwards, each node's impact on the overall network is evaluated
d) The fully linked layer provides the result, and it is received by the classifier at the final stage

The purpose of the proposed model is to solve graph-structured data (Xiang et al., 2021). Hence, it has been developed in such a way that it is an efficient Graph Convolutional Network-based model. This is done by the efficient extraction of all neighboring nodes lying in the graph network. The proposed model, which is a deep neural network, consists of 3 layers and one output layer, and it is built on a GCN layer, as depicted in Figure 1.

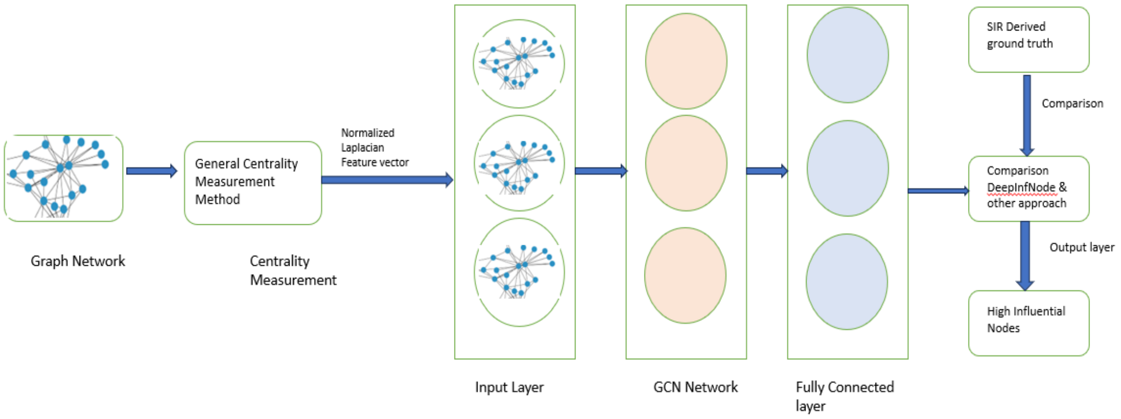


Fig. 1: Proposed framework for influential Node identification

The proposed model is created when these layers are incorporated one by one. Normalization is done at the input layer. At the data input layer, for each node in the network, we create the Laplacian and feature vector (Yu et al., 2020). Some traditional centralities that reflect the node's structural qualities are considered node features in the research study. The centralities like Betweenness centrality, degree centrality, K-shell centrality, or density centrality have been used and discussed in the later stage of the paper. Additionally, feature normalization is done to prevent overfitting. The normalization is defined by Equation 7 below:

$$F_k = \frac{p_k}{N} - 0.5 \tag{7}$$

Where Fk is the node's ranking order and centrality characteristics, k value. The rank of a node in a network is specified by the centrality attribute k. The number of nodes in the network is denoted by N. For each feature, the normalization ranges from -0.5 to 0.5, whereas the standardization contains the same value.

The graph convolutional network (GCN) layer is a semi-supervised learning technique (Zhang et al., 2023). It is used for getting a node vector with the utilization of the graph structure and feature vector. The layer is represented by Equation 8:

$$H^{l+1} = \sigma(AH^l\omega^l + B^l) \tag{8}$$

Nodes are represented in the GCN layer as:

a) Asymmetric normalized Laplacian matrix
b) Trainable weights
c) Bias
d) Nodes

The non-linear function is described with Exponential Linear Unit (ELU). Neighboring nodes of the input layer are described as feature vectors (H0). To make the optimum utilization of Node characteristics, a skip connection is added to the GNN layer. To avoid overfitting, the dropout technique is used.

Architecture Diagram and Explanation

The GCN network works from top to bottom, as shown in Figure 3. The data processing starts from the input layer and ends at the output layer, as illustrated in Figures 2 and 3. For each of the nodes in the network, the feature is extracted from its neighboring nodes and its own feature. The average function has been used in our research work for each node in the network. As a result, the average value is obtained for each node in the network. Once the average value for each node in the network is obtained, the resulting vector is created by passing the value to the neural network. The output of the first layer is considered the input for the second layer. GCN, being part of a semi-supervised learning approach on the graph, uses both Node features and structure for data processing. The weighted average for all adjacent nodes and the node itself is computed to make the degree and feature vector. Here in Figure 3, I0 & I1 are the input layers, H0, H1 and H2 are the hidden layers, whereas O1 and O2 are the output layers.

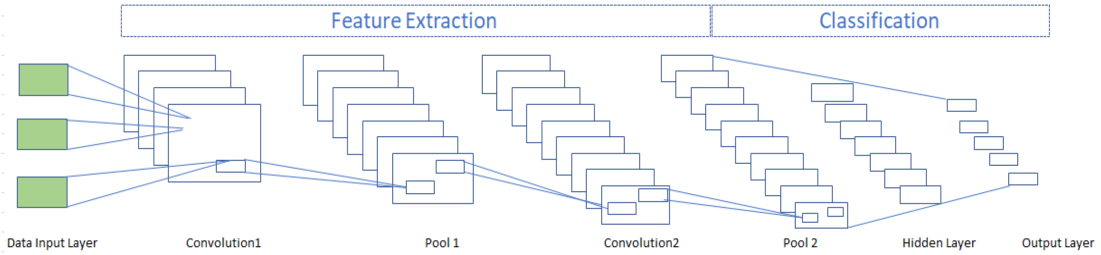


Fig. 2: Proposed Architecture Diagram for influential Node identification

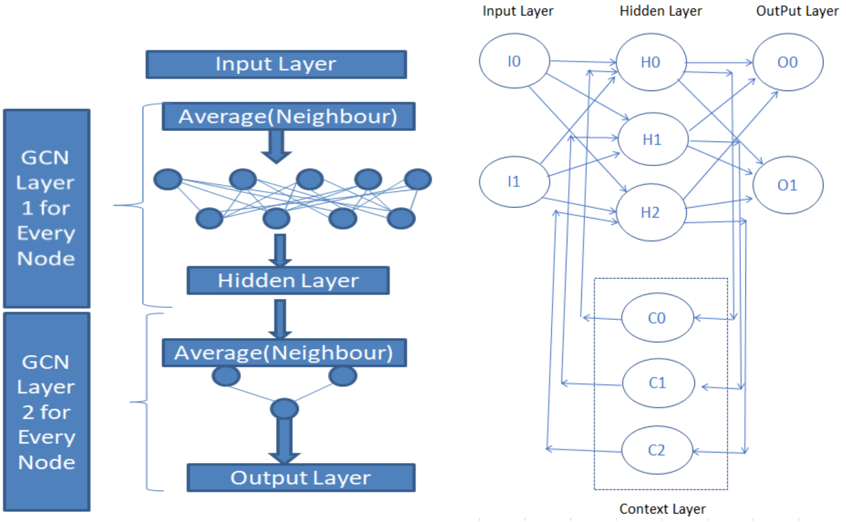


Fig. 3: Proposed framework for influential Node identification

Fully Connected Layers

The fully connected Layers are completely interconnected. These task learning layers are supposed to be fully connected after the GCN. The FC layer is followed by ELU's Nonlinear function. To avoid overfitting, the dropout technique is used for the first two fully connected layers.

Output Layer

In the final step, a fully linked layer provides output to the loss function (Log-SoftMax). The Susceptible-Infected-Recovered (SIR), which is a combination of Susceptible (the probable), Infected(confirmed), and Recovered (removed from the population of infection), is a fundamental framework to study the spread of infectious disease. The SIR model is used to provide a baseline against proposed classified findings, optimizing NLL loss, which is considered a fundamental loss function in classification and probabilistic models.

