Enhanced Variational Graph Convolutional Networks with Multi-Scale Convolutions and Attention Mechanisms for Dynamic Network Analysis

Aabid Ahmad Mir
Malaysian Institute of Information Technology, Universiti Kuala Lumpur, Malaysia
miraabid@gmail.com

Megat F. Zuhairi
Malaysian Institute of Information Technology, Universiti Kuala Lumpur, Malaysia
megatfarez@unikl.edu.my (corresponding author)

Shahrulniza Musa
Malaysian Institute of Information Technology, Universiti Kuala Lumpur, Malaysia
shahrulniza@unikl.edu.my

Fuhid Alanazi
Faculty of Computer and Information Systems, Islamic University of Madinah, Saudi Arabia
alanazi@iu.edu.sa

Abdallah Namoun
AI Centre, Faculty of Computer and Information Systems, Islamic University of Madinah, Saudi Arabia
namoun@iu.edu.sa

Ahmed Alrehaili
Faculty of Computer and Information Systems, Islamic University of Madinah, Saudi Arabia
rehaili@iu.edu.sa

Received: 30 October 2024 | Revised: 03 December 2024 | Accepted: 8 December 2024

 DOI: https://doi.org/10.48084/etasr.9443

ABSTRACT

The dynamic and constantly evolving landscape of cyber threats demands innovative methods capable of adapting to the complex relationships and structures inherent in network data. Traditional methods often struggle to adequately capture the intricacies of dynamic networks, especially in terms of evolving temporal dynamics and multiscale dependencies. The proposed solution, Enhanced V-GCN, combines the structural insights of Graph Convolutional Networks (GCNs) with the temporal modeling capabilities of Variational Autoencoders (VAEs), further augmented by multiscale convolutions and attention mechanisms. Multiscale convolutions enable the model to aggregate information across broader neighborhood ranges, while attention mechanisms prioritize the most critical nodes and edges, dynamically adapting to changes within the network. This enhanced approach allows V-GCN to effectively capture both nodal and structural patterns, significantly improving performance in node classification tasks. The Enhanced V-GCN model has demonstrated superior performance in node classification, outperforming baseline models with an accuracy of 98.00%, precision of 97.93%, recall of 98%, and an F1-score of 97.92%, indicating robust classification capabilities and exceptional generalization across diverse network structures.

Keywords-anomaly detection; dynamic networks; deep learning; graph neural networks; node classification

I. INTRODUCTION

A remarkable surge has been observed in the utilization of graph-based methods across various domains, due to significant advancements in information and communication technologies. Graph data, found in applications such as social networks, communication networks, the Internet of Things (IoTs), transportation networks, biological networks, and human disease networks, harbors substantial information and manifests diverse characteristics. Such characteristics include node or edge attributes that contain the properties of entities or connections. Moreover, the nature of graph data in such applications is dynamic, constantly expanding, and increasingly complex. The complexity arises from the multidimensional interactions and the evolving nature of the graphs, necessitating advanced analytical approaches to capture both local and global structural properties efficiently.

At the same time, computing systems have undergone rapid evolution and have transitioned to large-scale, collaborative, and distributed frameworks. This paradigm shift has been facilitated by the introduction of various computing principles such as cloud computing, IoT, edge computing, and federated learning [1]. Effective exploration of extensive graph data in future-generation computing systems requires careful consideration of crucial aspects, including the efficacy of graph learning [2], scalability of large-scale computations, and preservation of privacy within federated computing settings involving multisource graphs. Additionally, the management of graph dynamics in distributed environments poses significant challenges. Researchers have recognized the demand for innovative graph learning theories, specialized platforms tailored to accommodate large-scale graphs, and advanced techniques to process graph-based data. Consequently, several noteworthy research topics have emerged, encompassing knowledge graph reasoning [1], graph self-supervised learning [3], temporal graph modeling [4], and graph embedding techniques [5, 6]. Moreover, applications such as graph-based anomaly detection [7, 8], community detection [9], social recommendation systems, and social influence analytics have received significant attention within the research community.

This study presents enhancements to the Variational Graph Convolutional Network (V-GCN) [10] model by integrating multiscale convolutions and attention mechanisms. The enhancements aim to improve the model's ability to adapt to and accurately analyze dynamic and complex network structures, offering superior performance in node classification tasks and a deeper understanding of graph dynamics.

A. Background and Motivation

Graph representation learning and anomaly detection in complex graphs pose significant challenges due to the diverse network types encountered in real-world scenarios [11]. Most of the existing research focuses on simple graphs, disregarding the complexity introduced by heterogeneous graphs with multiple node types [12, 13], spatiotemporal graphs evolving over time [14], and hypergraphs with relations extending beyond pairwise relationships [15]. Research aims to address these challenges by developing novel anomaly detection and prediction methods, specifically tailored for complex graphs [16]. By incorporating attribute and structure information, as well as modeling the temporal characteristics of dynamic networks, this study aims to improve the accuracy and applicability of anomaly detection techniques [17-19].

In the contemporary landscape of cybersecurity, the paradigm has shifted toward the analysis of dynamic network traffic for the early detection and prevention of malicious activities. The ubiquitous and ever-evolving nature of cyber threats demands innovative approaches that can adapt to the dynamic relationships and structures within network data [20]. Traditional methods often fail to capture the complexities of dynamic networks, particularly in the context of evolving cyber threats [21]. This necessitates a paradigm shift towards advanced techniques that can harness the temporal dependencies and evolving patterns inherent in dynamic network traffic. GCNs [22] have demonstrated unparalleled efficiency in modeling complex relationships within graph-structured data. Their ability to capture complex dependencies among interconnected entities makes them particularly suitable for dynamic network traffic analysis. However, the temporal dimension of network data introduces an additional layer of complexity that traditional GCNs may not fully address. The temporal evolution of network structures, the emergence of new connections, and changes in node behaviors over time, as shown in Figure 1, require a more detailed approach.

B. Dynamic Network Traffic Analysis

Dynamic network traffic analysis plays a key role in modern cybersecurity, tasked with learning regular patterns and abnormalities among constantly evolving network interactions. Unlike its static counterpart, dynamic analysis investigates the temporal dimension, recognizing networks as dynamic entities that undergo constant evolution. This field is driven by the understanding that modern cyber threats are dynamic, constantly adapting, and exploiting vulnerabilities in complex ways. Thus, understanding the dynamics of network traffic is crucial to timely threat detection and the implementation of effective cybersecurity protocols. Dynamic network traffic analysis involves understanding complex relationships, dependencies, and behaviors that evolve within a network over time. This includes monitoring the establishment and termination of connections, identifying emerging patterns, and detecting anomalies indicative of potential security breaches [11]. Traditional methods often face challenges to keep up with the dynamic characteristics of networks, demanding novel approaches capable of adapting to changing structures and behaviors [16].

