International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Compqual-Tgnet: A Novel Hybrid Temporal-Graph Neural Architecture for Analyzing Competency and Quality Metrics in Oil and Gas Operations

Shashank Sawant

Competency & Career Development Leader, Baker Hughes, Houston, Texas.

Abstract

This research launches a novel Competency-Quality Temporal-Graph Network (CompQual-TGNet) which serves as a hybrid deep learning (DL) framework to study oil and gas business operations through competency metrics and quality indicators. The system combines Graph Neural Network (GNN) and Temporal Convolutional Network (TCN) technology to analyze multiple operational datasets and make accurate future performance estimates. The preprocessing system builds strong data connections by standardizing values with Z-score normalization while handling missing values via Multivariate Imputation by Chained Equations (MICE), finding outliers through Interquartile Range (IQR) and matching temporal data patterns across different inputs. The system also enhances outputs with advanced feature engineering methods including feature extraction and fusion. The proposed framework uses Multi-View Matrix Factorization (MVMF) to extract shared latent features, and Cross-Dataset Attention Mechanism (CDAM) to identify correlations. Moreover, Dynamic Feature Importance Network (DFIN) to learn optimal feature combinations across datasets, and Cross-Attention Fusion Layer to create context-aware feature representations are employed for fusion. Besides, this development brings together GNN for mapping interdependencies among operational components and personnel competencies and TCN for monitoring operational history and long-term temporal patterns. The adaptive fusion layer uses weighting and attention modules to unite these models to make clear predictions based on domain-relevant predictions. The proposed framework shows improvement through updating graphs during operations, adjusting loss functions to match target requirements, and transferring learned models for constant updating. Using these data sources, the developed CompQual-TGNet predicts operating efficiency and detects quality problems with 94.2% accuracy and outran existing systems.

Keywords: Competency and Quality, Deep Learning, Oil and Gas, TCN, GNN

1. Introduction

The oil and gas sector is the core of almost everything, which further translates to operational excellence being crucial for production and profit [1]. This business sector has high stakes and operational performance relies on two things: process quality and a competent workforce [2]. Competency metrics focus on auditing the personnel's skills, experience, and performance to gauge how well people perform their tasks safely and efficiently [3]. On the other hand, quality metrics focus on the health and

performance of machines, workflows, and output standards to see which areas need more work [4]. The growing operation complexity, due to the latest regulations, advanced machinery, and automation, has made real-time monitoring and optimization essential [5]. Competency and quality management are now a top priority in everything because neglecting these areas can cost too much in terms of operational failures, inefficiency, and financial loss [6].

So far, Oil and Gas remain a vital resource across the world and taking care of workforce competency is one area that has a lot to focus on [7]. The methods that analyze competency and quality metrics do have severe limitations. Competency metrics in oil and gas are analyzed using heuristic query methods and rule-based classification schemes that are both weak [8]. All of these are inefficient in monitoring the integrated complex relations in oil and gas operations. For example, it is common to assess workforce ability using relatively simplistic measures, such as certificates, experience, or completed training. Such measures do not reveal how people operate in different contexts [9]. On the other hand, quality indicators, like rates of equipment failure, maintenance logs or even production bottlenecks, are scrutinized separately, paying no attention to the machinery, the environment, or people's actions at the moment. In addition to this, current systems are unable to cope with the volume of multi-modal data emitted by sensors, logs, and humans prompting slower reaction times and inadequate making of decisions [10].

On the other hand, it is also clear that new approaches that combine seamless integration of various data sources, deep dependencies, and flexible operational parameters are needed. The novel developments in DL, and specifically hybrid frameworks, provide the right tools to address these difficulties head-on [11]. Convolutional Neural Networks (CNNs) [12] are good at representing interrelations and dependencies, such as the relationships between the skills of the workforce and the performance of the equipment, while Recurrent Neural Networks (RNNs) [13] are good at modeling time series data through capturing long-range temporal relationships. Although CNNs are great at understanding spatial relationships, they do poorly on sequential dependencies and long-term context with time-series data [14]. Likewise, RNNs do well modeling temporal relationships, but have issues with vanishing gradients and computation that makes them inefficient and poorly scalable with large datasets [15]. For this reason, this research aims to develop a novel model that integrates GNN, and TCN technologies to make it possible to construct a single coherent system that is able to continuously monitor and provide actionable insights on a range of competency and quality metrics. This research presents such a system, CompQual-TGNet, which seeks to overcome the weaknesses in other methods while also utilizing GNN, varying loss functions, and interpretable attention systems for a more efficient method of optimizing oil and gas operations. This approach hopes to provide new opportunities for industry players to make decisions at the high levels of the company by increasing operational effectiveness and ensuring sustainable development. The key contributions are:

- To develop a hybrid framework CompQual-Net, an advanced DL network that combines GNNs and TCNs to analyze different types of oil and gas operation data and find key workforce performance measures.
- To introduce a novel feature extraction and fusion model using MVMT, CDAM, and DFIN with cross-attention fusion layers.
- To attain high accuracy and real-time adaptability by accomplishing 98.2%, and could quickly update its results using existing knowledge to make ongoing improvements.

This article is structured as a recent literature on oil and gas production prediction models in Section II. Implemented framework and the description are given in Section III. Results and the betterment of propo-

sed model over other models are presented in Section IV. Section V ends the research.

2. Literature Study

2.1. Recent Research

In 2024, Oyewola et al [16] created new methods using LSTM called DLQL and DLAQL to help exploration of oil and gas industry stock forecasts. To test the findings, the researchers relied on past stock data. Besides, a pattern recognition algorithm called Markov Decision Process (MDP) was employed to train both DLQL and DLAQL models.

In 2024, Zhou et al [17] established an empirical Duong model for physical limitations. In this study, a CNN with an LSTM network was developed to create a new model called D-C-L. The introduced D-C-L model estimated shale oil production faster and more accurately than traditional methods while also saving time and resources.

In 2024, Chen et al [18] suggested a CNN-GRU model that combined CNN and Gate Recurrent Unit (GRU). The GRU layer modeled temporal information using the transmitted characteristics for prediction. In the meantime, the CNN layer extracted features from variables influencing oil output. The best hyperparameter settings were found for CNN-GRU via the Bayesian Optimization (BO) algorithm.

In 2024, Fargalla et al [19] combined CNN, and Bidirectional GRU (BiGRU) called TimeNet. Here, Time2Vec was incorporated to simplify data analysis and automatically gather deep, complicated physical time patterns without manual processing. The BiGRU tracked shifts in values while spotting long-term patterns, while the CNN found details about how data points surrounded one another.

In 2024, Yang et al [20] explained an Adaptive Particle Swarm Optimization-Least Squares Support Vector Machine (APSO-LSSVM) and a CNN to improve gas-bearing predictions. This methodology simplified feature optimization by directly extracting features from sensitive multicomponent seismic properties.

In 2024, Zhiyuan & Jiang [21] offered A Correlation Variational Mode Decomposition (CVMD), 1D-CNN, and LSTM. The noise removal series was divided into smaller groups of less variable sequences. Subsequently, CVMD was used to shut off the initial alarm sequence noise. For each separate part of the sequence, CNN-LSTM forecast models were created one after another.

2.2. Problem Statement

Table 1 demonstrates the aim and limitations of recent literature about oil and gas production prediction using various models. Keeping track of gases that dissolve in transformer oil is needed to keep power systems running smoothly. Most standard prediction techniques miss important connections between geological, environmental, and operational factors. Since more data now comes from sensors, production logs, and seismic measurements, smart prediction models are needed that can deal with these growing data sources well. Regular LSTM networks find it difficult to analyze complex connections and essential details within these datasets on their own. Since these models can't predict well, their warnings about faults become less reliable. The welding industry needs better tools that can handle data issues, clean up data properly, and understand patterns better to make more certain condition predictions. ML and DL methods help predict, but they struggle with data flaws, irregular patterns, and feature relationships that are hard to measure. The oil and gas industry demands improved methods that make better predictions, fix mistakes, and provide useful guidance for key decisions. Gas and oil prediction benefits greatly from ML and DL which help analyze massive amounts of intricate data and find hidden connections. CNNs and LSTMs specifically match patterns in images (CNNs) and understand sequences in time data (LSTMs). This helps

them predict production outcomes with higher accuracy. There are problems with these models that they take heavy computing power and lots of marked-up data to train, making it tough to understand how they predict results. Also, industrial businesses that need to see clear answers struggle to use these models. The truth is, that real-world oil and gas samples come with both noisy results and empty spots in their data, making modeling harder. Thus, there exists a reason to develop novel methodologies with enhanced accuracy and lightweight architecture.