NLL loss is defined in Equation 9 as:

$$NLL = -\sum_i log(\hat{y}_i) \tag{9}$$

Methods

Preprocess the Proposed Model

Deep learning is a subset of machine learning that requires very large amounts of data in terms of volume. Smaller or medium-sized data networks have fewer nodes in them; therefore, the proper classification of data is not possible using a deep learning algorithm (Zhang et al., 2019). Hence, as per the requirement of the deep learning algorithm, large-sized network data is taken along with the proposed model. The large dataset is processed using transfer learning technology, where data is reused on different tasks. The generalization of the graph is done as part of the pre-tuning activity. The whole dataset is distributed into two parts: Training data and testing data. The model is first trained using training data, and further, the actual execution of the model is done with the testing data (Zhao at al., 2020). The execution process has been discussed as a plotted diagram in Figure 1. After the implementation of testing data, network nodes are predicted by the model. The algorithm for the DeepInfNode proposed framework is described in Algorithm 1. Additionally, the notation is described in Table 1.

Table 1: Sample notation with description of notation

Sample Notation	Description of Notation
G	Graph Network
G = (V, E, X)	Input Graph data set
N	Number of Nodes in the network
H_{ln}	Influential nodes with a higher rank
l	Laplacian normalized graph
s_{ln}	Set of influential nodes

Algorithm 1: Identification and generation of influential nodes ln from Graph Network

```
Data Input: Graph Network G
Data Output: Predicted high-ranked Influential Nodes $H_{ln}$

Step 1: Load the graph network dataset G = (V, E, X) as data input
Step 2: Initialize feature vector. Do pre-processing of the dataset.
Step 3: For each i ∈ N do
            Using graph input data and equations 1,2,3, and 4, calculate the neighbor feature vector
            Generate a symmetric normalized Laplacian L before the data into the GCN Model
Step 4: End
Step 5: For each l ∈ L do
            Node feature evaluation layer-wise to obtain the node feature
            Usage of Dropout to prevent overfitting by equation 8
Step 6: End
Step 7: Generate a set of influential nodes $s_{ln}$, taking the result from the FC layer and putting it into the Log Softmax * classifier
Step 8: Evaluate loss function L
Step 9: Compare model output
* Log-Softmax is used to solve the classification problem
```

Methodology Applied for Creating the Dataset

The experimental dataset is always complicated and large, but for the majority of cases, the label of influential nodes is always missing. Hence, to overcome this issue, the Susceptible Infected Recovered (SIR) model is applied for influence node simulation. A similar approach has been applied to other research activities as well. However, through the literature review, we have observed that the quantification of the rate of infection in the SIR model is missing. A metric is designed for the selection of the infection rate with the introduction of discrimination. SIR test and other infection rate tests have a major effect on the identification of node influence (Zohdi et al., 2022). For the real-time data analysis, in case of an infection breakout, the sample is either too small before the breakout, or it is too huge in case of a breakout. A similar observation is seen in the case of nodes as well. The number of such nodes is either too small or too high in number for the case of an infection breakout. Therefore, instead of the epidemic threshold, discrimination is used. Discrimination is improved by increasing the SIR value. D is the common discrimination index and is denoted by Eq. 9:

$$D = \frac{CH - C_l}{N(H-L)} \tag{10}$$

Where CL High and low influence group total influence capacities:

H = most significant capacities
L = least significant capacities
N = Proportion of the highest importance group

The Common discrimination index D is computed for each node to identify the influencing capacity using the SIR simulation. From the SIR, the infection rate is used to compute the most significant and influential node in the network. The node having the highest value is the most influential in the network.

Setup for Experimental Analysis

To experimentally establish the hypothesis, we performed the experimental analysis, for which we took real-world network datasets available on Kaggle. To establish the diversity of hypotheses, experimental data are taken from different domains and of different sizes. Nodes and Edges of all four datasets have been summarized in Table 2. The social media data is the largest, whereas the bank data is the smallest in size. Additionally, the success criteria are defined to quantify the result. To do a comparative analysis of the proposed framework, machine learning algorithms, neural network algorithms, and centrality-based algorithms are used.

Table 2: Summary of Benchmark different experimental datasets

Dataset	Type of Network	#Nodes or Vertices	# Edges in Network
Facebook Data	Social Network	4581	88247
Twitter-Dataset	Social Network	4879	77589
Credit card fraud detection	Banking Network	2261	7729
Disease prediction using machine learning	Medical network	3624	63587

Data Set Preparation

Experimental analysis is conducted on four different datasets available on Kaggle of different domains, Facebook Data, Twitter-Dataset, Credit Card Fraud Detection, and Disease Prediction using machine learning. For the experimental analysis, the dataset is chosen from different domains, which means two datasets from social media, one from the banking domain, and one from the healthcare domain, so that the said hypothesis can be established for different domains. Additionally, these datasets are available in the public domain:

Data source link
Facebook dataset: Facebook Data
Twitter Dataset: Twitter-Dataset
Credit card fraud detection: Credit Card Fraud Detection
Disease prediction using machine learning: Disease Prediction Using Machine Learning

The data statistics for all four experimental datasets are mentioned in Table 2, whereas the Optimal Infection Rate (OFR) for various data networks has been mentioned in Table 3, which helps in the identification of influential nodes on these networks.

Table 3: Eigen Value and Eigen Vector

Eigenvalues:
[1.66687898e+07 -4.42582859e-10 5.50528165e+02 8.33592559e+02]
Eigenvectors:
[[7.07097100e-01 -7.07106781e-01 -3.70024996e-03 3.59154714e-06]
[7.07097100e-01 7.07106781e-01 -3.70024996e-03 3.59154714e-06]
[5.22874886e-03 -2.95063223e-15 9.99222628e-01 3.90742860e-02]
[-2.09548804e-04 -3.64439175e-16 -3.90737245e-02 9.99236308e-01]]

- ➢ Facebook Dataset: It is a 99003 X 15-dimensional data containing the fields like userid, age, dob_day, dob_year, dob_month, gender, tenure, friend_count, friendships_initiated, likes, likes_received, mobile_likes, mobile_likes_received, www_likes, www_likes_received. The dataset contains the user ID, and for each user ID, other fields are mapped accordingly
- ➢ Twitter Dataset: The dataset contains the following fields: Tweet_ID, Username, Text, Retweets, Likes, Timestamp. It is a 10000 x 6-dimensional dataset
- ➢ Credit Card Fraud Detection: it is a 284807×31 dimension data, and the principal component is mentioned in terms of v1,v2 & so on. The total number of transactions in the dataset is 284807
- ➢ Disease prediction using machine learning: It is 4921 X 133 dimension data and has fields like itching, skin_rash, nodal_skin_eruptions, and so on. The positive for disease is denoted by 1, whereas the negative for disease is denoted by 0

First of all, the complete dataset is analyzed, and the noisy data is removed from the experimental dataset. The activity is done to remove any duplicate datasets or data redundancy. The complete dataset is divided into two parts with the 80:20 principle, which is also known as the holdout distribution. This strategy is applied so that we get the training data and testing data for the model defined. The proposed model is trained with the training data and tested with 20% of the data for experimental analysis. Splitting graph data helps in node subset or subgraph identification and assigning a weight to each node in the network.

The initial perception is that around 10% of the nodes are the most influential nodes in the network, whereas 90% of the nodes are not influential in the network. As per our hypothesis, we have taken 5% of the nodes as the most influential nodes in the network, despite 10% of the nodes generally being influential in the network. Additionally, for the ease of implementation and to experiment with a more realistic scenario, the nodes with a degree higher than 3 are taken into consideration.

Evaluation Criteria

The verification and validation criteria have been defined for the proposed framework. Predictive performance and hyperparameters are the two metrics for verification purposes for the proposed framework. The predictive framework uses the predictive performance for Area under Curve (AUC) for DeepInfNode, whereas the hyperparameter checks the deviation of accuracy of prediction on different hyperparameters.