Graph-based representations offer a robust framework for analyzing dynamic networks. Let $G = (V, E)$ represent a graph, with V the node set denoting network entities and E the edge set indicating relationships between them. GCNs have emerged as a highly effective approach within this framework, leveraging the graph structure to learn feature representations of nodes, facilitating the modeling of complex relationships in a network. The temporal dynamics of a dynamic network, illustrated through a series of snapshots in the DynKDD dataset [23], are shown in the dynamic network plot in Figure 1. Each frame represents a distinct snapshot in time, employing a 3D adjacency tensor Adj_t to capture the temporal dynamics.

$$H^{(l+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\,\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \quad (1)$$

where $H^{(l)}$ denotes the node feature matrix in layer l, $\hat{A} = A + I$ represents the adjacency matrix of the graph with added self-connections, D signifies the degree matrix of $\hat{A}$, and $W^{(l)}$ denotes the weight matrix in layer l, with σ being the activation function. These plots illustrate the progression of connections or interactions over time within a dynamic network, providing insights into the evolving structural dynamics of the graph over time.

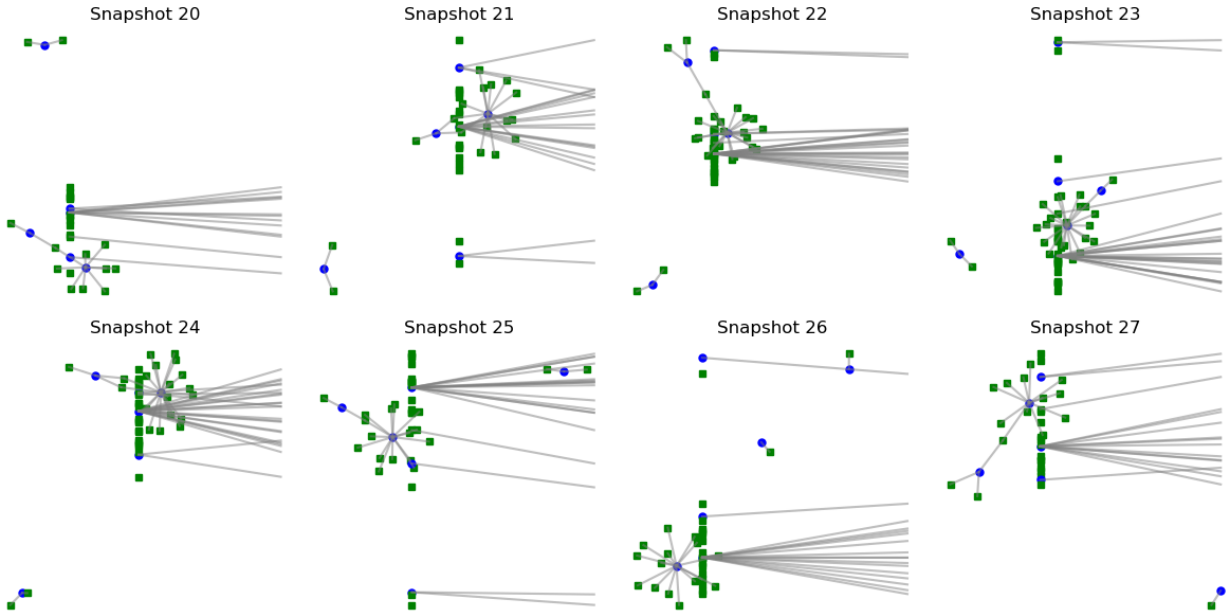

Fig. 1. Dynamic relationships among nodes and edges at different time intervals.

However, the temporal dimension adds an extra level of complexity. Effective dynamic network traffic analysis demands methods that not only can capture the inherent structural dependencies within the network but also can learn the temporal evolution of these structures [4], as illustrated in:

$$H_t^{(l+1)} = \sigma(\hat{D}_t^{-\frac{1}{2}}\hat{A}_t\hat{D}_t^{-\frac{1}{2}}H_t^{(l)}W_t^{(l)}) \quad (2)$$

In such scenarios, advanced models such as VAEs play a crucial role. VAEs introduce a probabilistic aspect, enabling the representation of uncertainty in the temporal transformations of network graphs. The dynamic network traffic analysis process can be formalized using VAEs, where the latent variables z are regarded as random variables with a prior distribution $p(z)$. The encoding process is encapsulated by the variational distribution $q_\phi(z|x)$, which estimates the true and complex posterior distribution $p(z|x)$. The objective of a VAE is to reduce the Kullback-Leibler (KL) divergence between $q_\phi(z|x)$ and $p(z)$ stated as:

$$KL(q_\phi(z|x)||p(z)) = \frac{1}{2}\sum_{i=1}^{K}(\sigma_i^2 + \mu_i^2 - log(\sigma_i^2) - 1) \quad (3)$$

where K denotes the dimensionality of the latent space, μ_i and σ_i represent the mean and standard deviation of the i^{th} element of the variational distribution $q_\phi(z|x)$, and the summation includes all dimensions of the latent space [24]. Integration of GCNs and VAEs in the proposed V-GCN model marks a notable development in dynamic network traffic analysis. The V-GCN aims to seamlessly incorporate structural information from GCNs with the probabilistic temporal modeling capabilities of VAEs. This integration aims to enhance the accuracy of node classification, a task that is crucial in identifying potential threats within dynamic networks. Dynamic network traffic analysis presents a comprehensive challenge that requires a thorough understanding of evolving network structures. By leveraging graph-based models, particularly through the integration of GCNs and VAEs, new possibilities for comprehensive and effective analysis are revealed, significantly improving the reliability of cybersecurity frameworks against evolving cyber threats.

C. Existing Techniques for Dynamic Traffic Analysis

Dynamic network traffic analysis has witnessed substantial development, driven by the critical requirement to understand and respond to evolving cyber threats. This section presents an extensive review of existing techniques, ranging from conventional to state-of-the-art methods.

1) Traditional Approaches

Conventional approaches to dynamic network analysis often rely on heuristic methods [25, 26], rule-based systems [27, 28], and statistical measures [29]. Time-series analysis [30, 31], including techniques such as moving averages and anomaly detection based on predefined thresholds, serves as a fundamental tool for detecting deviations from expected patterns [32, 33]. Although these methods demonstrate effectiveness in certain contexts, their limitations become

apparent when confronting rapidly evolving and sophisticated cyber threats.

