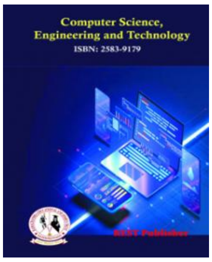
Computer Science, Engineering and Technology

Vol: 3(4), December 2025

REST Publisher; ISSN: 2583-9179

Website: https://restpublisher.com/journals/cset/

DOI: https://doi.org/10.46632/cset/3/4/3



A Comparative Study of Recurrent Neural Network (RNN) with Gray Relational Analysis for Temporal Data

***Chinnasami Sivaji, Anusuya Mohan, Soniya Sriram, M. Ramachandran**

REST Labs, Kaveripattinam, Krishnagiri, Tamil Nadu, India.

*Corresponding Author Email: chinnasamirsri01@gmail.com

Abstract: *A Recurrent Neural Network (RNN) is a specialized form of neural network that is adept at handling sequential data by retaining information from prior inputs. In contrast to conventional feedforward neural networks, RNNs incorporate loops in their architecture, allowing them to leverage data from previous time steps to affect the current output. This characteristic renders RNNs especially effective for applications that involve sequences, including time-series forecasting, natural language processing, and speech recognition. A fundamental component of RNNs is their hidden state, which acts as a dynamic memory that is refreshed with each incoming input. This allows RNNs to capture dependencies across time steps, which is crucial for understanding context in sequences. In language modeling, the interpretation of a word often relies on the words that come before it, a task that Recurrent Neural Networks (RNNs) handle well. However, RNNs struggle with issues like vanishing gradients, which hinder their ability to capture long-range dependencies. To overcome this, models such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) were introduced. These models incorporate gates that regulate the flow of information, allowing them to better learn long-term dependencies. RNNs remain a powerful tool for working with sequential data, facilitating the modeling of temporal relationships, but their effectiveness depends on careful design and optimization. Research significance: Recurrent Neural Networks (RNNs) hold significant research value because of their capacity to simulate temporal and sequential data, which is essential in many fields. They are frequently employed in natural language processing for tasks such as sentiment analysis, language translation, and text generation. In time-series analysis, RNNs enable accurate forecasting in finance, healthcare, and climate modeling. They also are essential in speech recognition and video processing, handling dependencies across time steps. Research focuses on improving RNNs, addressing challenges like vanishing gradients, and enhancing efficiency through architectures like LSTMs and GRUs, solidifying their relevance in advancing AI and machine learning applications. Methodology: A technique for analyzing the relationships between several variables, particularly in situations when data is limited or unclear, is called gray relational analysis, or GRA. In order to comprehend the relationships between variables, it evaluates how similar or different they are. GRA aids decision-makers in identifying critical factors, prioritizing actions, and improving processes in complex fields like engineering, finance, and management. By converting both qualitative and quantitative data into gray numbers, GRA addresses uncertainty and provides valuable insights for problem-solving, decision-making, and performance improvement, leading to more informed and effective strategies. Alternative taken as Simple RNN, LSTM, GRU, Bidirectional RNN, Deep RNN, Vanilla RNN, Echo State Network, Attention-based RNN, Transformer RNN, GRU with Attention. Evaluation preference taken as Prediction Accuracy, Model Robstness, Learning Efficiency, Training Time, Complexity. Attention-based RNN has the lowest score, Deep RNN has the highest rank, according to the results.*