Assessment Process and Deployment Details

The comparative study has been done between the proposed framework and other traditional baseline models, such as:

- Logistic regression
- Support Vector Machine (SVM)
- Rank-based convolutional network (RCNN)
- Similarity-based Graph Neural Network (SGNN)
- Graph Convolutional Network (GCN)

These methods are a node structure-based framework and work for the identification of the most influential node in the given dataset. The node characteristics are the main factor for any method used in machine learning. In the current research, Centrality-based methods like Betweenness centrality, density centrality, and k-shell centrality are used to measure and describe the content. These centrality-based methods are the general and traditional centrality-based mechanisms. Interconnection between the nodes in the network is examined through these centrality-based approaches and is widely used by researchers. These centrality-based approaches provide us with the nodes that are most influential in the network, and they are based on the node value. Precision, Recall, and F1 are the measures that certify the efficiency of centrality-based techniques. The AUC metric cannot be created through centrality-based techniques because of its formulation. For the current framework implementation, a fixed-size neighbor network is designed with a sample size of 50 and 100, depending on the size of the dataset. A GCN layer with 8 units exists in the proposed model. The network consists of fully connected layers, which consist of 16,8,2 units representing neural network architecture. In a fully connected layer or dense layer, each unit in one layer is fully connected to every existing unit in the next layer. The first layer (16 Units) has 16 units, and each unit has its own biases and weights. The second layer (8 units) captures a more abstract representation by reducing the number of units. The third layer (2 units) is the final layer, with 2 units corresponding to the output dimension. Further, less significant nodes are removed from the network for accuracy and to narrow down the research process. A similar process is applied to all the nodes in other datasets as well.

Results and Discussion

According to the requirement, the whole dataset is divided into two parts, i.e., one as training data and one as testing data, and the data proportion maintained is 80% and 20%, which means 80% of the dataset is kept as training data and 20% dataset is kept as testing data for each dataset network. Since the nature of the data in all four data networks is different, the algorithm is trained for all four datasets with the training data. Further, the algorithm is tested on the remaining 20% dataset. Table 3 shows the eigenvalue and eigenvector for the Twitter dataset. Whereas Table 4 shows the Loss and Test Accuracy for a single EPOCH for 50 iterations.

Table 4: Loss and Test Accuracy for single EPOCH

EPOCH	Loss	Test Accuracy	EPOCH	Loss	Test Accuracy	EPOCH	Loss	Test Accuracy
Epoch: 20	Loss: 5.0387	Test Accuracy: 0.0100	Epoch: 180	Loss: 3.9814	Test Accuracy: 0.0700	Epoch: 340	Loss: 3.6785	Test Accuracy: 0.0900
Epoch: 40	Loss: 4.4082	Test Accuracy: 0.0000	Epoch: 200	Loss: 3.9419	Test Accuracy: 0.0900	Epoch: 360	Loss: 3.6443	Test Accuracy: 0.0900
Epoch: 60	Loss: 4.2808	Test Accuracy: 0.0100	Epoch: 220	Loss: 3.9030	Test Accuracy: 0.0900	Epoch: 380	Loss: 3.6103	Test Accuracy: 0.0900
Epoch: 80	Loss: 4.2109	Test Accuracy: 0.0400	Epoch: 240	Loss: 3.8641	Test Accuracy: 0.0900	Epoch: 400	Loss: 3.5766	Test Accuracy: 0.1100
Epoch: 100	Loss: 4.1538	Test Accuracy: 0.0200	Epoch: 260	Loss: 3.8254	Test Accuracy: 0.0900	Epoch: 420	Loss: 3.5430	Test Accuracy: 0.1300
Epoch: 120	Loss: 4.1048	Test Accuracy: 0.0400	Epoch: 280	Loss: 3.7869	Test Accuracy: 0.0900	Epoch: 440	Loss: 3.5094	Test Accuracy: 0.1400
Epoch: 140	Loss: 4.0621	Test Accuracy: 0.0400	Epoch: 300	Loss: 3.7497	Test Accuracy: 0.0900	Epoch: 460	Loss: 3.4758	Test Accuracy: 0.1300
Epoch: 160	Loss: 4.0212	Test Accuracy: 0.0700	Epoch: 320	Loss: 3.7133	Test Accuracy: 0.0900	Epoch: 480	Loss: 3.4423	Test Accuracy: 0.1400

The Twitter dataset is put under experimental analysis, and different plots are drawn in two vertical plots:

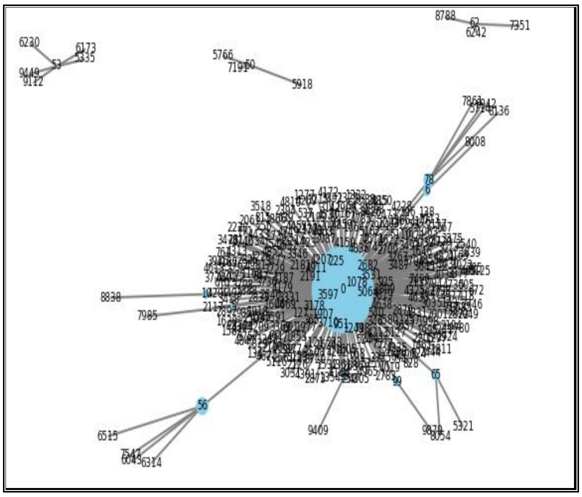
a) Traditional Approach
   1. Betweenness Centrality: Plot No 4.1
   2. Degree Centrality: Plot No 4.2
   3. Closeness Centrality: Plot No 4.3
   4. Eigenvalue & Eigen Vector: Plot No 4.4
   5. ROC Plot: Plot No 4.5
   6. Top 100 retweets on Tweet ID: Plot No 4.6

b) GNN Approach
   1. Dataset Graph Visualization: Plot no 5.1
   2. GAT Attention Weight with Node ID: Plot No 5.2
   3. GNN Node Activation Heat Map with Node ID: Plot No 5.3
   4. GNN Embedded Projections: Plot No 5.4
   5. GNN Embedded Visualization (Influential Node Highlighted): Plot No 5.5
   6. GNN Embedded Visualization: Plot No 5.6
   7. ROC Curve Different class: Plot No 5.7
   8. GNN Embedding – Most Influential Node: Plot No 5.8

Plot Analysis: Twitter Dataset

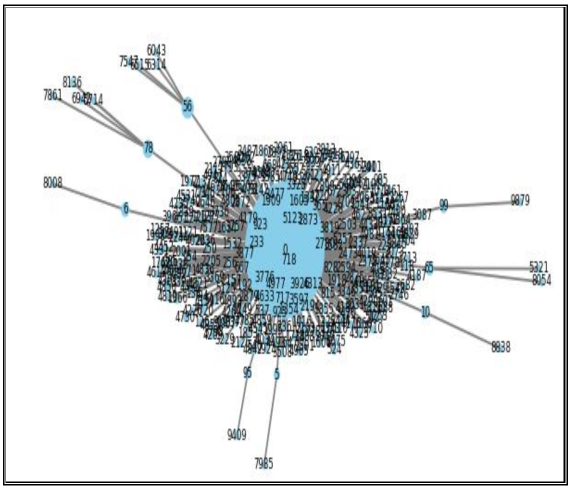
The following are the different plots generated and analyzed in Figure 4: Twitter Dataset – Traditional Approach.

Following is the extract and analysis for the GNN Approach in Figure 5: for the Twitter Dataset.