2) Graph-based Approaches

In recent years, there has been a notable shift toward graph-based methods, harnessing the intrinsic structure of network data. Graph-based models, such as Graph Neural Networks (GNNs) [34], have proven effective in capturing complex relationships within dynamic networks. GNNs, including GCNs [22], have been employed in node classification, demonstrating the potential to adapt to evolving network structures.

a) Graph Neural Networks (GNNs)

GNNs have been utilized for node classification [35, 36], learning the embedding of each node based on its surrounding neighborhood. GNNs are specialized neural network models designed to work with graph-structured data. A graph is constructed based on predefined relationships within the data, with nodes and edges assigned attributes or features represented as real-valued vectors. Such feature vectors serve as input for GNNs, which perform a feature aggregation step. During this process, each node collects information from its neighboring nodes, guided by the graph structure. GNNs learn patterns of similarity or dissimilarity among node features by leveraging this feature aggregation, enabling them to address various prediction tasks. With advancements in the field and the increasing availability of computational resources and datasets, GNNs have been applied to a wide range of problems, including node classification, community detection, and other graph-based learning applications.

b) Graph Convolutional Networks (GCNs)

GCNs extend the concept of neural networks to graph-structured data [22]. GCNs introduce the fundamental operation of graph convolution, distinct from traditional convolutions in the Euclidean space. In the context of graphs, each node is characterized by a feature vector, and the graph convolution operation is designed to gather features from a node's local neighborhood. The propagation rule across layers in a GCN is shown in (1). The multiplication with $\widehat{D}^{-\frac{1}{2}}\ \widehat{A}\widehat{D}^{-\frac{1}{2}}$ facilitates the computation of a weighted average of neighbor features, thereby integrating both the graph structure and node features into the learning process.

However, the spectral formulation of the graph convolution operation in GCNs relies on the eigen-decomposition of the graph Laplacian, which may pose computational challenges for large graphs. To mitigate this, GCNs employ approximations of spectral convolutions using a localized first-order approximation approach. GCNs have exhibited success across a range of tasks, including node classification, showing their adaptability and efficiency in capturing dependencies within graph-structured data. The adaptation of GCNs to address dynamic and larger graphs continues to be a promising research field, with recent progress emphasizing scalability and temporal dynamics.

c) Variational Graph Autoencoders (VGAEs)

VGAEs emerge as a powerful tool for learning representations of nodes in a graph that evolves over time [37]. VGAEs leverage the inherent structure and dynamics of the graph to capture the complex relationships between nodes and their evolving structure. VGAEs extend traditional autoencoder architectures to graphs by learning low-dimensional representations (node embeddings) that encode both the structural and dynamic properties of the graph. The encoder function $q(Z|X,A)$ maps each node X_i and the adjacency matrix A to a distribution over latent variables z_i capturing the latent structure of the graph. This distribution is assumed to follow a Gaussian distribution, $N(\mu_i, \sigma_i^2)$ where μ_i and σ_i^2 represent the mean and variance of the latent representation for node X_i . Additionally, the decoder function $p(A|Z)$ reconstructs the adjacency matrix A based on the learned node embeddings Z , aiming to capture the underlying graph topology. VGAEs are trained by maximizing the Evidence Lower Bound Objective (ELBO), which balances the reconstruction loss and the KL-divergence between the learned latent distribution and a prior distribution $p(Z)$, effectively guiding the model to learn meaningful and generalizable node representations that adapt to the dynamic nature of the graph. Overall, VGAEs offer an approach to capture temporal dynamics in dynamic graphs, enabling effective node classification tasks in evolving network structures.

D. Limitations of Current Models

Modern graph-based learning approaches, especially those utilized for analyzing dynamic networks, demonstrate significant limitations in capturing temporal changes. Although static graph models, such as GCNs and their variants, have proven effective in processing graph-structured data, their fundamental architecture lacks the temporal aspect required for studying networks that undergo temporal evolution [11, 21].

1) Static Nature of GCNs

Unlike dynamic graphs, where the relationships between nodes change over time, conventional GCNs operate on fixed graph structures and are unable to capture temporal variations in network topology. This static nature of GCNs restricts their applicability in scenarios where the underlying graph undergoes dynamic changes. The lack of a temporal dimension implies that evolutions occurring over time are not inherently taken into account.

2) Challenges in Temporal Variability

The primary obstacles to modeling temporal variability involve addressing several fundamental challenges. First, capturing temporal dependencies is essential, as it outlines how the current state of a network at any given time relies on its preceding states [38]. Second, accommodating variable time scales is crucial, given that dynamic networks often exhibit diverse dynamics operating on distinct time scales, e.g., from frequent changes in communication networks to more gradual shifts in social networks [39]. Therefore, models must have the flexibility to adapt to these varying scales for accurate analysis. Third, non-stationarity presents another significant challenge, as many dynamic networks demonstrate behaviors where their statistical properties evolve over time. This phenomenon poses a notable challenge for models built on the assumption of stationary distributions [40]. Although existing models efficiently handle learning from static graph structures, their

efficacy in dynamic scenarios, where temporal variability plays a pivotal role, remains limited. The future progression of dynamic network analysis relies on developing models that can efficiently capture, analyze, and predict the temporal evolution of networks. Although advances in temporal graph networks and recurrent neural architectures offer promise, further research is essential to fully address these complex challenges.

E. Contributions

This study introduces a novel enhancement to V-GCN by incorporating multiscale convolutions and attention mechanisms to address the challenges associated with dynamic graph analysis. Integration of multiscale graph convolutions enables the model to capture both local and global structural dependencies in dynamic networks. Attention mechanisms dynamically prioritize significant nodes and edges, thus ensuring adaptive and context-aware learning. The integration of GCNs with VAEs with multiscale convolutions and attention mechanisms introduces an enhanced probabilistic aspect, allowing the model to effectively represent uncertainty in evolving network graphs. This dual ability significantly improves performance in node classification tasks, particularly in dynamic and complex environments. The proposed enhanced V-GCN outperforms baseline models (e.g., GCN, GraphSAGE, GAT) across key metrics, including accuracy, precision, recall, and F1-score. The robust performance of the model highlights its ability to generalize effectively across diverse network structures. This approach addresses the limitations of static graph models, such as their inability to capture temporal variations and evolving patterns in network data. The model is specifically tailored for cybersecurity applications, offering robust anomaly detection in rapidly changing network environments. This study incorporates efficient training strategies and benchmarks on datasets such as DynKDD, demonstrating scalability and adaptability to real-world scenarios. Beyond cybersecurity, the proposed model has potential applications in social network analysis, biological networks, and transportation systems, paving the way for future interdisciplinary advances in dynamic graph representation learning.