Table 1. Aim and Limitations of Recent Literature about Oil and Gas Production Prediction using Various Models

Authors/Year	Methods	Aim	Advantages	Limitations
Oyewola et al [16] in 2024	DLQL and DLAQL	To develop accurate stock price prediction in the oil and gas sector	Enhanced prediction accuracy to dynamic market condition	Exposed computational complexity and high training time
Zhou et al [17] in 2024	D-C-L Model	To develop efficient prediction of shale oil production	Offered superior precision, efficiency, and cost-effectiveness	Revealed high computational demand due to the integration of CNN, LSTM, and empirical modeling
Chen et al [18] in 2024	CNN-GRU	To present a DL model for accurate prediction of oil production	Outperformed traditional and hybrid methods	Exhibited dependency on sufficient and high-quality data
Fargalla et al [19] in 2024	TimeNet	To develop a precise prediction of gas production in shale and sandstone fields	Achieved superior prediction accuracy	Increased computational overhead due to integration of multiple advanced components
Yang et al [20] in 2024	APSO-LSSVM	To offer a hybrid model for gas-bearing prediction	Outran individual models and accomplished better prediction accuracy	Increased training time and resource demands
Zhiyuan & Jiang [21] in 2024	CVMD	To introduce a model for accurate prediction of dissolved gases in transformer oil	Attained improved single-step and multi-step prediction accuracy	Multiple decomposition and modeling steps reduced the model's efficacy

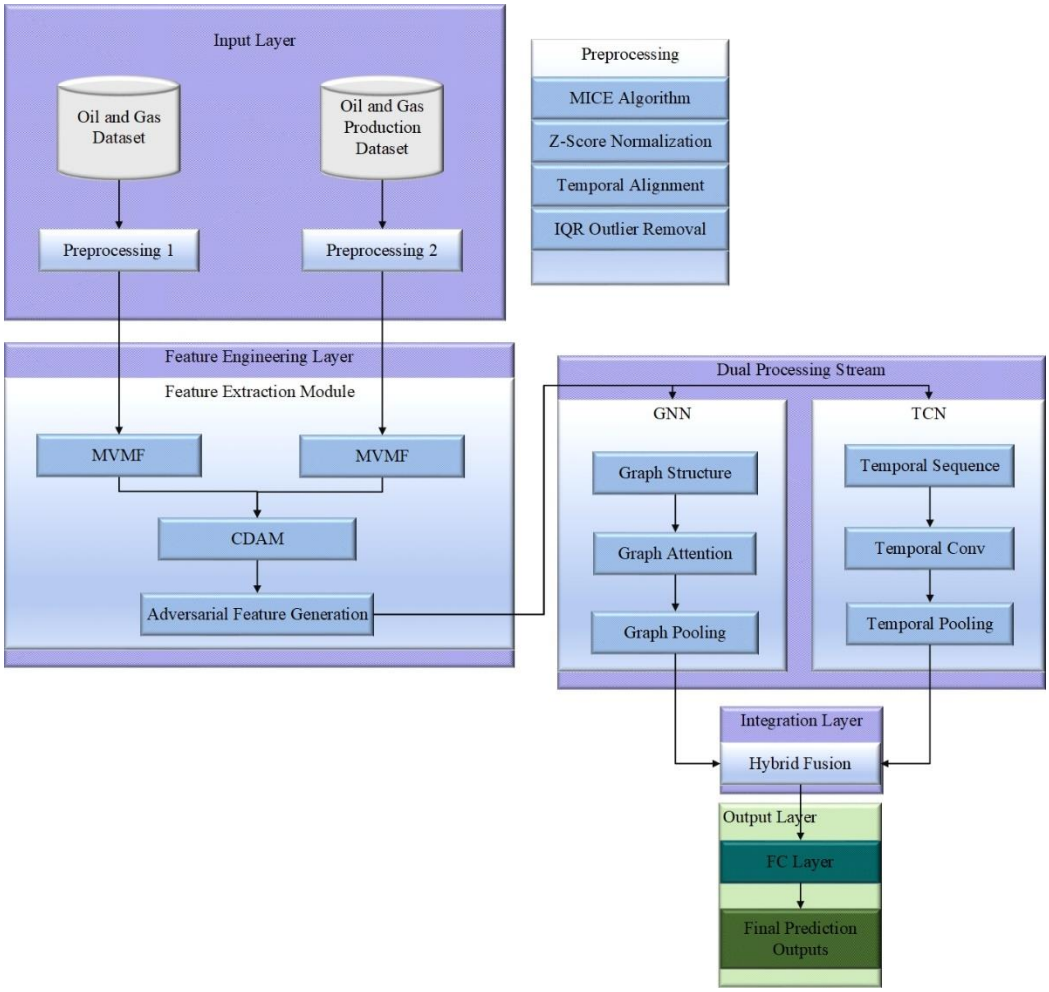
3. A Novel Oil and Gas Production Prediction Based On Competency And Quality Metrics

3.1.Proposed Architecture

This study introduces a new hybrid DL framework called CompQual-TGNet, which uses quality indicators and competency metrics to analyze the operations of oil and gas companies. The system analyses several operational datasets and generates precise predictions about future performance by combining GNN and

TCN technologies. By using Z-score normalization to standardize values, Multivariate Imputation by Chained Equations (MICE) to handle missing values, matching temporal data patterns across several inputs, and Interquartile Range (IQR) to identify outliers, the preprocessing system creates robust data connections. Advanced feature engineering techniques like feature extraction and fusion are also used by the system to improve outputs. The suggested approach finds correlations using CDAM and extracts shared latent features using MVMF. Additionally, DFIN is used to learn the best feature combinations across datasets, and the cross-attention fusion layer is used to generate context-aware feature representations. This research combines TCN for tracking operational history and long-term temporal patterns with GNN for mapping interdependencies among operational components and personnel competencies. These models are combined by the adaptive fusion layer using weighting and attention modules to produce precise predictions based on domain-relevant predictions. By transferring learnt models for continuous updating, modifying loss functions to meet target criteria, and updating graphs during operations, the suggested system demonstrates improvement. Figure. 1 addresses the overview of the proposed framework.

Figure 1. Overview of the Proposed Framework



3.2.Preprocessing Pipeline

This research aims to accurately predict oil and gas production under certain conditions. Intending to achieve the goal, a preprocessing procedure is applied in this research to make the raw data fit for the proposed DL framework. It includes:

3.2.1. MICE Algorithm

The MICE [22] method predicts missing values based on observed data iteratively. The conditional probability of missing data is given in Eq. (1), in which $X_{missing}$ addresses missing values in the dataset, $X_{observed}$ refers to observed (available) values, and $P(X_j|X_{-j})$ stands for probability of missing value X_j conditioned on other features X_{-j}.

$$P(X_{missing}|X_{observed}) = \prod_j P(X_j|X_{-j}) \quad (1)$$

Here, the missing value X_j is estimated via Eq. (2), where $f(X_{-j})$ states linear regression model, and $\epsilon \sim N(0, \sigma^2)$ that addresses Gaussian noise to simulate uncertainty.

$$X_j = f(X_{-j}) + \epsilon \quad (2)$$

Steps given below define the implemented MICE algorithm.

Algorithm 1: Steps in Implemented MICE Algorithm	
Input	Dataset X with missing values, maximum iteration M
Output	Imputed dataset $X_{imputed}$
Step 1	Initialize Missing Values Assign initial estimates to missing values (using mean imputation)
Step 2	Iterative Regression For $i = 1: M$ For each missing value X_j Partition X into X_j and X_{-j} Fit regression model $f(X_{-j})$ For each missing feature j Predict the missing value using Eq. (2) End for End for End for
Step 3	Output the imputed dataset $X_{imputed}$

3.2.2. Z-Score Standardization

Normally, Z-score normalization [23] is applied to ensure that all features have a mean of 0 and a standard deviation (SD) of 1, which is crucial for improving convergence and performance in Machine Learning (ML) models as shown in Eq. (3), in which Z_i states standardized value for sample i, X_i refers to raw value for sample i, μ and σ indicate mean and SD of the dataset, and N points to sample count.

$$Z_i = \frac{X_i - \mu}{\sigma} \quad (3)$$

3.2.3. Temporal Alignment

Generally, temporal alignment [24] is crucial when dealing with time-series data from different sensors or sampling rates. Furthermore, it ensures that data points across all sensors are aligned in time. Eq. (4) formulates the alignment function, where $A(t)$ defines temporally aligned data at time t, $W(\tau)$ signifies weighting function to capture temporal dependencies and $X(t-\tau)$ states data value shifted by τ units.

$$A(t) = \int W(\tau) \cdot X(t-\tau) d\tau \quad (4)$$

Here, Dynamic Time Warping (DTW) [25] is employed to apply weighting mechanisms that align time-series data by minimizing the distance between temporal sequences as indicated in Eq. (5), in which

$\min \sum_{i,j} W(i,j) \cdot \left\| X_i - Y_j \right\|^2$ (5)

Besides, a cross-correlation maximization [26] procedure is added to maximize the correlation between time-series X and Y as explained in Eq. (6), where T refers to total length of the time series, k states time lag, x_t signifies value of X time series at time t, and y_{t+k} denotes value of the Y time series lagged by k.