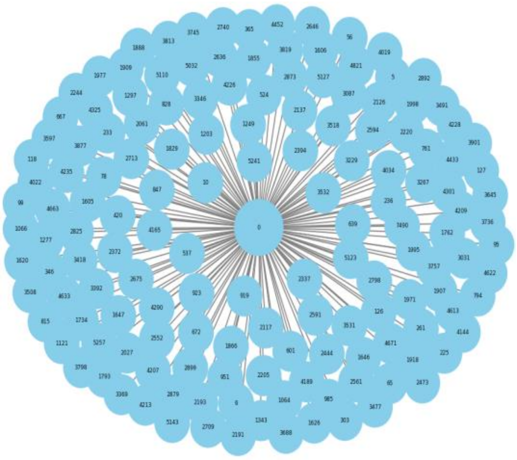
Betweenness Centrality

Fig. 4.1: Betweenness Centrality Degree



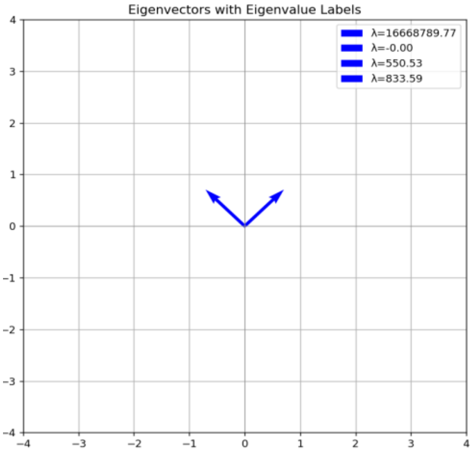
Degree Centrality

Fig. 4.2: Degree Centrality



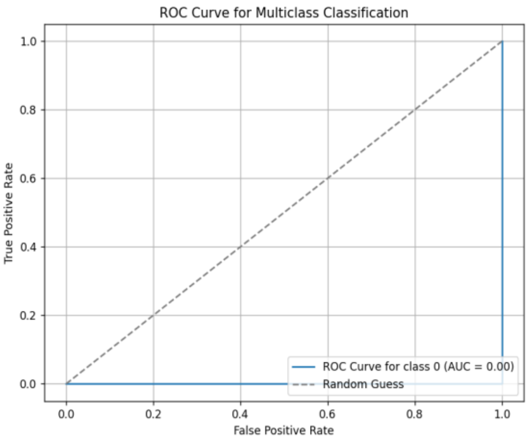
Closeness Centrality

Fig. 4.3: Closeness Centrality



EigenValue, and EigenVector

Fig. 4.4: Eigenvalue and Eigenvector



ROC Curve

Fig. 4.5: ROC Curve for Twitter Dataset

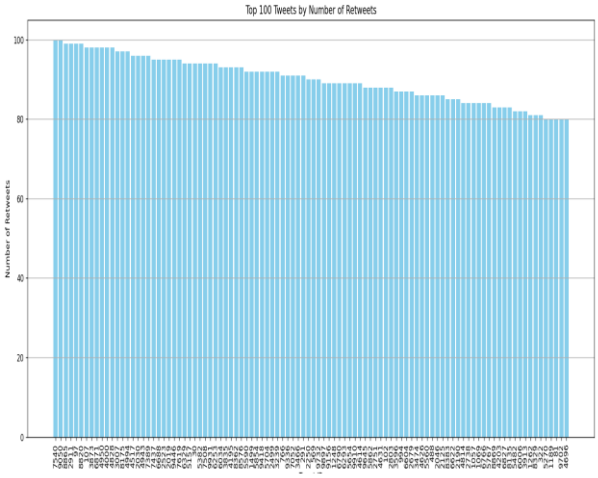


Fig. 4.6: Top 100 Retweets by TweetId

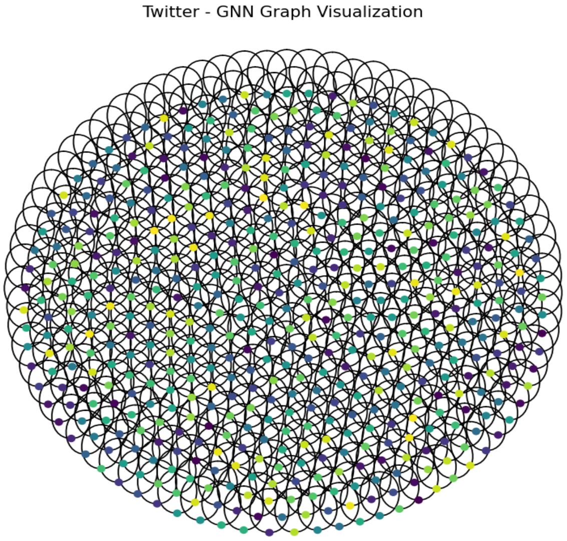


Fig. 5.1: Dataset Graph Visualization

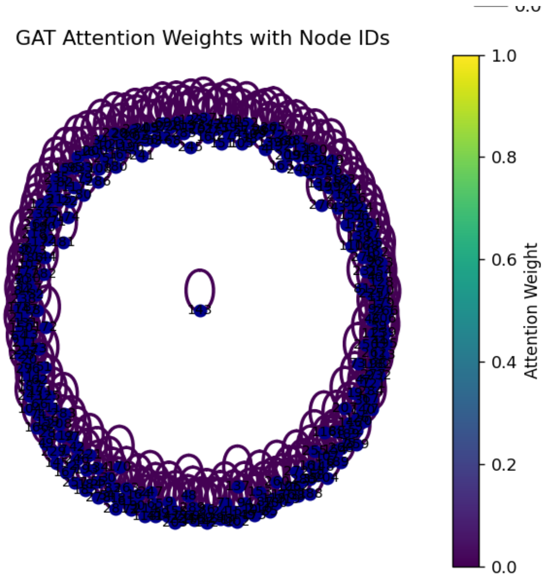


Fig. 5.2: GAT Attention Weight with Node ID

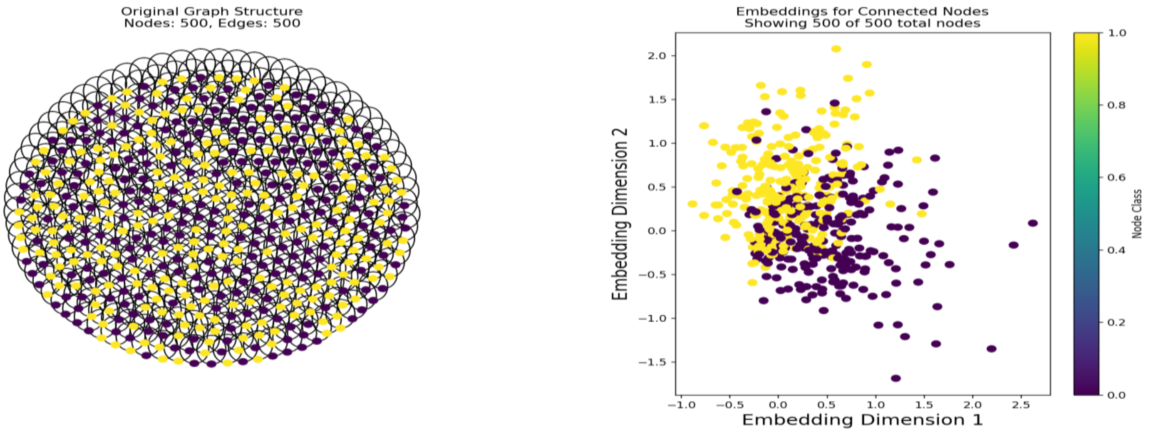


Fig. 5.4: GNN Embedded Projections

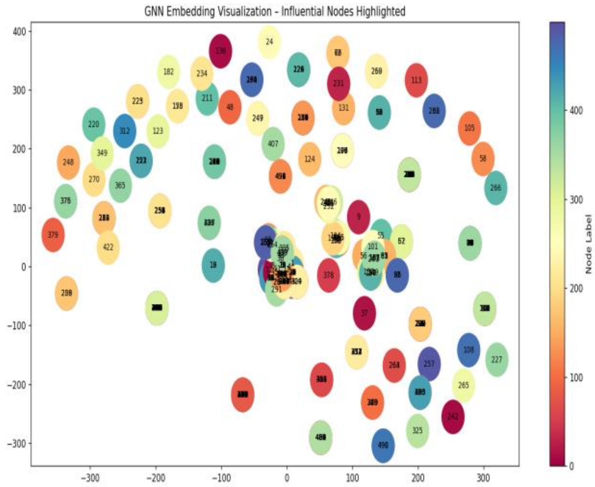


Fig. 5.5: GNN Embedded Visualization (Influential Node Highlighted)