II. METHODOLOGY

The proposed enhanced V-GCN introduces advanced structural adaptations to traditional GCN methods by incorporating multiscale convolutions and attention mechanisms, as shown in Algorithm 1. Such innovations are designed to enhance the model's capabilities in handling dynamic and complex graph data, particularly in cybersecurity applications where rapid adaptation to evolving network conditions is crucial.

Algorithm 1 Enhanced Variational Graph Convolutional Network (V-GCN)

Input: Graph $G = (V, E)$, Node features $X \in R^{n \times d}$
Output: Node representation Z
procedure $MultiScaleConv(X, A)$
  for $k = 1$ to K do
    $A_k \leftarrow A^k$
    $D_k \leftarrow diag(\sum_j (A_k)_{ij})$
    $H^{(k)} \leftarrow \sigma(D_k^{-\frac{1}{2}} A_k D_k^{-\frac{1}{2}} H W_k)$
  end for
  $H \leftarrow Concatenate(H^{(1)}, \ldots, H^{(k)})$
  return H
end procedure
procedure $AttentionLayer(H, A)$
  $Z \leftarrow 0$
  for $i \in V$ do
    for $j \in N(i)$ do
      $\alpha_{ij} \leftarrow \frac{\exp(LeakyReLU(a^T[Wh_i || Wh_j]))}{\sum_{k \in N(i)} \exp(LeakyReLU(a^T[Wh_i || Wh_k]))}$
      $z_i \leftarrow z_i + \alpha_{ij} W h_j$
    end for
  end for
  return $\sigma(Z)$
end procedure
procedure $VAEEncoder(H, A)$
  $\mu, log\sigma^2 \leftarrow Linear(H), Linear(H)$
  Sample $Z \sim N(\mu, exp(log\sigma^2))$
  return Z
end procedure
procedure $VGCN(G, X)$
  Calculate $\hat{A} \leftarrow A + I$
  $H \leftarrow MultiScaleConv(X, \hat{A})$
  $H' \leftarrow AttentionLayer(H, \hat{A})$
  $Z \leftarrow VAEEncoder(H')$
  Minimize:
  $L = ReconLoss(X, Z) + \beta KL(N(\mu, \sigma^2) \parallel N(0, I))$
end procedure

A. Multiscale Graph Convolutions

Multiscale graph convolutions enable the model to capture information from a broad range of node neighborhoods, enhancing its ability to perceive both local and global graph structures. This is accomplished through the following steps.

1) Extended Neighborhood Aggregation

For each node, features from various hop distances are aggregated. This allows the model to capture a more comprehensive view of the graph topology as represented in:

$$H^{(l+1)} = \sigma\left(\sum_{k=0}^{K} D_k^{-1/2} A_k D_k^{-1/2} H^{(l)} W_k^{(l)}\right) \quad (4)$$

where A_k is the adjacency matrix considering k-hop neighbors, D_k is the degree matrix corresponding to A_k, $H^{(l)}$ is the node feature matrix at layer l, $W_k^{(l)}$ is the weight matrix for the k^{th} neighborhood at layer l, and σ is the nonlinear activation function.

2) Layer-Wise Feature Integration

Features aggregated from different neighborhood scales are integrated to form a comprehensive feature representation for each node. This integration allows the network to learn which

scales of neighborhoods are most informative for specific tasks, such as node classification.

B. Attention Mechanisms

To prioritize the most informative parts of the graph dynamically, an attention mechanism is integrated into the graph convolution process.

1) Node-Level Attention

Each node calculates attention coefficients with its neighbors to prioritize which nodes should influence its next state more significantly. The attention coefficients are calculated using:

$$\alpha_{ij} = \frac{exp(LeakyReLU(a^T[Wh_i||Wh_j]))}{\sum_{k\in N(i)} exp(LeakyReLU(a^T[Wh_i||Wh_k]))} \quad (5)$$

where a is the learnable parameter vector of the attention mechanism, W is the shared weight matrix, h_i is the feature vector of node i, $||$ denotes concatenation, and $N(i)$ is the set of neighbors of node i.

2) Feature Transformation with Attention

The node features are then transformed considering the calculated attention coefficients, enabling selective focus on more important nodes during feature aggregation as shown in:

$$h'_i = \sigma\left(\sum_{j\in N(i)} \alpha_{ij} W h_j\right) \quad (6)$$

where h'_i is the updated feature vector of node i.

C. Integration of GCNs and VAEs

The structural information captured by the enhanced GCN layers is complemented by the probabilistic modeling capabilities of VAEs, allowing for a robust feature representation that incorporates both deterministic and stochastic aspects.

1) Variational Autoencoder Framework

The VAE transforms the aggregated node features into a latent space where each node's features are represented as distributions, capturing the underlying uncertainty in the graph dynamics. The reparameterization trick is used to sample from such distributions, providing a way to perform backpropagation through stochastic nodes.

2) KL Divergence Minimization

The model minimizes the Kullback-Leibler divergence between the approximate posterior and the prior, enhancing the model's ability to generalize, as shown in:

$$KL\ Loss = \sum_{i=1}^{N} KL(q(z_i|X)||p(z_i)) \quad (7)$$

where $q(z_i|X)$ is the variational posterior, $p(z_i)$ is the prior distribution, and N is the number of nodes in the graph.

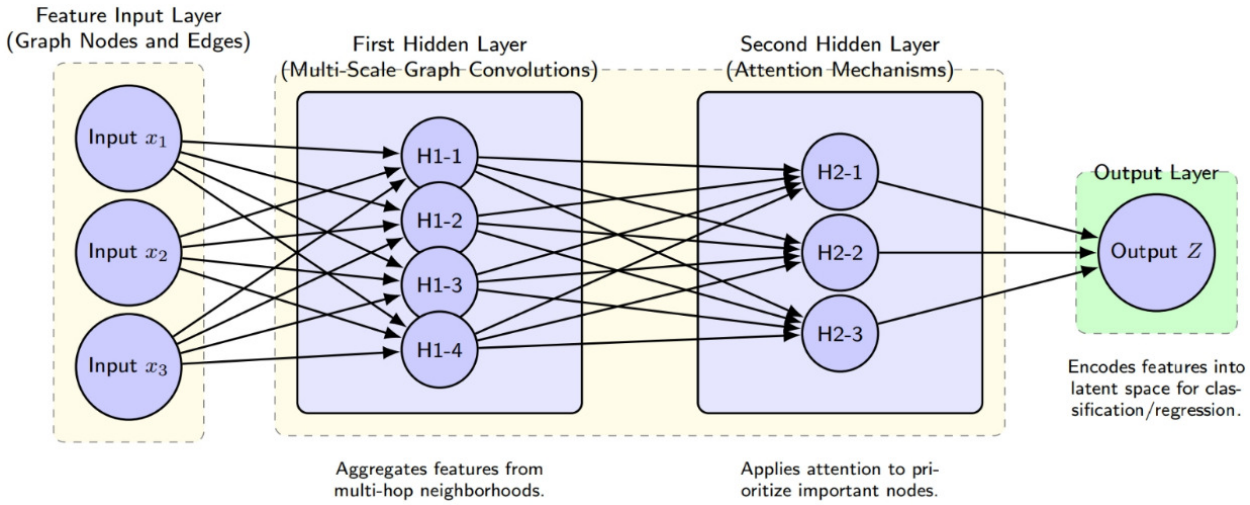

Fig. 2. Deep learning workflow of an enhanced V-GCN with multiscale convolutions, attention, and VAE integration.