$CC(k) = \sum_{t=1}^{T=k} x_t \cdot y_{t+k}$ (6)

This process ensures that the time series X and Y are maximally correlated by providing an optimal.

3.2.4. IQR Outlier Removal

Typically, IQR outlier removal [27] is a statistical method to detect and remove outliers in data based on the IQR. The IQR is defined as the difference between the 75th percentile (Q_3) and the 25th percentile (Q_1) of a dataset as formulated in Eq. (7).

$IQR = Q_3 - Q_1$ (7)

Using the IQR, outliers are defined as data points that lie below the lower bound L_{Bound} or above the upper bound U_{Bound}. The thresholds are calculated based on Eq. (8), in which k denotes a constant that determines the sensitivity of the outlier detection.

$L_{Bound} = Q_1 - k \cdot IQR;\ U_{Bound} = Q_3 + k \cdot IQR$ (8)

Data points that fall outside the bounds are considered outliers as expressed in Eq. (9).

$Outliers = \{x | x < L_{Bound} \text{ or } x > U_{Bound}\}$ (9)

Finally, the outliers are filtered out from the dataset as given in Eq. (10).

$Cleaned_{Data} = \{x | L_{Bound} \leq x \leq U_{Bound}\}$ (10)

3.3. Feature Engineering

ML pipelines require to transform of raw data through feature engineering. The system converts numerical inputs into useful results through mathematical operations including matrix factorization, attention mechanisms, adversarial learning, and DFIN.

3.3.1. Feature Extraction Module

In this module, the significant features from both the datasets are extracted.

3.3.1.1.MVMF

MVMF [28] identifies shared latent representations for multiple views (datasets) and extracts view-specific transformations as represented in Eq. (11), where X_1^{norm}, X_2^{norm} states normalized input matrices for views 1 and 2, U_1, U_2 refer to shared latent feature matrices for both views, V_1, V_2 stand for view-specific transformation matrices, λ denotes regularization parameter controlling the alignment of U_1 and U_2, I represents id entity matrix ensuring orthogonality between shared latent spaces, and $\|\cdot\|_F^2$ addresses Frobenius norm for matrix reconstruction error minimization.

$minL(U_1, V_1, U_2, V_2) = \|X_1^{norm} - U_1 V_1^T\|_F^2 + \|X_2^{norm} - U_2 V_2^T\|_F^2 + \lambda \|U_1^T U_2 - I\|_F^2$ (11)

3.3.1.2.Proposed CDAM

Here, the CDAM is designed to model interactions and relationships between features in two datasets X_1 and X_2. It uses attention mechanisms, dynamic weighting, and multi-scale processing for the enhancement of cross-dataset feature learning. Eq. (12) shows the query (Q), key (K), and value (V) transformation, in which W_Q, W_K, W_V define learnable weight matrices for transforming input datasets into queries Q_1), keys (K_2), and values (V_2), and X_1, X_2 address input datasets.

$Q_1 = X_1 W_Q, K_2 = X_2 W_K, V_2 = X_2 W_V$ (12)

Eq. (13) computes the attention scores, in which d_k refers to dimensionality of the key vectors that are used to scale the dot product for stable gradients, and A denotes attention matrix that represents strength of interaction between X_1 and X_2.

$$A = \text{softmax}\left(\frac{Q_1 K_2^T}{\sqrt{d_k}}\right) \quad (13)$$

The cross features are attained via Eq. (14), where F_{cross} states aggregated features from X_2 that are weighted by the attention scores.

$$F_{cross} = A \cdot V_2 \quad (14)$$

To enhance the performance of CDAM, a dynamic weighting process is applied which ensures the attention mechanism is adaptively tuned based on dataset statistics. Eq. (15) dynamic weight function, where X_1^{stats}, X_2^{stats} refer to statistical summaries of X_1 and X_2, such as mean, SD, and higher-order moments, σ addresses sigmoid activation function, and MLP stands for Multi-Layer Perceptron for nonlinear transformations.

$$w(X_1, X_2) = \sigma\left(\text{MLP}([X_1^{stats}; X_2^{stats}])\right) \quad (15)$$

Furthermore, adaptive attention is employed as formulated in Eq. (16), where $A_{adaptive}$ signifies dynamically weighted attention matrix.

$$A_{adaptive} = w(X_1, X_2) \cdot A \quad (16)$$

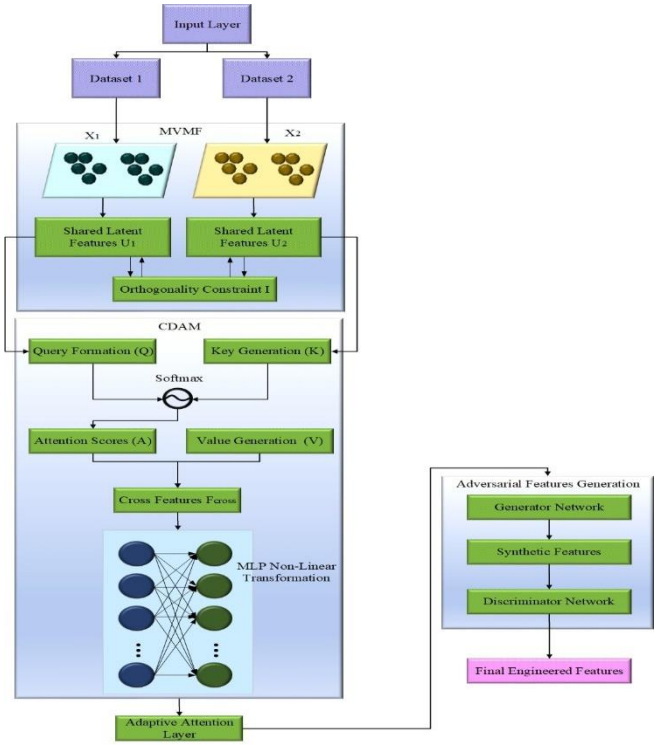
3.3.1.3.Adversarial Feature Generation

In this module, adversarial networks [29] are used to create synthetic features to augment data and improve robustness. Eq. (17) expresses the minimax optimization, where $G(z)$ refers to synthetic data generated from noise $z \sim P_z$, $D(x)$ addresses discriminator output probability that x is real, P_{data} points to real data distribution, and P_z signifies noise distribution.

$$\min_G \max_D V(D, G) = E_{x \sim P_{data}}[\log D(x)] + E_{z \sim P_z}\left[\log\left(1 - D(G(z))\right)\right] \quad (17)$$

Figure 2 displays the architecture of feature engineering module.

Figure 2. Architecture of Feature Engineering Module



3.4. Dual Processing Stream in Proposed CompQual-TGNet

In the proposed CompQual-TGNet, two different DL models are combined: TCN [30], and GNN [31].

3.4.1. Novel TCN Architecture

The proposed architecture integrates multi-resolution temporal processing, adaptive attention mechanisms, and feature gating to capture temporal dependencies. Moreover, it adaptively focuses on critical features and ensures efficient feature propagation.