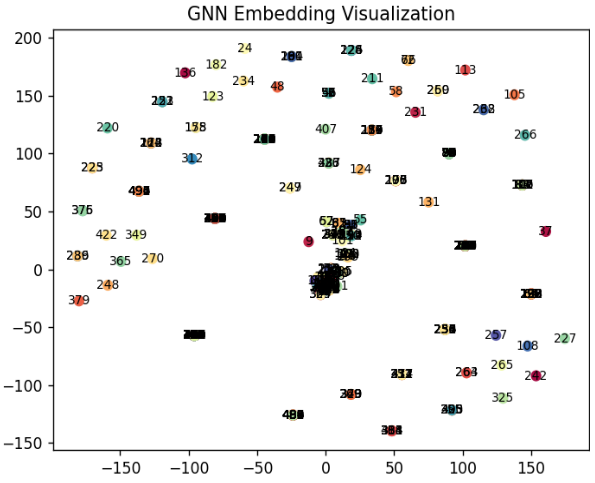


Fig. 5.6: GNN Embedded Visualization

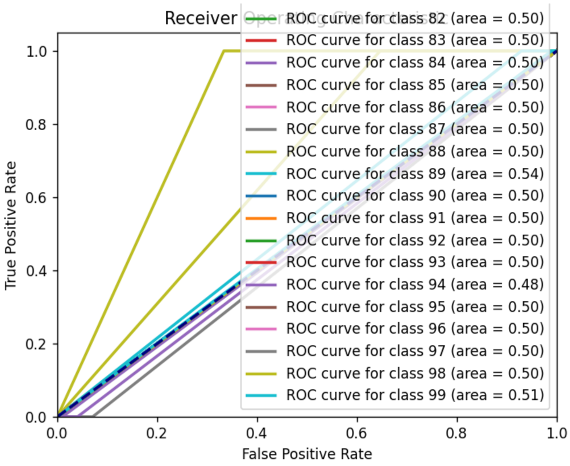


Fig. 5.7: ROC Curve Different class

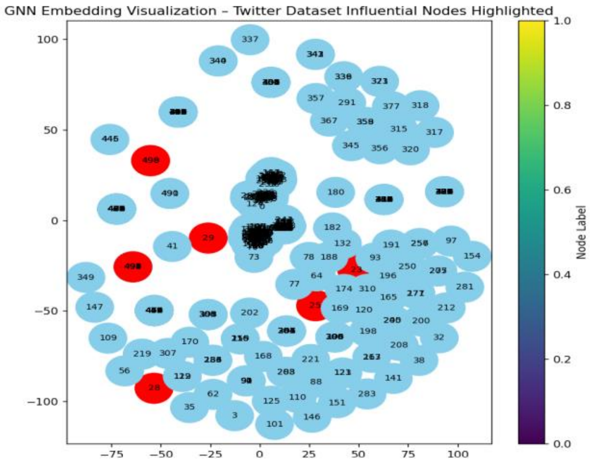


Fig. 5.8: GNN Embedding – Most influential Node

Traditional Approach Outcome

The betweenness centrality helps in identifying how often one node lies between the shortest paths with other nodes in the network. The plot in Figure 4.1 shows the betweenness centrality and strong influence in the network. The larger nodes with deep blue indicate the strong influence of the node in the network. The nodes 0, 56,65,78,3013,3073 are the highly influential nodes in the network.

The degree centrality counts the direct connection of any particular node with other nodes in the network. It identifies the most connected nodes in the network. From the plot in Figure 4.2, it is evident that the nodes 0, 56, and 78 are the most connected nodes in the network and have the highest degree centrality value.

The closeness centrality helps in identifying how quickly one node can reach other nodes in the network. The smaller the closure distance, the higher the closeness centrality. From the plot in Figure 4.3, it is evident that node 0 is in the center and has the highest closeness centrality value.

Eigenvalues and eigenvectors help in understanding transformation, dimensionality reduction, and graph-based learning. The plot in Figure 4.4 shows four eigenvalues and two eigenvectors. The high eigenvalue denotes the dominant direction of variance, whereas two moderate eigenvalues show a moderate influence but not a dominant one. The zero eigenvalue shows no direction. Additionally, the eigen vector is directed towards (-1,1) and (1,1).

The ROC plot is a graphical representation to evaluate the performance of a binary classification model across different threshold values. ROC plot is done in Figure 4/5, and the value suggests that it is failing completely. It is yet worse than random guessing, where AUC is 0.5. Yet another interpretation from the plot is that the model might be confused between positive and negative values.

The plot in Figure 4.6 suggests that the top 100 tweets with the retweet count suggest that the dataset consists of a mixed number for retweet count for different tweet IDs. As we move from right to left towards the y-axis, the number of retweets increases. We see the cascading effect on the remaining tweets in the data network because of the high tweet count.

Deep Inf Node Via GNN Approach Outcome

The GNN Approach is implemented for the identification of influential nodes. The plot in Figure 5.1 is considered the first plot in the GNN approach, which shows the different node colors showing categorical differentiation, dense connectivity, and network structure. The high-density nodes in the center show the highly influential nodes in the network.

The plot in Figure 5.2 shows the circular plot displaying attention weight for different nodes in the network. Attention weight close to light yellow color (1.0) shows a strong connection, whereas the attention weight close to dark purple color (0.0) shows a weak connection. Most of the nodes in the network lie close to low attention weights, whereas a few nodes show high attention weights, which help in node classification or link prediction in the GNN approach.

The Activation heatmap plot shown in Figure 5.3 shows the activation strength in the GNN plot. The purple color shows the low activation strength, whereas the green or yellow color shows the high activation strength for the nodes. The plot shows the nodes with strong influence in the network, with the green or yellow color helping identify the most influential node in the network.

The projection plot, Figure 5.4, shows the dimensionality reduction from multi-dimension to two dimensions as Embedding dimensions 1 and 2. The cluster of points in the embedding dimension plot shows the grouping of data within the plot. The color gradient shows the cluster of data within the plot. The yellow nodes show the highly embedded space, whereas the purple nodes show the low embedded space. In summary, the plot helps in identifying the distinct classes and their relationship.

After the dimensionality reduction, the next plot is done for the Influential node highlighted in Figure 5.5 and Figure 5.6. The scatter plot shows the different nodes based on their influence value. The plot helps with cluster analysis, node influence, and quality. Eventually, it helps in understanding the integral relationship between different nodes and influence within the GNN framework. The second scatter plot shows the highly influential node based on the value. The plot shows the cluster and outlier. The layout provides insight into the relationship between the nodes.

The receiver operating characteristics provide the curves for different classes in the model, as shown in the plot in Figure 5.7. The False Positive and True Positive are plotted on the x-axis and y-axis in the plot showing the proportion of actual negative and actual positive. The AUC plot 0.50 indicates the model will perform as random guessing for class 82, whereas the other value 0.48, indicates the model will perform slightly less than random guessing for class 88. The AUC plot of 0.51 indicates the model will perform slightly better than random guessing for class 91. Conclusively, the plot for the said dataset indicates that none of the classes displays a strong predictive capability.

The two-dimensional scatter plot, as shown in Figure 5.8, shows the highly influential node in the Twitter dataset. The cluster of nodes is available in the center of the plot, indicating lower structural importance. The circles marked in red are of high influence and high importance. The plot indicates the significance distribution of influence in the dataset.

Performance Prediction

The infection rate is a learned proxy for influence resulting from the simulation of how information spreads through a network using the SIR model. The SIR simulation model is executed multiple times on the available dataset, infecting its neighbors starting from each node. The data for infection rates is captured and tabulated in Table 5. From the table data 5, the infection rate for online social media is highest, whereas for disease prediction, it is lowest. The values 0.13 and 0.17 are considered moderate, not aggressive, but not candidates to be ignored.