D. Deep Learning Workflow

Figure 2 illustrates a Fully Connected Neural Network (FCNN) that incorporates features of a V-GCN, specifically designed for tasks that involve graph-based data. It begins with the feature input layer, which represents the input features derived from graph nodes and edges, such as $Input\ x_1$, $Input\ x_2$, and $Input\ x_3$. These features enter the network, where they are sequentially transformed through layers that perform multiscale graph convolutions, attention mechanisms, and VAE integration. The first hidden layer is responsible for multiscale graph convolutions. This layer captures information from different levels of neighborhood relationships within the graph by aggregating features from various hops or distances around each node. This enables the model to consider both local and global contexts, enriching the feature representations with multi-hop neighborhood information. Each hidden node in this layer (H1-1, H1-2, H1-3, H1-4) processes the multiscale features. The second hidden layer applies attention mechanisms to dynamically weigh the significance of each node's neighbors. The attention mechanism calculates an attention coefficient for each neighbor connection, allowing the model to prioritize nodes based on their relevance. Selective focus helps the model capture critical relationships in the graph while

reducing the influence of less relevant nodes. The nodes in this layer (H2-1, H2-2, H2-3) reflect the attention-based feature transformation. The output layer performs the integration of GCNs and VAEs, creating a latent representation that combines graph-based features with the probabilistic characteristics of a VAE. This layer, represented by output node Z, encapsulates the final encoding, incorporating uncertainty and probabilistic information from the VAE to handle complex, noisy, or uncertain data. The latent representation can then be used for downstream tasks, such as classification or regression.

III. EXPERIMENTAL EVALUATION

The system was running on Ubuntu 22.04 LTS, which was chosen for its seamless integration of machine learning libraries and tools, facilitating experiments and model development. Jupyter Notebook was used, running on Python 3.11.5, for interactive computing, combining code execution with rich text and visualizations. This setup ensured compatibility with cutting-edge libraries and facilitated the implementation of the proposed system. Experiments were carried out using essential Python libraries, such as torch, torch_geometric, numpy, pandas, scikit-learn, and matplotlib. The primary dataset, DynKDD, is a dynamic adaptation of the NSL-KDD dataset, addressing its static nature by introducing temporal dynamics. Its division into training and test sets allows for comprehensive learning and evaluation. Each DynKDD record is timestamped, converting it into a sequence of time-ordered snapshots, capturing network interactions over time. PyTorch geometric data objects were instantiated from the data and the V-GCN model was initialized for training. Performance evaluation metrics ensured effectiveness and generalizability in real-world settings.

A. Training and Validation

The training and validation of the V-GCN exhibited a successful learning phase, represented by a consistent decrease in loss and a substantial increase in accuracy, as shown in Figure 3. The training and validation process progressed over 100 epochs, with the learning trajectory recorded at each step. Initially, the model experienced a steep loss, indicating rapid learning, which tapered as the model began to converge to a stable state. The loss dropped from 3.1077 in the first epoch to 0.0617 by the final, showing significant learning and model improvement. As the model trains, there is a notable and consistent increase in accuracy on both the training and testing datasets. The initial accuracy was relatively high at 0.6543 for training and 0.6604 for testing, indicating that the model was able to learn meaningful representations from the data right from the beginning. As the epochs progressed, the training accuracy showed a steady climb and ultimately plateaued at 0.9842, indicating that the model maximized its learning from the training data. The validation accuracy corresponded to this trend closely, starting at 0.6604 and reaching 0.9819, a sign of excellent generalization. Throughout the training process, the V-GCN model demonstrated a strong ability to generalize from the training to the validation data, which is demonstrated by the narrow gap between training and testing accuracy (see Table I). This close alignment is indicative of a well-tuned model that manages to avoid overfitting while still capturing the complexities of the data. The increasing trend in validation accuracy up to the final epoch suggests that the model's predictions are reliable and can be trusted when applied to new, unseen data.

TABLE I. EPOCH-WISE V-GCN MODEL TRAINING AND VALIDATION PERFORMANCE

Epoch	Loss	Train Acc.	Test Acc.
1	3.1077	0.6543	0.6604
10	0.4783	0.9007	0.9015
20	0.2594	0.9310	0.9300
30	0.1960	0.9435	0.9460
40	0.1565	0.9575	0.9583
50	0.1293	0.9619	0.9628
60	0.1112	0.9704	0.9701
70	0.0955	0.9730	0.9718
80	0.0820	0.9762	0.9768
90	0.0704	0.9811	0.9799
100	0.0617	0.9842	0.9819

B. Performance Evaluation

Accuracy, precision, recall, and F1 score were employed to evaluate the performance of the V-GCN model in node classification tasks, as they offer comprehensive insight into the model's ability to make correct predictions, capture true positive and false negative instances, and strike a balance between precision and recall. The training loss graph exhibited an initial sharp decrease followed by a tapering trend as the epochs progressed, indicating rapid error reduction initially, followed by slower fine-tuning as convergence approached. The training loss decreased from an initial value exceeding 3.06 to approximately 0.059 over 100 epochs. Conversely, the training accuracy displayed a consistent increase, leveling off as it approached higher values, signifying improved correctness of model predictions with training. Starting around 76.95%, the training accuracy steadily increased to more than 98% after 100 epochs. These trends suggest a progressive enhancement in the model's predictive accuracy on the training data, aligning with expectations in successful model training scenarios. The testing metrics revealed high performance, with an accuracy of 98.01%, a precision of 97.93%, a recall of 98.01%, and an F1 score of 97.94%. These results demonstrate the model's robust generalization capability, assuming that the test dataset adequately represents the true distribution. The model's performance on node classification can be considered highly satisfactory. Against common benchmarks in various fields, accuracy and F1 scores above 97% are generally indicative of a high-quality model. The lack of discrepancy between training and testing performance suggests that the model is not overfitting. The model achieved very high accuracy relatively quickly, which indicates that the features were highly informative and the model architecture is particularly well-suited to this task.