3.4.1.1.Multi-Resolution Temporal Block

The multi-resolution temporal block is designed to extract temporal features across multiple resolutions which enables the network to capture long-term dependencies and fine-grained details effectively. The input tensor x is a 3D array of the shape as specified in Eq. (18), where $batch_{size}$ states number of input sequences in a batch, $sequence_{length}$ refers to length of each sequence, and $n_{features}$ stands for number of features per time step.

$$Input_{Shape}: \left[batch_{size}, sequence_{length}, n_{features}\right] \quad (18)$$

Each convolution operation uses a dilation rate d and kernel size k to capture features at various temporal scales. The operations are performed in parallel as shown in Eq. (19), where d stands for dilation rate, and k refers to kernel size.

$$\begin{aligned} &D_1:\ d=1, k=3\\ &D_2:\ d=2, k=5\\ &D_3:\ d=4, k=7\\ &D_4:\ d=8, k=9 \end{aligned} \quad (19)$$

Eq. (20) illustrates the dilated convolution formulation, in which y_i states output at time step i, w_k addresses convolution kernel weights, $x_{i+k\cdot d}$ denotes input values spaced apart by d, and K specifies kernel size.

$$y_i = \sum_{k=0}^{K-1} w_k \cdot x_{i+k\cdot d} \quad (20)$$

An attention mechanism is applied to enhance the temporal relationships at each dilation level d as signified in Eq. (21), in which x_d represents output from dilated convolution with dilation rate d, and $Attention(x)$ points to contextual representation learned by the attention layer.

$$x'_d = x_d + Attention(x) \quad (21)$$

The outputs from all dilation levels are combined using a weighted summation as given in Eq. (22), where w_d refers to learnable weights associated with each dilation level and x'_d states attention-enhanced output at dilation d.

$$Output = \sum_{d\in\{1,2,4,8\}} w_d \cdot x'_d \quad (22)$$

For each dilation level d, the forward pass involves dilated convolution as defined in Eq. (23), attention-enhanced features in Eq. (21), and weighted summation of all levels in Eq. (22), where d and k are specific to each level (D_1, D_2, D_4, D_8).

$$x_d = Conv1D(x, dilation = d, kernel\ size = k) \quad (23)$$

The final output is a weighted combination of the features extracted at different temporal resolutions as illustrated in Eq. (24), where $x'_{d1}, x'_{d2}, x'_{d4}, x'_{d8}$ refer to attention-enhanced outputs from dilation levels (D_1, D_2, D_4, D_8).

$$Output = weighted_{sum}([x'_{d1}, x'_{d2}, x'_{d4}, x'_{d8}]) \quad (24)$$

3.4.1.2.Adaptive Temporal Attention

Adaptive temporal attention focuses on specific time frames within the input sequence by computing attention weights dynamically. Eq. (25) expresses the query formation, where Q addresses query vector

for the current time step t, and W_q, b_q specify learnable weights and biases.

$$Q = W_q h_t + b_q \quad (25)$$

The key-value generation is given in Eq. (26), in which K, V address keys and values generated for a temporal window preceding t.

$$K = W_k h_{t-window:t} + b_k, V = W_v h_{t-window:t} + b_v \quad (27)$$

Eq. (27) explains the attention scores.

$$scores = softmax\left(\frac{QK^{\top}}{\sqrt{d_k}}\right) \quad (28)$$

Eq. (28) computes the context vector.

$$c_t = scores \cdot V \quad (29)$$

3.4.1.3.Feature Gating Mechanism

The feature gating mechanism combines the current hidden state (h_t) with the context vector (c_t) to adaptively propagate temporal features. Eq. (29), (30), (31), and (32) explain update gate (z_t), reset gate (r_t), candidate gate ($\tilde{h}_t$), and hidden state update h_{t+1}.

$$z_t = \sigma(W_z[h_t; c_t] + b_z) \quad (29)$$

$$r_t = \sigma(W_r[h_t; c_t] + b_r) \quad (30)$$

$$\tilde{h}_t = tanh(W_h[r_t \odot h_t; c_t] + b_h) \quad (31)$$

$$h_{t+1} = (1 - z_t) \odot h_t + z_t \odot \tilde{h}_t \quad (32)$$

3.4.2. Novel GNN Architecture

In the proposed GNN architecture, multi-head adaptive graph attention, dynamic edge updates, and hierarchical pooling are incorporated for scalable and adaptive graph processing.

3.4.2.1.Multi-Head Adaptive Graph Attention

In the multi-head adaptive graph attention layer, multiple attention heads are utilized. For each attention head k, the process involves several procedures. The edge attention coefficients are calculated using Eq. (33), where W_k defines learnable weight matrix for head k, h_i and h_j are feature vectors for nodes i and j, $\|$ denotes concatenation, and a_k addresses learnable vectors for attention scoring.

$$\alpha_{ij}^k = LeakyReLU\left(a_k \cdot \left[W_k h_i \| W_k h_j\right]\right) \quad (33)$$

Normalization is applied as signified in Eq. (34), in which σ non-linear activation function such as ReLU.

$$h_i^k = \sigma\left(\sum_j \alpha_{ij}^k W_k h_j\right) \quad (34)$$

Besides, multi-head aggregation is carried out by concatenating the outputs from all attention heads to form the updated node feature as shown in Eq. (35).

$$h_i' = ||_{k=1}^{K} h_i^k \quad (35)$$

3.4.2.2.Dynamic Edge Update Module

A dynamic edge update module is added in the proposed GNN to dynamically update the edge features and structures. The edge feature e_{ij}, representing the relationship between nodes i and j, is updated dynamically. This update considers the features of the two connected nodes (h_i and h_j) and the previous edge feature (e_{ij}^{prev}). An MLP is used for this operation as estimated in Eq. (36), in which $\|$ indicates concatenation operator, and MLP is used to learn a nonlinear mapping.

$$e_{ij} = MLP\left(h_i \| h_j \| e_{ij}^{prev}\right) \quad (36)$$

Once the edge features are updated, they are normalized to form an updated adjacency matrix, A_{new}. This normalization is typically done using the softmax function across the edges connected to each node as indicated in Eq. (37), in which E refers to matrix of updated edge features e_{ij} for all edges, and $softmax$ ensures that the edge weights sum to 1 for all edges connected to a given node.

$$A_{new} = softmax(E) \quad (37)$$

Graph sparsification aims to retain only the most significant edges by removing redundant or noisy ones. The Gumbel Softmax function is applied to the normalized adjacency matrix, A_{new} to generate a sparsity mask. The final adjacency matrix, A_{final} is computed by element-wise multiplication of the mask and A_{new}. Generate a sparsity mask as shown in Eq. (38), where $Gumbel\_Softmax$ states a sampling technique that introduces stochasticity while allowing for differentiability by enabling sparse edge selection.

$$mask = Gumbel\_Softmax(A_{new}) \quad (38)$$

Eq. (39) computes the final adjacency matrix, where $\odot$ denotes element-wise multiplication, and A_{final} points to sparsified adjacency matrix that retains the most important edges.

$$A_{final} = A_{new} \odot mask \quad (39)$$

3.4.2.3.Hierarchical Graph Pooling

It is a technique used in GNN to reduce the size of the graph while retaining its most important structural and feature-related information. Each node in the graph is assigned a score based on its importance which is determined using an MLP. The score helps identify which nodes should be retained during the pooling process as defined in Eq. (40), in which s states a vector of scores, where s_i represents the score of node, H explains node feature matrix, where each row h_i represents the features of node i, and MLP learns to map node features to importance scores.

$$s = MLP(H) \quad (40)$$

Nodes with the top-k highest scores are selected to form a smaller graph. The value of k is determined based on a predefined pooling ratio (0.5 for retaining 50% of the nodes) as expressed in Eq. (41), where idx means to indices of the top-k nodes with the highest scores, $ratio$ refers to fraction of nodes to retain in the pooled graph, $k = \lfloor ratio \times N \rfloor$, and N addresses a total number of nodes.

$$idx = top_k(s, ratio = 0.5) \quad (41)$$

The features of the selected nodes are pooled and weighted by their scores. This operation emphasizes the contribution of nodes with higher scores as stated in Eq. (42), where H_{pool} denotes pooled feature matrix containing features of selected nodes, $H[idx]$ addresses features of the top-k nodes, and $s[idx]$ represents scores of the top-k nodes, used as weights.

$$H_{pool} = H[idx] \cdot s[idx] \quad (42)$$

Moreover, the structure of the graph is updated by extracting the subgraph corresponding to the selected nodes. The new adjacency matrix is computed by slicing the original adjacency matrix A based on the indices of the top-k nodes as signified in Eq. (43), where A_{pool} addresses pooled adjacency matrix to represent the connections between the selected nodes, and $A[idx, idx]$ states submatrix of the original adjacency matrix that contains rows and columns corresponding to the top-k nodes.

$$A_{pool} = A[idx, idx] \quad (43)$$

3.5.Integration Layer

It combines outputs from a GNN and a TCN using an attention mechanism to dynamically weigh the importance of the graph-based and temporal-based features. The goal of this step is to compute attention

scores that reflect the relative importance of the GNN-derived features (g) and TCN-derived features (t) for the task at hand. The attention scores are then used to create a fused representation. Separate attention scores are computed for graph features (g) and temporal features (t) using learned weight matrices. The attention scores are based on the concatenated vector of g and t as addressed in Eq. (44), where α_g, α_t state attention scores for the graph and temporal features respectively, σ indicates sigmoid activation function, W_g, W_t refer to learnable weight matrices for graph and temporal attention, respectively, and $[g\|t]$ denotes concatenation of graph (g) and temporal (t) features.