Table 5: Infection rate for different datasets

Network	Highest Value of D	Infection Rates
Facebook Data	0.458	0.13
Twitter-Dataset	0.359	0.17
Credit Card Fraud Detection	0.473	0.11
Disease prediction using machine learning	0.327	0.09

The value for Precision, Recall, F1, and AUC is captured for all four datasets and has been tabulated as metric data in Table 6. The plot for the same data is plotted in Figure 6 as an evaluation methodology. Observations are taken for all four datasets under the scope of the validation activity. These are taken for a fixed set of evaluation metrics for DeepInfNode and other general evaluation methodologies. From the raw data in Table 6 and Figure 6, it is evident that the stats received for DeepInfNode are impressive and encouraging as compared to other methodologies, which establish the superiority of the DeepInfNode method over other baseline methods. It is evident that, for most cases except recall, DeepInfNode is superior for all four datasets. In an ideal case, when precision increases, recall decreases, and the same is evident in Table 6, which proves the authenticity of the dataset obtained after experimental analysis.

Table 6: Summary of results from DeepInfNode and other approaches for influential node identification

Dataset	Evaluation Metrics	Evaluation Methodology								
		DeepInfNode	BC	LR	k-shell	SVM	DNC	RCNN	GCN	SGNN
Facebook Dataset	Precision	92	15.8	87.7	28.4	89.4	89.6	88.8	91.3	91.2
	Recall	73.9	16.5	60.2	17	75.7	64.1	61.9	80.4	74.8
	F1-Score	89	13.2	83.5	13.1	82	77.9	73.8	84.7	82.3
	AUC	97.4	68.5	90.9	37	96.4	92.8	91.3	92.9	91.9
Twitter Dataset	Precision	94	65.1	91.9	37.3	91.6	89.8	91.7	91.2	88.7
	Recall	88.8	12	79.9	15.2	90.8	69.9	70.3	90.9	88.2
	F1-Score	91.8	17.9	81.8	17.4	89.7	76.6	80.3	91.3	91.3
	AUC	99.5	69.5	95.8	48	97.7	94.9	89.8	97.7	93.5
Credit Card Fraud Detection	Precision	92.1	78.9	81.8	67.6	86.7	86	89.8	89.4	85.9
	Recall	87.8	40.1	45.6	19.3	85.1	67.9	89.9	75.7	79.7
	F1-Score	93.2	37.9	53.2	30.7	86	72.9	84.3	82	85
	AUC	99.7	81.8	94.6	71.9	92.1	92.7	94.1	92.1	93.4
Disease prediction using machine learning	Precision	97.7	60.4	87.9	58.8	94.9	87	89.5	97	92.3
	Recall	88.1	26	67.9	57.1	86.3	73.8	58	85	88.9
	F1-Score	92.8	37.1	75.1	42	87	81.9	71.9	91.9	89.5
	AUC	99.3	58.5	96.2	62.1	92.9	94.9	83.8	96.7	95.7

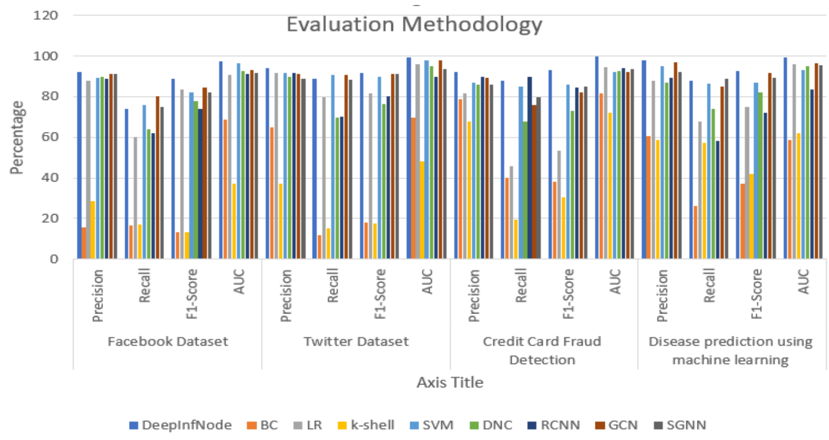


Fig. 6: Evaluation Methodology

The best result from DeepInfNode is marked in Bold and light Green, whereas the best result from the generalized evaluation technique is marked as bold and orange.

The proposed model seems to be more effective and encouraging for an undirected graph as compared with a directed graph. The proposed model has come up with promising results in terms of accuracy of data and performance for all datasets except the dataset used for Disease prediction using machine learning, where DeepInfNode and other baseline methods provide almost similar results. Accuracy, precision, and F1 score are highest in the case of DeepInfNode, and its results are best compared with other baseline algorithms for the Facebook dataset. Similarly, better performance has been seen in the case of AUC and F1-score for DeepInfNode when compared with other baseline methods. This proves that DeepInfNode is more efficient than other baseline methods. Different observations have been tabulated in Table 6.

From the efficiency perspective, the machine learning approach stands in 2nd position, whereas the centrality-based approach stands in 3rd position. Among the different centrality-based approaches, the best performance is shown by degree centrality because of its simplicity, direct influence, and its wide application. DC and SVM have the optimum observation for F1-score and Precision for social media data like the Twitter dataset and Facebook dataset. Vertex properties are still valid and useful for the suggested model for calculating k-shell and betweenness centrality.

The suggested model, DeepInfNode, still performs better than machine learning based models. The same has been confirmed in the stats gathered in Table 6. Since DeepInfNode uses GCN structure and node features, it performs better than machine learning algorithms that use node features only. A lot of baseline centrality-based approaches are applied during the validation activity, but out of these, k-shell centrality approaches are found to perform well; the rest of the centrality-based approaches are not up to the mark.

Even a machine learning based approach using structural features performs better than a centrality-based approach. Functional importance in the case of centrality-based methodology is not fully utilized. The DeepInfNode has the accuracy percentage for all the datasets available, like Facebook, Twitter, and Credit Card Fraud Detection, except for Disease prediction using machine learning. The same stats have been tabulated in Table 4 as well. GCN and SGNN perform better among all baseline methods in the majority of cases. Almost similar trends for the F1-score are seen for all the available datasets used in experimental analysis. The propagation strategy and deep graph neural network illustrate that the observation for F1-Score for DeepInfNode is always better for different datasets, like the Facebook dataset (89%), Twitter Dataset (91.8%), Credit Card Fraud Detection (92.1%), and Disease prediction using machine learning (92.8%).

These are the most significant data that have been stored in tabular format, and these observations indicate that the DeepInfNode is the best technique out of the eight evaluated techniques and can be considered as the starting point. The AUC value for Facebook data for DeepInfNode is 97.4%, whereas GCN is 92.9%. The AUC value for Twitter data for DeepInfNode is 99.4%, whereas GCN is 97.7%. The AUC value for Credit Card Fraud Detection data for DeepInfNode is 99.7%, whereas GCN is 92.1%. The AUC value for Disease prediction using machine learning for DeepInfNode is 99.3%, whereas GCN is 96.7%. The stats clearly indicate the DeepInfNode model, in which both structural and node attributes are the best technique out of all baseline evaluated techniques.

Attribute Analysis

The critical hyperparameters like learning rate, batch size, regularization parameters, or number of layers have been examined in the current section, along with the impact of these parameter selections on the efficiency of the deep learning model. Graph-based network size is one of the critical hyperparameters. Figure 7 shows the performance of the skip connection (AUC and F1-score) from 10-90 for the proposed model.

Figure 7 below depicts the performance impact of Skip Connection (SC).