C. Comparative Analysis

The comparative analysis of V-GCN with baseline models, such as GAT [41], GraphSAGE [42], and GCN [22], for node classification results exhibits a distinct performance difference as shown in Figure 4. The performance was evaluated over 100 epochs and the key performance indicators were loss, training accuracy, final accuracy, precision, recall, and F1 score, as presented in Table II.

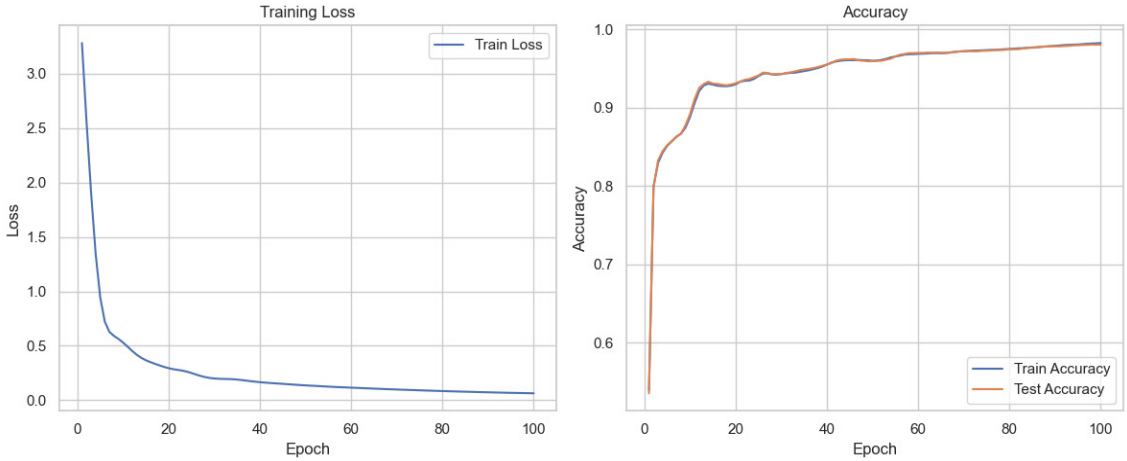

Fig. 3. V-GCN training and validation curves.

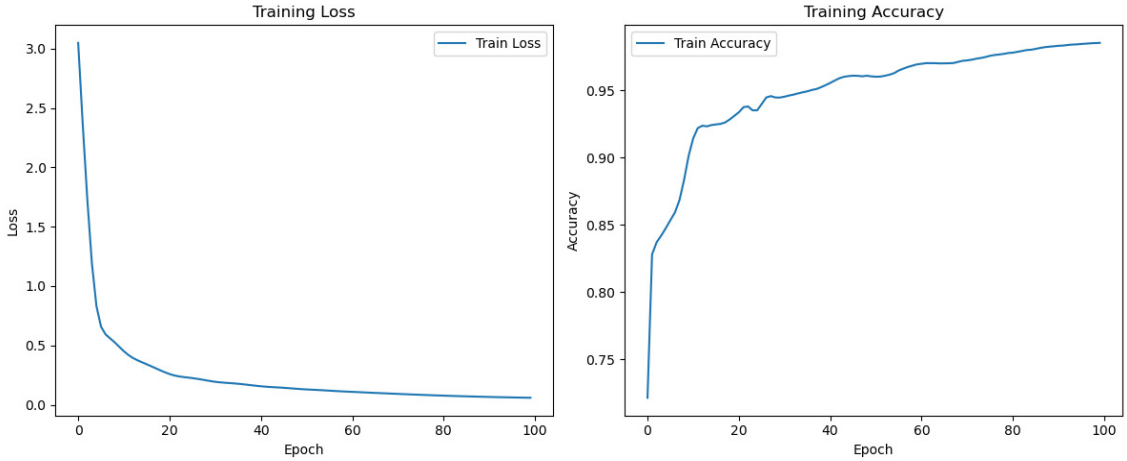

Fig. 4. Loss reduction and accuracy.

TABLE II. COMPARISON OF NODE CLASSIFICATION RESULTS FOR V-GCN AGAINST BASELINE MODELS

Model	Training accuracy	Accuracy	Precision	Recall	F1 score
GCN [22]	0.7243	0.7306	0.8167	0.7306	0.7274
GraphSAGE [42]	0.7885	0.7900	0.7831	0.79000	0.7822
GAT [41]	0.8916	0.8617	0.8628	0.8617	0.8605
V-GCN	0.9840	0.9800	0.9793	0.9800	0.9792

The GCN model achieved a final training accuracy of 72.43% and a test accuracy of 73.06%, indicating a reasonably good fit with minimal overfitting, as evidenced by the narrow margin between training and testing accuracy. However, the loss at the end of training was relatively higher compared to other models, suggesting less efficient learning over epochs. Although precision was notably high at 81.67%, recall was equal to accuracy, implying correct predictions only 73.06% of the time for positive class instances. The F1 score, which balances precision and recall, was the lowest among the models at 72.74%, suggesting potential gaps in the capture of relevant graph data patterns. In comparison, GraphSAGE exhibited a slightly higher accuracy of 79.00% than GCN, with training accuracy closely mirroring the final accuracy, indicative of good generalization without overfitting. Despite its lower precision of 78.31%, GraphSAGE compensated with higher recall, resulting in a comparable F1 score of 78.22%, indicating its effectiveness in identifying relevant positive class instances. GAT demonstrated a substantial improvement over GCN and GraphSAGE, achieving an accuracy of 86.17% on the test set, highlighting the efficacy of the attention mechanism in learning graph representations. Its balanced precision and recall, both around 86%, led to an F1 score of 86.05%, indicating effective identification of relevant instances with minimal incorrect predictions. V-GCN outperformed all models, with the lowest loss and the highest training and test accuracies (98.40% and 98.00%, respectively), reflecting highly effective learning. Its precision and recall, both nearly 98%, underscored its precision and robustness in classifying correct labels, while the highest F1 score of 97.92% highlighted its balanced performance. Despite the high training accuracy, similar test accuracy suggested an exceptional generalization without overfitting.