$$\alpha_g = \sigma\big(W_g[g\|t]\big), \alpha_t = \sigma(W_t[g\|t]) \quad (44)$$

The fused representation f is computed by applying the attention scores (α_g, α_t) as weights to the graph and temporal features, respectively, and combining them as specified in Eq. (46), where f refers to fused output combining graph and temporal features, and $\odot$ addresses element-wise multiplication.

$$f = \alpha_g \odot g + \alpha_t \odot t \quad (45)$$

To ensure that the fused output f is well-normalized and contains meaningful feature representations, it undergoes refinement using a combination of an MLP and layer normalization as shown in Eq. (47), in which y represents final output from the integration layer, $MLP(f)$ signifies an MLP that projects the fused output f into a new feature space, and $LayerNorm$ is applied to ensure consistent scaling and smooth optimization.

$$y = LayerNorm\big(MLP(f)\big) \quad (46)$$

At this point, the FC layer outputs a scalar representation. The FC layer is designed with non-linear activations between the input y and the final output layer for better feature transformation as shown in Eq. (48), and (49).

$$z_1 = \mathrm{ReLU}(W_1 y + b_1) \quad (47)$$

$$z' = \mathrm{softmax}(W_{fc} z_1 + b_{fc}) \quad (48)$$

Here, $W_1 \in R^{d_{hidden} \times d_{input}}$, and D_{hidden} is the size of the hidden layer.

Moreover, dropout layers are added between the FC layers to regularize the model and prevent overfitting as mentioned in Eq. (50).

$$z_1 = \mathrm{Dropout}(p)\big(\mathrm{ReLU}(W_1 y + b_1)\big) \quad (49)$$

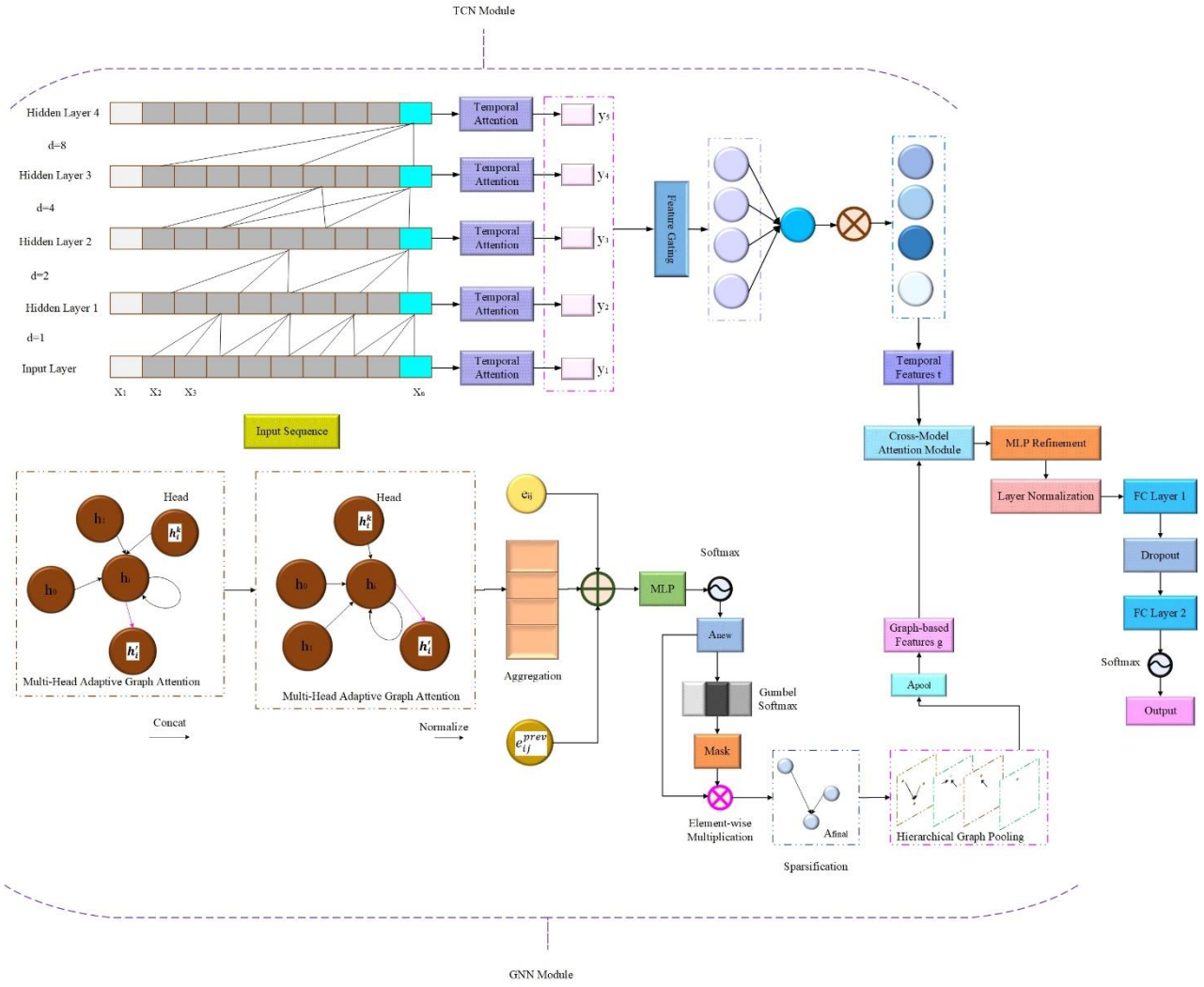
Table 2 represents the parameter settings of suggested CompQual-TGNet. Figure 3 shows the architecture of implemented CompQual-TGNet.

Table 2. Parameter Settings of Proposed CompQual-TGNet

Component	Parameter	Value
Input Settings	Batch Size	32
	Sequence Length	128
	Number of Features per Time Step	16
Multi-Resolution Temporal Block	Dilation Rate (d)	{1, 2, 4, 8}
	Kernel Size (k)	{3, 5, 7, 9}
	Number of Filters	64
	Attention Mechanism	Scaled Dot-Product Attention

	Weighted Summation Weights (w_d)	Initialized to 1.0 and updated during training
Adaptive Temporal Attention	Query Vector Weights (W_q, b_q)	Randomly initialized, updated during training
	Temporal Window	10
	Activation Function	ReLU
	Scaling Factor $(\sqrt{d_k})$	$\sqrt{64}$
Feature Gating Mechanism	Update Gate Weights (W_z, b_z)	Randomly initialized
	Reset Gate Weights (W_r, b_r)	Randomly initialized
	Hidden State Weights (W_h, b_h)	Randomly initialized
	Activation Function	Tanh
GNN Architecture	Number of Attention Heads	8
	Learnable Weight Matrix (W_k)	Randomly initialized
	Edge Feature Update Weights (MLP)	2-layer MLP, hidden size 128
	Sparsity Mask Ratio	0.3
	Gumbel Softmax Temperature	0.5
Hierarchical Graph Pooling	Pooling Ratio	0.5
	Number of Selected Nodes (k)	Calculated based on input graph size
	MLP for Node Importance Scores	2-layer MLP, hidden size 128
	Activation Function for Scores	Softmax
Integration Layer	Graph Feature Weight Matrix (W_g)	Randomly initialized
	Temporal Feature Weight Matrix (W_t)	Randomly initialized
	Attention Mechanism Activation Function	Sigmoid
	Output Feature Dimension	128