The following is the observation as per Figure 7:

a) For the Facebook dataset, AUC and F1-score remain unaffected by the size of the closest network
b) A fixed pattern of inclination and then decline is seen for AUC & F1-Score for the Twitter dataset and credit card fraud detection
c) For the Disease prediction using a machine learning dataset, before getting into the stabilization state, it first decreases and then increases. It’s the size of the neighboring network that impacts the effectiveness of the model. Further, the purpose is to optimize the performance of the model, possibly by shrinking the adjoining network. Hence, the study started with a medium-sized set of 50 or 100, depending on the data size, and then performed the required tuning activity for better performance

The optimum utilization of node features is the basis of Skip Connection (SC); hence, SC is used in the current research work as well. The impact of SC is validated on AUC and F1-score to check the performance of the proposed algorithm. As explained in Figure 7, it is evident that SC has improved the model’s performance optimally. Additionally, different stats gathered during the experimental process indicate that it is worth adding SC to the whole development process. The network consists of influential nodes as well as non-influential nodes. The number of influential nodes has an impact on the overall network. In the current context, the top 10% nodes are taken as the most influential nodes in the network. To certify the 10% most influential nodes, the validation is done for different node percentages like 5%,10%,15%,20%, and 25%. Though 5% of the most influential nodes are very optimal, the coverage of this node percentage is very low. Hence, we settle on the top 10% of the most influential nodes in the network for our research work.

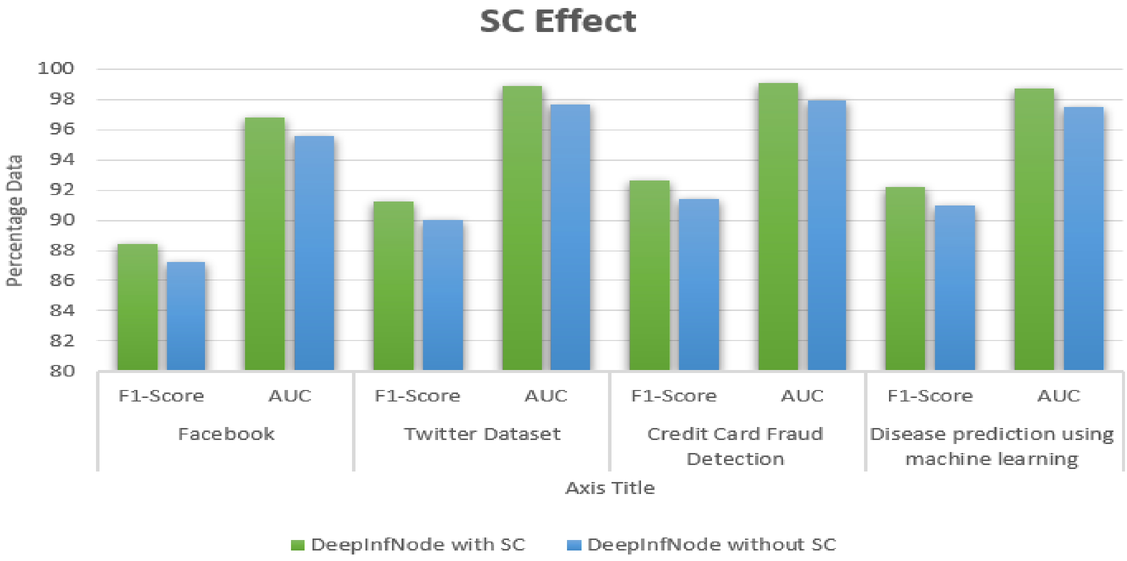


Fig. 7: Skip Connection effect

The best observation is marked in bold.

According to the hypothesis, F1 Score and AUC are computed for all four datasets for the top 5,10,15,20, and 25% respectively, and observations were captured in Table 7. For the Facebook dataset, the maximum F1-Score% captured is 86.9 for 5%, and the AUC is 95.3% for 10% of influential nodes. For the Twitter dataset, the maximum F1-Score% is 91.9 for 10%, and the AUC is 93.5 for 15% of influential nodes. For Credit Card Fraud Detection, F1-Score% is 93% for 5% and 97% for 10% of influential nodes. For the Disease Prediction dataset, F1-Score% is 91.9 for 5%, and AUC is 89.8 for 10% of the influential nodes. Since 5% of the sampling is too small to conclude the result, as mentioned above, as well as 10% of influential nodes are taken into consideration for identifying the most influential node in the dataset network.

Computational Cost

In order to identify the most influential node in the network using attribute information, the centrality measure is used by the proposed method through the GCN layer, and the evaluation of the influence node is done for the infection rate under SPR.

The best computation is marked in bold. The infrastructure used for validation activity is high-end commodity hardware consisting of Core i5, 8GB RAM, on a client-server architecture deployed on cloud infrastructure. The execution is done on a Linux machine. Python libraries, along with MATLAB, are used to develop Deep Learning models. Execution time for each model is shown in Table 8. All the executions are done on the same hardware infrastructure, and execution time is noted and tabulated in Table 8. The worst execution time for the Twitter dataset is for K-Shell, and the time is 6.94 secs. Whereas the best time taken is for DeepInfNode, and it is 1.56 and 1.62 seconds, respectively, for the Facebook and Twitter datasets. The timing mentioned in Table 6 is in seconds. Conclusively, we can say the proposed DeepInfNode, SGNN, and GCN are better-performing models than other traditional benchmark techniques.

Other Implementation Scopes

Since the proposed model is accurate, performance-intensive, and better than other traditional models, it can be used for doing analysis for any type of data in any domain. Also, currently, the static dataset is in the scope of experimental validation; the online data available on social media can also be in the scope of future activity.

Table 7: Below shows the Effect of influential node percentage on F1 & AUC

Dataset		Influence node%	F1-Score %	AUC %
Facebook dataset	Top	5%	86.9	93.9
	Top	10%	81	95.3
	Top	15%	79.3	89.2
	Top	20%	75.9	84.1
	Top	25%	72.7	82.3
Twitter Dataset	Top	5%	91.2	92.6
	Top	10%	91.9	95
	Top	15%	85.9	93.5
	Top	20%	79.9	87.8
	Top	25%	76.1	76.1
Credit Card Fraud Detection	Top	5%	93	96.4
	Top	10%	87.8	97
	Top	15%	83.6	92.9
	Top	20%	81.4	90.8
	Top	25%	79.6	87.6
Disease prediction using machine learning	Top	5%	91.9	85
	Top	10%	73.2	89.8
	Top	15%	66	77
	Top	20%	63.1	71.9
	Top	25%	61.1	68.2

Table 8: Below shows the computational cost of various implied methods

Method	Time (Facebook)	Time (Twitter)
Betweenness	4.56	6.32
K-Shell	4.91	6.94
DNC	5.73	6.84
LR	2.72	2.94
SVM	1.92	2.01
RCNN	2.53	1.81
GCN	1.58	1.75
SGNN	1.62	1.72
DeepInfNode	1.56	1.62

Conclusion and Future Scope

From the literature review, it is evident that the number of the most influential nodes in a large network is low, and the identification of such nodes is not easy. Additionally, it is critical to find the most influential and important node in the network based on its usage. The end goal is to find the most influential node in the complex network, and the same has been discussed and investigated in the current research paper. The traditional method, like machine learning algorithms or centrality measurement methods, considers network topology or node properties, making the work complex and limiting its effectiveness. To resolve these issues, the node identification problem is converted into a classification problem, and the DeepInfNode graph learning method is used for the categorization of the best influential nodes in the network. To validate and certify the methodology, the proposed approach is tested with four real-time datasets consisting of two online social media datasets, one for banking system data, and one for healthcare data. The result generated through the proposed approach certifies that the proposed approach works better in all aspects when compared with other traditional methods. The proposed approach works fine for small networks as well as for complex networks. Overall, the proposed approach is above all the traditional approaches.

However, the DeepInfNode approach comes with concerns that need to be addressed. The current approach works with a fixed set of attributes. Adding more node attributes may result in performance issues that need to be resolved. This algorithm should be validated for more complex networks than those currently used. The algorithm should be validated for weighted graphs and multilayer networks, which will increase the complexity of the network. The proposed method is validated for graph-based neural networks, but it should be validated for other types of neural networks, like FNN, CNN, or RNN, for influential node identification in the social network, which is generally very complex in nature. Though the proposed method is validated for social media or healthcare, we are still left with multiple domains, like financial, airline, and so on, where this algorithm needs to be validated. Additionally, the proposed approach is to be incorporated with other traditional methods for influential node identification. Incorporating graph mining and deep learning approaches with the proposed approach is another very important aspect that can be explored. Identification of influential nodes along with the dominant influential spreader is another scope of study that can extend current research work.