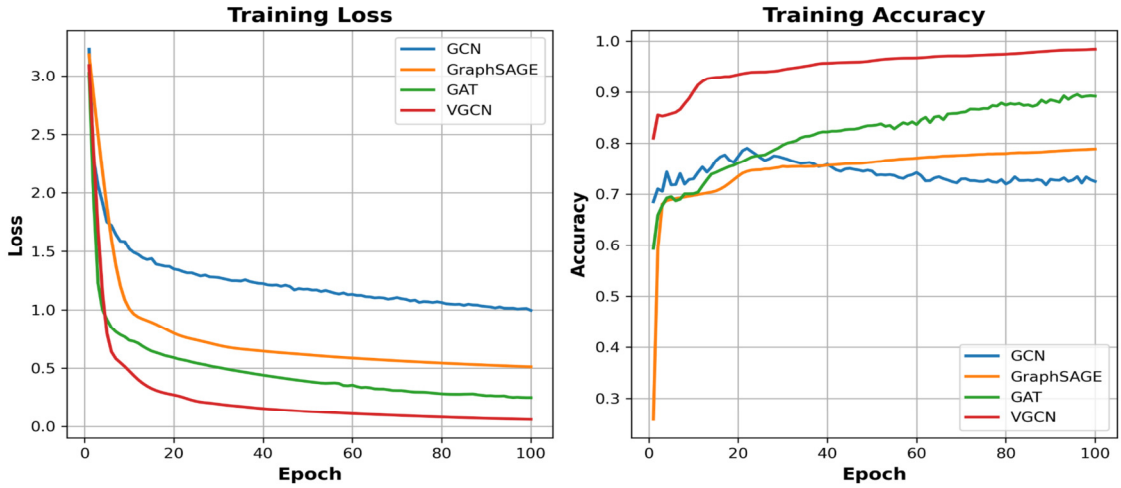

Fig. 5. Comparative analysis of GAT and V-GCN for node classification.

IV. LIMITATIONS AND FUTURE WORK

Although the V-GCN model represents a notable advance in dynamic graph representation learning, it is subject to limitations regarding temporal granularity, dependency on data quality, and adaptability that must be addressed. Regarding temporal granularity, the model's effectiveness depends on its ability to capture and process temporal dynamics in graph data. However, its operation at a certain temporal resolution may lead to missing critical transient dynamics, necessitating methods for handling higher temporal granularity to capture rapid changes. The dependency on data quality is another challenge, as the model performance is heavily influenced by the quality of input data, emphasizing the importance of robust preprocessing and data cleaning techniques to mitigate issues arising from noisy or biased data. Additionally, the model's adaptability to new and evolving patterns in dynamic networks is crucial for its long-term applicability, requiring mechanisms such as online learning and transfer learning to facilitate quick adaptation without extensive retraining. Despite these limitations, the V-GCN model holds promise for cybersecurity applications and beyond, with potential applications in diverse fields such as social network analysis, biological network interpretation, and transportation networks, provided that it is customized to specific application requirements.

V. CONCLUSION

The V-GCN model represents a significant advance in the field of graph-based learning, particularly for dynamic networks that are characterized by evolving relationships and attributes over time. This study demonstrates that V-GCN not only addresses the inherent limitations observed in traditional GCNs by incorporating temporal dynamics through the integration of VAEs but also enhances the model's ability to handle complex, evolving data structures effectively. Throughout the experimental evaluation, V-GCN consistently exhibited superior performance in node classification compared to other prevalent models such as GAT, GraphSAGE, and traditional GCN. The model's capacity to maintain high accuracy, precision, recall, and F1 score underlines its robustness and effectiveness in handling node classification tasks within dynamically changing graphs. The introduction of a variational approach within the GCN framework allows for a probabilistic representation of node features, thereby introducing an element of uncertainty handling that is crucial for real-time and dynamic network environments. The training and validation phases highlighted the model's efficiency in learning from dynamic graph data, demonstrating a steady decrease in loss and an increase in accuracy over epochs. This indicates not only the model's ability to learn significant representations from the data but also its ability to generalize well on unseen data, a critical factor for practical applications. The V-GCN model stands out as a robust framework for dynamic network analysis, offering promising directions for future research and potential applications across various domains such as cybersecurity, social network analysis, and beyond. The model's ability to integrate structural and temporal information makes it a valuable tool for researchers and practitioners looking to gather deeper insights into the evolving patterns of complex networks.

REFERENCES

[1] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A Survey on Knowledge Graphs: Representation, Acquisition, and Applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514, Oct. 2022, https://doi.org/10.1109/TNNLS.2021.3070843.

[2] F. Xia *et al.*, "Graph Learning: A Survey," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 109–127, Apr. 2021, https://doi.org/10.1109/TAI.2021.3076021.

[3] Y. Liu *et al.*, "Graph Self-Supervised Learning: A Survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 5879–5900, Jun. 2023, https://doi.org/10.1109/TKDE.2022.3172903.

[4] P. Jiao *et al.*, "Temporal Network Embedding for Link Prediction via VAE Joint Attention Mechanism," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 7400–7413, Sep. 2022, https://doi.org/10.1109/TNNLS.2021.3084957.

[5] S. Pan, R. Hu, S. F. Fung, G. Long, J. Jiang, and C. Zhang, "Learning Graph Embedding With Adversarial Training Methods," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2475–2487, Jun. 2020, https://doi.org/10.1109/TCYB.2019.2932096.

[6] M. Wu, S. Pan, L. Du, and X. Zhu, "Learning Graph Neural Networks with Positive and Unlabeled Nodes," *ACM Transactions on Knowledge*

Discovery from Data, vol. 15, no. 6, pp. 1–25, Jun. 2021, https://doi.org/10.1145/3450316.

[7] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou, and G. Karypis, "Anomaly Detection on Attributed Networks via Contrastive Self-Supervised Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 6, pp. 2378–2392, Jun. 2022, https://doi.org/10.1109/TNNLS.2021.3068344.

[8] Y. Liu *et al.*, "Anomaly Detection in Dynamic Graphs via Transformer," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12081–12094, Sep. 2023, https://doi.org/10.1109/TKDE.2021.3124061.

[9] D. Jin *et al.*, "A Survey of Community Detection Approaches: From Statistical Modeling to Deep Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 1149–1170, Oct. 2023, https://doi.org/10.1109/TKDE.2021.3104155.

[10] A. A. Mir, M. F. Zuhairi, S. Musa, M. H. Alanazi, and A. Namoun, "Variational Graph Convolutional Networks for Dynamic Graph Representation Learning," *IEEE Access*, vol. 12, pp. 161697–161717, 2024, https://doi.org/10.1109/ACCESS.2024.3483839.

[11] J. Ren, F. Xia, I. Lee, A. Noori Hoshyar, and C. Aggarwal, "Graph Learning for Anomaly Analytics: Algorithms, Applications, and Challenges," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 2, pp. 1–29, Apr. 2023, https://doi.org/10.1145/3570906.

[12] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous Graph Neural Network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, Jul. 2019, pp. 793–803, https://doi.org/10.1145/3292500.3330961.

[13] J. Zhao, X. Wang, C. Shi, B. Hu, G. Song, and Y. Ye, "Heterogeneous Graph Structure Learning for Graph Neural Networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, pp. 4697–4705, May 2021, https://doi.org/10.1609/aaai.v35i5.16600.