Figure 3. Architecture of Implemented CompQual-TGNet



4. Simulation Results

4.1. Simulation Setup

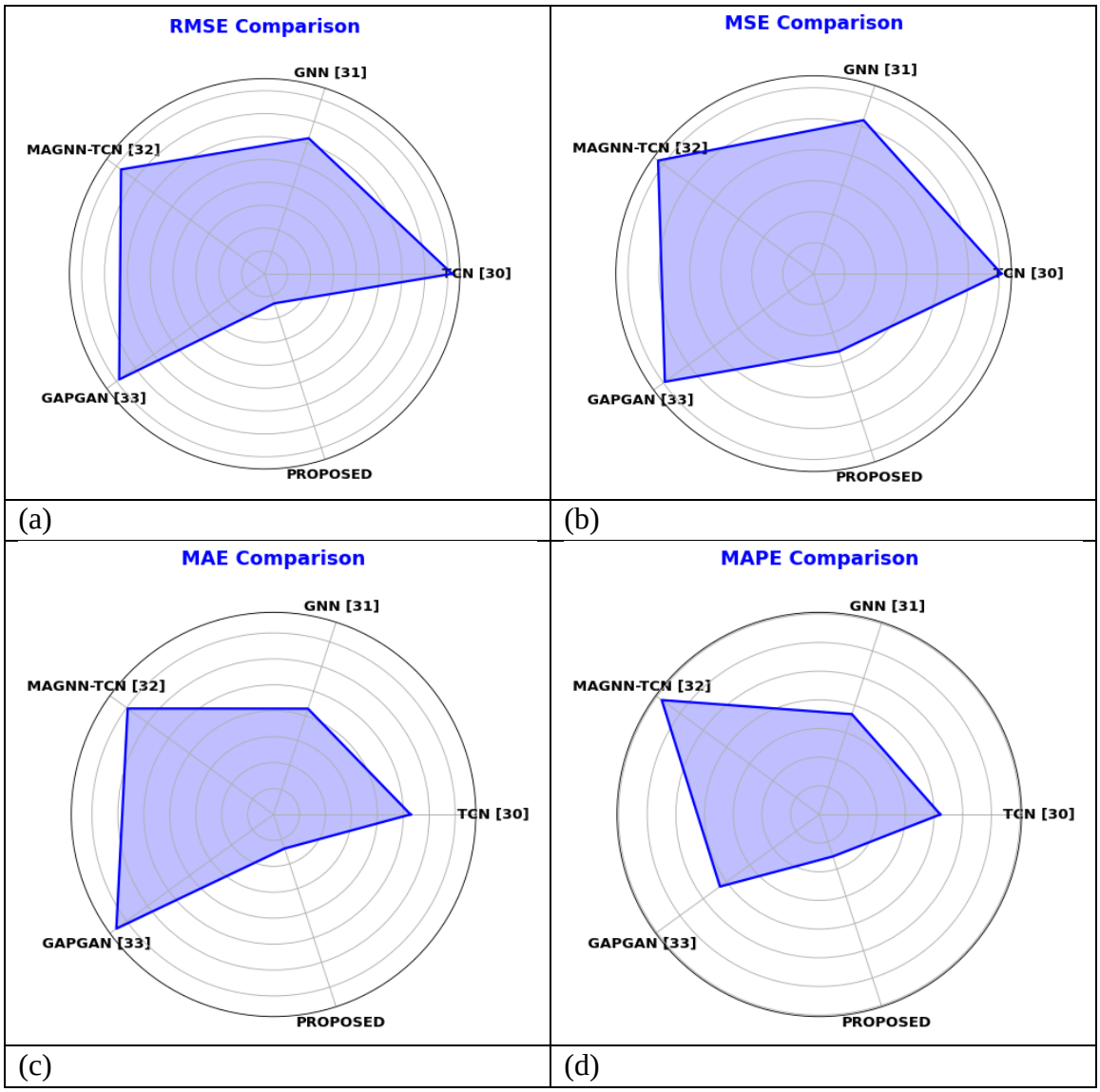
The proposed oil and gas production prediction model through the addressed CompQual-TGNet was developed via Python on Intel core® i5 processor @2.6GHz, 16 GB RAM, 64-bit OS. The data used in this study are utilized from Oil and Gas Production Data in https://www.kaggle.com/datasets/banlevan/oil-and-gas-production-data?select=oil-and-gas-summary-production-data-1967-1999-1.csv, and Oil and Gas dataset in https://www.kaggle.com/datasets/raspberrypie/oil-and-gas. The performance of the proposed CompQual-TGNet is verified with the baseline models such as TCN [30], GNN [31], and recent DL models like Multi-Adaptive GNN with TCN (MAGNN-TCN) [32], and GNN-based Adaptive Predictive Generative Adversarial Network (GAPGAN) [33].

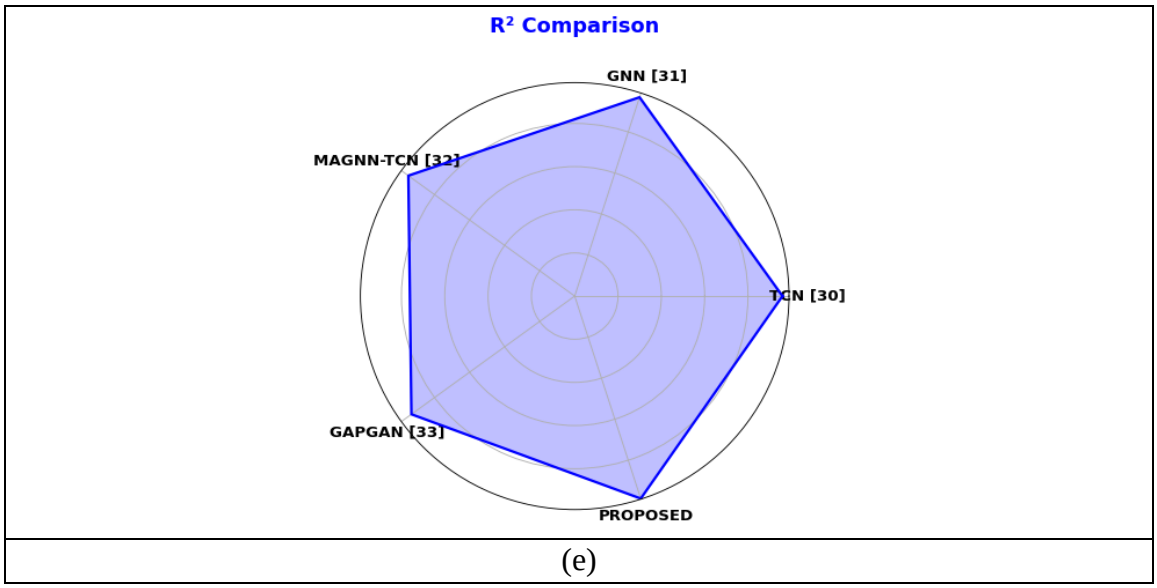
4.2. Algorithmic Analysis

The proposed oil and gas production prediction model and the attained results are given in this section. Here, the performance metrics used to evaluate the model performance are Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and Coefficient of Determination (R^2). Figure 4 illustrates the performance of proposed CompQual-TGNet over other existing DL models. Across all evaluation metrics, the proposed model delivers surpassing performance against existing methods TCN, GNN, MAGNN-TCN and GAPGAN.

The proposed CompQual-TGNet model delivers optimal predictive results by attaining RMSE of 0.0136 and MSE of 0.0264 together with MAE of 0.0138 and MAPE of 0.0154. The proposed model achieves exceptional performance in R^2 terms (0.9867) and best data fit capability for capturing variances. The proposed CompQual-TGNet model demonstrates the fastest testing time at 3.12 ms which confirms its capability to operate efficiently in real-time applications. The proposed solution demonstrates superior robustness and efficiency because of its complete system enhancement.

Figure 4. Performance of Proposed CompQual-TGNet Model over Existing Models for (a) RMSE, (b) MSE, (c) MAE, (d) MAPE, and (e) R^2





(e)

Table 3. Performance of Proposed CompQual-TGNet Over Other Existing Models for Various Metrics

Model	RMSE	MSE	MAE	MAPE	R^2	Testing Time (ms)
TCN [30]	0.0819	0.0606	0.0528	0.0422	0.9593	5.16
GNN [31]	0.0623	0.0521	0.0429	0.0368	0.9688	6.04
MAGNN-TCN [32]	0.0776	0.0621	0.0695	0.0679	0.9494	6.54
GAPGAN [33]	0.0786	0.0594	0.0749	0.0428	0.9319	5.48
Proposed CompQual-TGNet	0.0136	0.0264	0.0138	0.0154	0.9867	3.12