Acknowledgment

The authors are really grateful to Amity University Noida for providing the necessary opportunity for research and development for the manuscript. Special thanks to Dr. Laxmi Ahuja, Dr. Suman Mann, and Dr. Sanmukh Kaur for their guidance and constructive feedback throughout the development of the manuscript. Finally, we appreciate the editorial team of the Journal of Computer Science for their professional handling of the review process and insightful review comments. The review comments helped a lot in improving the quality of the paper.

Funding Information

No individual or organization has assisted with funds for the current research work.

Author’s Contributions

Rajnish Kumar: Main Author, research activity done, and documentation.

Laxmi Ahuja: Guide and Mentor.

Suman Mann and Sanmukh Kaur: Co-Guide and Mentor.

Ethics

This study did not require formal ethical approval as it did not involve human or animal subjects.

Data Availability

Sample data has been taken from Kaggle, and a reference has been mentioned in the Methods section.

Conflicts of Interest

The authors hereby confirm and declare that the current manuscript does not have any conflicts of interest.

References

Ayman, R. Abd Al-Azim, N., Gharib, T. F., Hamdy, M., & Afify, Y. (2022). Influence propagation in social networks: Interest-based community ranking model. *Journal of King Saud University - Computer and Information Sciences*, *34*(5), 2231–2243. https://doi.org/10.1016/j.jksuci.2020.08.004

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79. https://doi.org/10.1016/j.neucom.2017.11.077

Chen, D.-B., Sun, H.-L., Tang, Q., Tian, S.-Z., & Xie, M. (2019). Identifying influential spreaders in complex networks by propagation probability dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *29*(3), 033120. https://doi.org/10.1063/1.5055069

Guo, C., Yang, L., Chen, X., Chen, D., Gao, H., & Ma, J. (2020). Influential Nodes Identification in Complex Networks via Information Entropy. *Entropy*, *22*(2), 242. https://doi.org/10.3390/e22020242

Ibnoulouafi, A., & El Haziti, M. (2018). Density centrality: identifying influential nodes based on area density formula. *Chaos, Solitons & Fractals*, *114*, 69–80. https://doi.org/10.1016/j.chaos.2018.06.022

Jena, K. K., Bhoi, S. K., Mallick, C., Jena, S. R., Kumar, R., Long, H. V., & Son, N. T. K. (2022). Neural model based collaborative filtering for movie recommendation system. *International Journal of Information Technology*, *14*(4), 2067–2077. https://doi.org/10.1007/s41870-022-00858-4

Khan, W., & Haroon, M. (2022). An efficient framework for anomaly detection in attributed social networks. *International Journal of Information Technology*, *14*(6), 3069–3076. https://doi.org/10.1007/s41870-022-01044-2

Kumar, S., & Panda, B. S. (2020). Identifying influential nodes in Social Networks: Neighborhood Coreness based voting approach. *Physica A: Statistical Mechanics and Its Applications*, *553*, 124215. https://doi.org/10.1016/j.physa.2020.124215

Kumar, S., Mallik, A., Khetarpal, A., & Panda, B. S. (2022). Influence maximization in social networks using graph embedding and graph neural network. *Information Sciences*, *607*, 1617–1636. https://doi.org/10.1016/j.ins.2022.06.075

Lü, L., Chen, D., Ren, X.-L., Zhang, Q.-M., Zhang, Y.-C., & Zhou, T. (2016). Vital nodes identification in complex networks. *Physics Reports*, *650*, 1–63. https://doi.org/10.1016/j.physrep.2016.06.007

Maurya, S. K., Liu, X., & Murata, T. (2021). Graph Neural Networks for Fast Node Ranking Approximation. *ACM Transactions on Knowledge Discovery from Data*, *15*(5), 1–32. https://doi.org/10.1145/3446217

Okamoto, K., Chen, W., & Li, X.-Y. (2008). Ranking of Closeness Centrality for Large-Scale Social Networks. *Frontiers in Algorithmics*, *5059*, 186–195. https://doi.org/10.1007/978-3-540-69311-6_21

Ou, Y., Guo, Q., Xing, J.-L., & Liu, J.-G. (2022). Identification of spreading influence nodes via multi-level structural attributes based on the graph convolutional network. *Expert Systems with Applications*, *203*, 117515. https://doi.org/10.1016/j.eswa.2022.117515

Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., & Tang, J. (2018). DeepInf: Social Influence Prediction with Deep Learning. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2110–2119. https://doi.org/10.1145/3219819.3220077

Rodrigues, F. A. (2019). In Network centrality: an introduction. A Mathematical Modeling Approach from Nonlinear Dynamics to Complex Systems, 22, 177–196. https://doi.org/10.1007/978-3-319-78512-7_10

Sandhya, Ghose, U., & Bisht, U. (2020). Tailored feedforward artificial neural network based link prediction. *International Journal of Information Technology*, *12*(3), 757–765. https://doi.org/10.1007/s41870-019-00362-2

Shashidhar, R., Patilkulkarni, S., & Puneeth, S. B. (2022). Combining audio and visual speech recognition using LSTM and deep convolutional neural network. *International Journal of Information Technology*, *14*(7), 3425–3436. https://doi.org/10.1007/s41870-022-00907-y

Tran, Q. M., Nguyen, H. D., Huynh, T., Nguyen, K. V., Hoang, S. N., & Pham, V. T. (2022). Measuring the influence and amplification of users on social network with unsupervised behaviors learning and efficient interaction-based knowledge graph. *Journal of Combinatorial Optimization*, *44*(4), 2919–2945. https://doi.org/10.1007/s10878-021-00815-0

Ullah, A., Wang, B., Sheng, J., Long, J., Khan, N., & Sun, Z. (2021). Identification of nodes influence based on global structure model in complex networks. *Scientific Reports*, *11*(1), 6173. https://doi.org/10.1038/s41598-021-84684-x

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Li, P., & Bengio, Y. (2017). *Graph attention networks*. https://doi.org/10.48550/arXiv.1710.10903

Xiang, Y., Fujimoto, K., Li, F., Wang, Q., Del Vecchio, N., Schneider, J., Zhi, D., & Tao, C. (2021). Identifying influential neighbors in social networks and venue affiliations among young MSM: a data science approach to predict HIV infection. *AIDS*, *35*(Supplement 1), S65–S73. https://doi.org/10.1097/qad.0000000000002784

Yu, E.-Y., Wang, Y.-P., Fu, Y., Chen, D.-B., & Xie, M. (2020). Identifying critical nodes in complex networks via graph convolutional networks. *Knowledge-Based Systems*, *198*, 105893. https://doi.org/10.1016/j.knosys.2020.105893

Zhang, C., Li, W., Wei, D., Liu, Y., & Li, Z. (2023). Network Dynamic GCN Influence Maximization Algorithm With Leader Fake Labeling Mechanism. *IEEE Transactions on Computational Social Systems*, *10*(6), 3361–3369. https://doi.org/10.1109/tcss.2022.3193583

Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks: a comprehensive review. *Computational Social Networks*, *6*(1), 11. https://doi.org/10.1186/s40649-019-0069-y

Zhao, G., Jia, P., Zhou, A., & Zhang, B. (2020). InfGCN: Identifying influential nodes in complex networks with graph convolutional networks. *Neurocomputing*, *414*, 18–26. https://doi.org/10.1016/j.neucom.2020.07.028

Zohdi, M., Rafiee, M., Kayvanfar, V., & Salamiraad, A. (2022). Demand forecasting based machine learning algorithms on customer information: an applied approach. *International Journal of Information Technology*, *14*(4), 1937–1947. https://doi.org/10.1007/s41870-022-00875-3