[14] H. Huang, R. Shi, W. Zhou, X. Wang, H. Jin, and X. Fu, "Temporal Heterogeneous Information Network Embedding," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, Montreal, Canada, Aug. 2021, pp. 1470–1476, https://doi.org/10.24963/ijcai.2021/203.

[15] J. Wang, Y. Zhang, Y. Wei, Y. Hu, X. Piao, and B. Yin, "Metro Passenger Flow Prediction via Dynamic Hypergraph Convolution Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7891–7903, Sep. 2021, https://doi.org/10.1109/TITS.2021.3072743.

[16] Q. Ding and J. Li, "AnoGLA: An efficient scheme to improve network anomaly detection," *Journal of Information Security and Applications*, vol. 66, May 2022, Art. no. 103149, https://doi.org/10.1016/j.jisa.2022.103149.

[17] B. Chen *et al.*, "GCCAD: Graph Contrastive Coding for Anomaly Detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 8037–8051, Dec. 2023, https://doi.org/10.1109/TKDE.2022.3200459.

[18] H. Chen, X. Mei, Z. Ma, X. Wu, and Y. Wei, "Spatial–temporal graph attention network for video anomaly detection," *Image and Vision Computing*, vol. 131, Mar. 2023, Art. no. 104629, https://doi.org/10.1016/j.imavis.2023.104629.

[19] X. Wang *et al.*, "Heterogeneous Graph Attention Network," in *The World Wide Web Conference*, San Francisco, CA, USA, May 2019, pp. 2022–2032, https://doi.org/10.1145/3308558.3313562.

[20] P. Malhotra, Y. Singh, P. Anand, D. K. Bangotra, P. K. Singh, and W.-C. Hong, "Internet of Things: Evolution, Concerns and Security Challenges," *Sensors*, vol. 21, no. 5, Jan. 2021, Art. no. 1809, https://doi.org/10.3390/s21051809.

[21] A. A. Mir, M. F. Zuhairi, and S. M. Musa, "Graph Anomaly Detection with Graph Convolutional Networks.," *International Journal of Advanced Computer Science & Applications*, vol. 14, no. 11, 2023.

[22] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks." arXiv, Feb. 22, 2017, https://doi.org/10.48550/arXiv.1609.02907.

[23] A. A. Mir, M. F. Zuhairi, S. Musa, and A. Namoun, "Adaptive Anomaly Detection in Dynamic Graph Networks," in *2024 International Visualization, Informatics and Technology Conference (IVIT)*, Kuala Lumpur, Malaysia, Aug. 2024, pp. 200–206, https://doi.org/10.1109/IVIT62102.2024.10692372.

[24] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes." arXiv, Dec. 10, 2022, https://doi.org/10.48550/arXiv.1312.6114.

[25] V. Cerqueira, L. Torgo, M. Oliveira, and B. Pfahringer, "Dynamic and Heterogeneous Ensembles for Time Series Forecasting," in *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Tokyo, Japan, Oct. 2017, pp. 242–251, https://doi.org/10.1109/DSAA.2017.26.

[26] I. Sánchez, "Adaptive combination of forecasts with application to wind energy," *International Journal of Forecasting*, vol. 24, no. 4, pp. 679–693, Oct. 2008, https://doi.org/10.1016/j.ijforecast.2008.08.008.

[27] F. Collopy and J. S. Armstrong, "Rule-Based Forecasting: Development and Validation of an Expert Systems Approach to Combining Time Series Extrapolations," *Management Science*, vol. 38, no. 10, pp. 1394–1414, Oct. 1992, https://doi.org/10.1287/mnsc.38.10.1394.

[28] J. S. Armstrong, M. Adya, and F. Collopy, "Rule-Based Forecasting: Using Judgment in Time-Series Extrapolation," in *Principles of Forecasting: A Handbook for Researchers and Practitioners*, J. S. Armstrong, Ed. Boston, MA, USA: Springer US, 2001, pp. 259–282.

[29] J. Durbin and S. J. Koopman, *Time Series Analysis by State Space Methods*. Oxford University Press, 2012.

[30] J. D. Hamilton, *Time Series Analysis*. Princeton University Press, 2020.

[31] Y. Fu, D. Wu, and B. Boulet, "Reinforcement Learning Based Dynamic Model Combination for Time Series Forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 6, pp. 6639–6647, Jun. 2022, https://doi.org/10.1609/aaai.v36i6.20618.

[32] A. Alrehaili, A. Mir, and M. Junaid, "A retrospect of prominent cloud security algorithms," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 3, pp. 749–755, 2020.

[33] A. Mir, M. F. Zuhairi, S. Musa, T. A. Syed, and A. Alrehaili, "POSTER: A Survey of Security Challenges with 5G-IoT," in *2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, Riyadh, Saudi Arabia, Nov. 2020, pp. 249–250, https://doi.org/10.1109/SMART-TECH49988.2020.00063.

[34] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, Jan. 2009, https://doi.org/10.1109/TNN.2008.2005605.

[35] S. K. Maurya, X. Liu, and T. Murata, "Simplifying approach to node classification in Graph Neural Networks," *Journal of Computational Science*, vol. 62, Jul. 2022, Art. no. 101695, https://doi.org/10.1016/j.jocs.2022.101695.

[36] M. H. Alanazi, "G-GANS for Adaptive Learning in Dynamic Network Slices," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 14327–14341, Jun. 2024, https://doi.org/10.48084/etasr.7046.

[37] T. N. Kipf and M. Welling, "Variational Graph Auto-Encoders." arXiv, Nov. 21, 2016, https://doi.org/10.48550/arXiv.1611.07308.

[38] S. Tian, S. Zhang, H. Mao, R. Liu, and X. Xiong, "Temporal Network Link Prediction Based on the Optimized Exponential Smoothing Model and Node Interaction Entropy," *Symmetry*, vol. 15, no. 6, p. 1182, Jun. 2023, https://doi.org/10.3390/sym15061182.

[39] D. Ghosh *et al.*, "The synchronized dynamics of time-varying networks," *Physics Reports*, vol. 949, pp. 1–63, Feb. 2022, https://doi.org/10.1016/j.physrep.2021.10.006.

[40] O. Ryan, J. Haslbeck, and L. Waldorp, "Non-Stationarity in Time-Series Analysis: Modeling Stochastic and Deterministic Trends." OSF, Jul. 13, 2023, https://doi.org/10.31234/osf.io/z7ja2.

[41] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks." arXiv, Feb. 04, 2018, https://doi.org/10.48550/arXiv.1710.10903.

[42] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *Advances in Neural Information Processing Systems*, 2017, vol. 30.