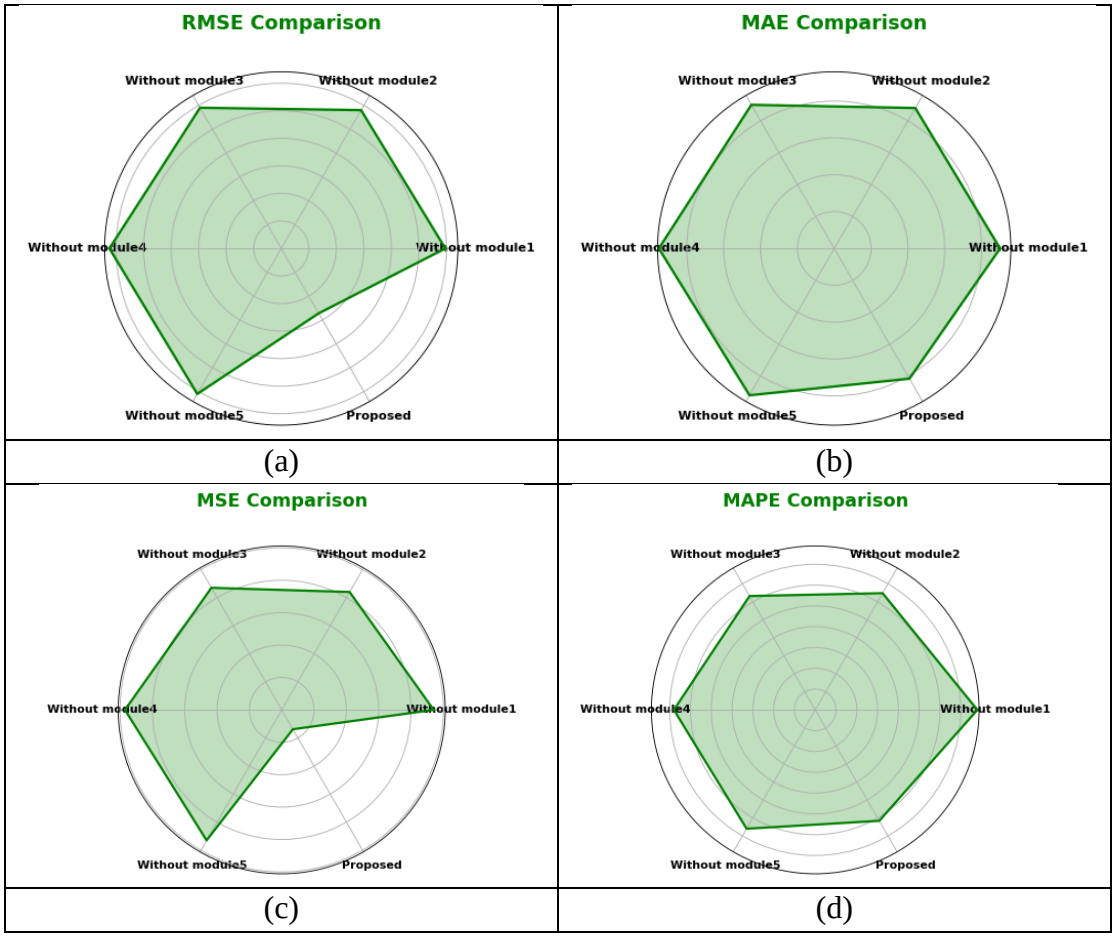
Table 3 demonstrates the performance of proposed CompQual-TGNet over other existing models for various metrics. The proposed model delivers significantly better performance than TCN, GNN, MAGNN-TCN, and GAPGAN during all evaluation metrics assessments. The proposed CompQual-TGNet model delivers optimal prediction accuracy based on its lowest RMSE of 0.0136 together with MSE of 0.0264, MAE of 0.0138 and MAPE of 0.0154. The proposed CompQual-TGNet model achieves maximum data variance capture by producing an R^2 value of 0.9867 along with the highest goodness-of-fit among models tested. Tests run on the proposed model show the fastest execution times at 3.12 ms which demonstrates its suitability for applications requiring live processing. The proposed approach demonstrates both enhanced robustness and operational efficiency through this extensive enhancement.

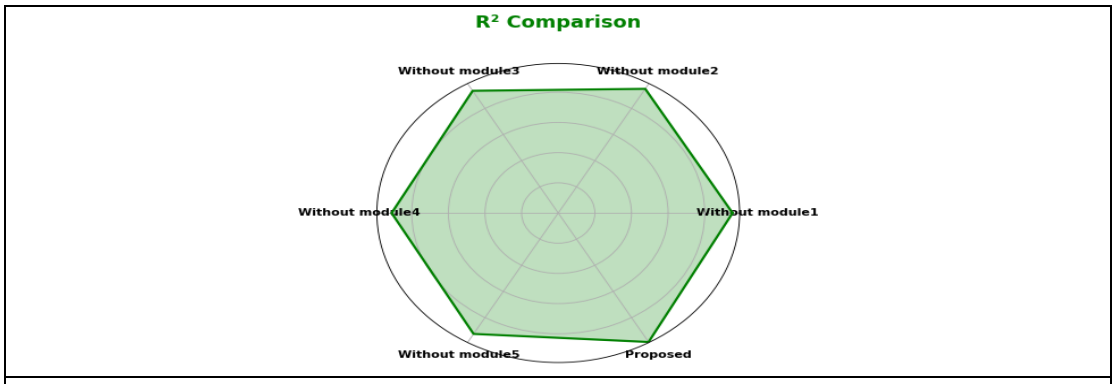
Table 4. Ablation Study

Methods	RMSE	MAE	MSE	MAPE	R^2
Without Multi-Resolution Temporal Block (Module 1)	0.0298	0.0225	0.094	0.0195	0.952
Without Adaptive Temporal Attention (Module 2)	0.029	0.022	0.084	0.0162	0.95
Without Feature Gating Mechanism (Module 3)	0.0295	0.0225	0.087	0.0158	0.935
Without GNN Architecture (Module 4)	0.0312	0.0238	0.097	0.017	0.91
Without Hierarchical Graph Pooling (Module 5)	0.0305	0.023	0.093	0.0165	0.925
Proposed CompQual-TGNet	0.0136	0.0204	0.0138	0.0154	0.9867

Table 4 explains the role of each module in the proposed CompQual-TGNet. In the context of the ablation study, the performance of the proposed CompQual-TGNet model is evaluated by sequentially removing key modules to observe their impact on different performance metrics: RMSE, MAE, MSE, MAPE, and R^2. All model components contribute to performance outcomes as confirmed by the best results achieved by the complete model. Without the multi-resolution temporal block (module 1), the model's RMSE increases slightly to 0.0298, and other metrics like MSE and MAE also show moderate degradation. When adaptive temporal attention (module 2) and feature gating mechanism (module 3) are excluded the performance metrics demonstrate deterioration as RMSE reaches 0.029 and RMSE reaches 0.0295. The model demonstrates decreased prediction accuracy and limited generalization when GNN architecture (module 4) and hierarchical graph pooling (module 5) are removed because both metrics increase to 0.0312 and 0.0305. The CompQual-TGNet model achieves the best performance due to its complete implementation resulting in an RMSE of 0.0136, MSE of 0.0138 and an exceptional R^2 of 0.9867.

Figure 5. Performance of Proposed CompQual-TGNet with respect to Each Module in the Proposed Architecture for (a) RMSE, (b) MSE, (c) MAE, (d) MAPE, and (e) R^2

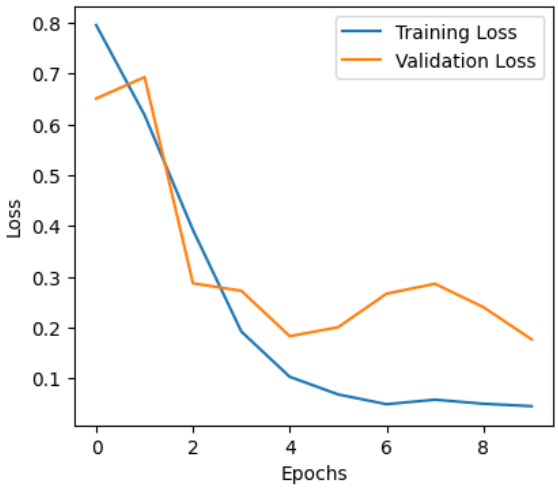




(e)

Figure 5 demonstrates the graphical representation of role of each module in the proposed CompQual-TGNet. The RMSE, MSR and R^2 metrics experience performance declines due to each component removal within the ablation study. The CompQual-TGNet model proves the significance of its components by reaching its highest performance through intact functionality. Without the critical modules that include Multi-Resolution Temporal Block and GNN Architecture model accuracy declines while error metrics rise. Figure 6 delivers the loss graph of proposed CompQual-TGNet.

Figure 6. Loss Graph of Proposed CompQual-TGNet Model



5. Conclusion

This study presented CompQual-TGNet, a new hybrid DL framework that analyzed the operations of oil and gas businesses using competency metrics and quality indicators. By merging GNN and TCN technologies, the system examined multiple operational datasets and produced accurate predictions regarding future performance. The preprocessing system established strong data linkages by standardizing values using Z-score normalization, detecting outliers using IQR, and matching temporal data patterns across several inputs. The system also employed sophisticated feature engineering methods to enhance outputs, such as feature extraction and fusion. The proposed method used MVMF to extract common latent features and CDAM to discover correlations. Furthermore, DFIN was utilized to discover the optimal feature combinations across datasets, and the cross-attention fusion layer was used to create context-aware

feature representations. Additionally, this study combined TCN for tracking operational history and long-term temporal patterns with GNN for mapping cross-dependencies between personnel capabilities and operational components. Weighting and attention modules were used by the adaptive fusion layer to bring these models together and produce precise forecasts based on domain-relevant predictions. Thus, this proposed model accomplished better results and proved its competence in predicting oil and gas production. In future, the research will upgrade CompQual-TGNet to analyze big operational datasets from multiple oil and gas companies. Besides, real-time input data feeds plus automatic model updates are included to make this system respond better to changing operational conditions. Finding better ways to optimize the model through advanced algorithms and adding external factors will help make better operational decisions.

6. References

1. Abdulridha, N. A., & Fathi, M. S. (2024). Enhancing organizational performance in oil and gas industry: a comprehensive review of quality management practices. Journal of Applied Engineering Design & Simulation (JAEDS), 4(1), 1-12.
2. Ebisinkemefa, T., & Teddy, A. L. (2024). Competency Management And Organizational Resilience In Retail Oil And Gas Firms In Bayelsa State. BW Academic Journal, 9-9.
3. Abdulla, H., & McCauley-Smith, C. (2024). Contribution mechanisms of project managers' behavioural competencies towards the success of oil and gas projects. International Journal of Construction Management, 24(2), 151-160.
4. Karachurina, R. F., Gulin, D. A., Burenina, I. V., & Sayfullina, S. F. (2024). New approaches to training engineers for the oil and gas industry. In E3S Web of Conferences (Vol. 486, p. 04019). EDP Sciences.
5. R Azmi, P. A., Yusoff, M., & Mohd Sallehud-din, M. T. (2024). A Review of Predictive Analytics Models in the Oil and Gas Industries. Sensors, 24(12), 4013.
6. Carrillo, P., Ohaegbu, P., & Anumba, C. A Framework for Strategic Decision-Making: Oil and Gas Industry Perspectives. Available at SSRN 4812508.
7. Al-Hajri, A., Abdella, G. M., Al-Yafei, H., Aseel, S., & Hamouda, A. M. (2024). A systematic literature review of the digital transformation in the Arabian gulf's oil and gas sector. Sustainability, 16(15), 6601.
8. Lawal, A., Yang, Y., He, H., & Baisa, N. L. (2024). Machine Learning in Oil and Gas Exploration-A Review. IEEE Access.
9. Ismail, F. B., Yuhana, M. I. F., Mohammed, S. A., & Sabri, L. S. (2024). Analytical Prediction of Gas Hydrate Formation Conditions for Oil and Gas Pipeline. Russian Journal of Applied Chemistry, 1-10.
10. Babayeju, O. A., Adefemi, A., Ekemezie, I. O., & Sofoluwe, O. O. (2024). Advancements in predictive maintenance for aging oil and gas infrastructure. World Journal of Advanced Research and Reviews, 22(3), 252-266.
11. Li, T., Tan, Y., Ahmad, F. A., & Liu, H. (2024). A new method to production prediction for the shale gas reservoir. Energy Sources, Part A: Recovery, Utilization, And Environmental Effects, 46(1), 9856-9869.
12. Zhou, W., Li, X., Qi, Z., Zhao, H., & Yi, J. (2024). A shale gas production prediction model based on masked convolutional neural network. Applied Energy, 353, 122092.

13. Xu, Z., & Leung, J. Y. (2024). A novel formulation of RNN-based neural network with real-time updating–An application for dynamic hydraulic fractured shale gas production forecasting. Geoenergy Science and Engineering, 233, 212491.
14. Chen, G., Tian, H., Xiao, T., Xu, T., & Lei, H. (2024). Time series forecasting of oil production in Enhanced Oil Recovery system based on a novel CNN-GRU neural network. Geoenergy Science and Engineering, 233, 212528.
15. Shaik, N. B., Jongkittinarukorn, K., Benjapolakul, W., & Bingi, K. (2024). A novel neural network-based framework to estimate oil and gas pipelines life with missing input parameters. Scientific Reports, 14(1), 4511.
16. Oyewola, D. O., Akinwunmi, S. A., & Omotehinwa, T. O. (2024). Deep LSTM and LSTM-Attention Q-learning based reinforcement learning in oil and gas sector prediction. Knowledge-Based Systems, 284, 111290.
17. Zhou, Q., Lei, Z., Chen, Z., Wang, Y., Liu, Y., Xu, Z., & Liu, Y. (2024). Shale oil production predication based on an empirical model-constrained CNN-LSTM. Energy Geoscience, 5(2), 100252.
18. Chen, G., Tian, H., Xiao, T., Xu, T., & Lei, H. (2024). Time series forecasting of oil production in Enhanced Oil Recovery system based on a novel CNN-GRU neural network. Geoenergy Science and Engineering, 233, 212528.
19. Fargalla, M. A. M., Yan, W., Deng, J., Wu, T., Kiyingi, W., Li, G., & Zhang, W. (2024). TimeNet: Time2Vec attention-based CNN-BiGRU neural network for predicting production in shale and sandstone gas reservoirs. Energy, 290, 130184.
20. Yang, J. Q., Lin, N. T., Zhang, K., Cui, Y., Fu, C., & Zhang, D. (2024). Deep learning CNN-APSO-LSSVM hybrid fusion model for feature optimization and gas-bearing prediction. Petroleum Science.
21. Zhiyuan, F., & Jiang, D. U. (2024). Prediction of Dissolved Gas Volume Fraction in Transformer Oil Based on Correlation Variational Mode Decomposition and CNN-LSTM. High Voltage Engineering, 50(1), 263-273.
22. Mensah, J. A., Appati, J. K., Boateng, E. K., Ocran, E., & Asiedu, L. (2024). Facenet recognition algorithm subject to multiple constraints: assessment of the performance. Scientific African, 23, e02007.
23. Rahmad Ramadhan, L., & Anne Mudya, Y. (2024). A Comparative Study of Z-Score and Min-Max Normalization for Rainfall Classification in Pekanbaru. Journal of Data Science, 2024(04), 1-8.
24. Zhang, H., Chen, H., Zhou, C., Chen, K., Liu, C., Zou, Z., & Shi, Z. (2024). Bifa: Remote sensing image change detection with bitemporal feature alignment. IEEE Transactions on Geoscience and Remote Sensing.
25. Luo, Y., Ke, W., Lam, C. T., & Im, S. K. (2024). An accurate slicing method for dynamic time warping algorithm and the segment-level early abandoning optimization. Knowledge-Based Systems, 300, 112231.
26. Zhang, J., Xu, H., Liu, W., Li, C., & Chen, Y. (2024). Information-Theoretic Approach to Joint Design of Waveform and Receiver Filter with Desired Cross-Correlation Properties for Imaging Radar. IEEE Transactions on Geoscience and Remote Sensing.
27. Magar, V., Ruikar, D., Bhoite, S., & Mente, R. (2024). Innovative Inter Quartile Range-based Outlier Detection and Removal Technique for Teaching Staff Performance Feedback Analysis. Journal of Engineering Education Transformations, 37(3).

28. Huang, S., Zhang, Y., Fu, L., & Wang, S. (2022). Learnable multi-view matrix factorization with graph embedding and flexible loss. IEEE Transactions on Multimedia, 25, 3259-3272.
29. Subedi, B., Sathishkumar, V. E., Maheshwari, V., Kumar, M. S., Jayagopal, P., & Allayear, S. M. (2022). [Retracted] Feature Learning-Based Generative Adversarial Network Data Augmentation for Class-Based Few-Shot Learning. Mathematical Problems in Engineering, 2022(1), 9710667.
30. Fan, J., Zhang, K., Huang, Y., Zhu, Y., & Chen, B. (2023). Parallel spatio-temporal attention-based TCN for multivariate time series prediction. Neural Computing and Applications, 35(18), 13109-13118.
31. Makhdomi, A. A., & Gillani, I. A. (2024). GNN-based passenger request prediction. Transportation Letters, 16(10), 1237-1251.
32. Ye, Y., Wang, J., Yang, J., Yao, D., & Zhou, T. (2024). Adaptive MAGNN-TCN: An Innovative Approach for Bearings Remaining Useful Life Prediction. IEEE Sensors Journal.
33. Liu, X., Yu, J., Gong, L., Liu, M., & Xiang, X. (2024). A GCN-based adaptive generative adversarial network model for short-term wind speed scenario prediction. Energy, 294, 130931.