Keywords: *Simple RNN, LSTM, GRU, GRA.*

1. INTRODUCTION

An artificial neural network type called a recurrent neural network (RNN) was created especially to process sequential data in which logical and temporal links between pieces are crucial. RNNs can operate as a memory by maintaining a concealed state thanks to their recurrent connections. Because of this capability, the network may store data from earlier processing stages and utilize it to instruct later processing stages. This memory-like characteristic enables RNNs to identify patterns in sequences and generate predictions depending on input context. Sequential tasks like pattern analysis, natural language processing, and speech recognition are frequently performed with them. [1] Each time step updates RNNs maintain a hidden state, which holds data from the previous time step. The hidden state from the previous time step is used in conjunction with the current input to determine the hidden state at the current time step. A backpropagation variant designed for continuous data, Backpropagation through time (BPTT), is used to train RNNs. The vanishing gradient problem can make it difficult for stable RNNs to detect long-range dependencies. To solve this problem and to increase memory capacity, variants such as long short-term memory (LSTM) networks and gate sequence units (GRU) are being developed. [2] RNN applications include tasks like sequential data analysis, time series forecasting, and voice detection. Sentiment analysis, automatic translation, and language modeling are all applications of RNNs in natural language processing. RNN technology has significantly improved sequential data analysis and decision-making in a number of different industries. Enhancing RNNs' interpretability, explainability, and capacity to handle longer sequences are the goals of future research. [3] Recurrent neural networks (RNNs) are used to process sequential input, where temporal connections and order of fragments are important. Video analysis, audio recognition, time series prediction, machine translation, natural language processing, and more can all benefit greatly from RNNs because they have made significant contributions to raising the bar in these domains. When processing sequential input, such as text and audio, where timing and ordering of data are important for understanding and prediction, RNNs are a type of artificial neural network that is particularly well suited to. [4] Human Activity Recognition (HAR) is a deep learning technique based on a recurrent neural network (RNN). We use a multimodal matrix (i.e., different body part angles over time) to represent the unique joint angles of the body parts over time. We next train and evaluate our RNN for HAR using these features. To evaluate the performance of our system, we used the MSRC-12 dataset from the Microsoft Research Center in Cambridge to compare the results of an RNN-based HAR with conventional HMM- and deep belief network (DBN)-based HAR. The RNN-based HAR consistently recognizes twelve human actions, outperforming traditional HMM- and DBN-based approaches, according to experimental results. An average classification accuracy of 99.55% was achieved. [5] The findings indicate an accuracy improvement of 2.01% and 7.06%, respectively, over the HMM-based and DBN-based HARs. The neural memory compressor is a stack of RNNs used to build an unsupervised learning system. By learning to predict the subsequent input based on the preceding ones, the RNN efficiently compresses the data at each level of the hierarchy. On occasion, the higher-level RNN recalculates its internal functions with only the rare inputs at each level as inputs. [6] The data received from the RNNs undergoes compression through this process, where the top-level RNN learns to reconstruct the original input sequence accurately. By minimizing the description word or the negative logarithm of the data likelihood, the system efficiently identifies and compresses important patterns in the input data. If there is substantial predictability in the source data, the highest-level RNN can categorize complex sequences with long gaps between key events using supervised learning. This neural history compressor can be represented by two RNNs: the "conscious" chunker (upper level) and the "subconscious" automatized (lower level). [7] The chunker develops the ability to predict and compress inputs that the automated system initially finds unpredictable. In turn, the automated system is trained to predict or replicate the slower-changing aspects of the chunker, allowing it to retain essential memories over a longer period. This process enables the automated system to predict many previously unpredictable inputs, allowing the chunker to focus on the remaining uncertainties. Notably, this approach helps address the vanishing gradient problem that often occurs during backpropagation in deep neural network training. [8] In 1993, the system accomplished a "Very Underground Learning" task that required the addition of over 1000 layers to a Recurrent Neural Network (RNN) over time, a capability that had already been demonstrated in 1992. One kind of artificial neural network that is well-known for their ability to process input sequences by utilizing internal memory or state. This memory enables them to handle inputs of capture temporal relationships at different lengths between elements in a sequence. [9] RNNs are bidirectional networks, meaning information can flow Both forward and backward. As a result, the network may learn from both historical and prospective context output from a particular node can affect later inputs to the same nodes. Applications of RNNs include tasks like unsegmented continuous handwriting and speech recognition, where They work especially well with sequential data because of the temporal correlations and order of the data are crucial. [10] Recurrent neural networks (RNNs), characterized by their infinite impulse response, are often referred to as IIRNs. The recurrent connections in these networks lead to a temporally dynamic behavior. In contrast, "convolutional neural networks" (CNNs) have a finite impulse response and are usually employed for jobs such as image recognition, as they do not include recurrent connections. Gated Memory, also known as gated state, refers to RNNs' ability to incorporate additional stored states that are directly managed by the network. [11] The issues of disappearing gradients and long-term dependencies are addressed by RNN architectures that include gated memory, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks. RNNs are sometimes called rebound neural networks (FNNs) due to the feedback loop created by their recurrent connections. These networks are Turing complete, meaning they can theoretically simulate any computation. This makes RNNs powerful and flexible for processing input sequences and solving complex problems. With this training dataset, RNNs have been trained. [12] If new individual signatories are incorporated into the existing system while the The suggested system is being used in practical situations, the classifier must be retrained using the new signatures. As a result, the proposed system functions in a WDM mode. To determine whether a signature is genuine or not, the characteristic vectors of all genuine test samples and all false signature test samples are compared. This is accomplished by feeding the test sample's feature vector into the RNN's LSTM and BLSTM models, after which the output layer of the RNN is used to retrieve the probability value for the predicted class. [13-15] These models have the potential to accelerate typical reservoir engineering tasks such as history matching and optimization. While the deep neural network (DNN) model delivers faster predictions, the RNN model provides better quality, based on a comparison of the two methods. Moreover, RNN-based proxy stream models can predict future times beyond the training data. In contrast to the full-physics flow simulator, both methods drastically cut down on computing time by as much as 100 times. A detailed search history matching task demonstrates the efficacy of the proxy flow model. [16] The changes are showcased using a hypothetical Brugge petroleum reservoir. Recurrent neural networks are utilized, and the model's performance is examined in both binary and multi-class classification tasks. We also analyze the impact of various learning rates and neuron counts on accuracy. In both binary and multi-class classifications, we evaluate RNN-IDS's performance against alternative machine learning methods. The experimental findings demonstrate RNN-IDS's superior intrusion detection capabilities. RNN-IDS improves detection accuracy and offers a fresh perspective on intrusion detection research by outperforming conventional classification techniques on the NSL-KDD dataset in both binary and multi-class tasks. [17-20]

2. MATERIALS & METHODS

Alternative: Simple RNN, LSTM, GRU, Bidirectional RNN, Deep RNN, Vanilla RNN, Echo State Network, Attention-based RNN, Transformer RNN, GRU with Attention.

1. Simple RNN

Simple RNN is the foundational architecture of RNNs. It consists of a single recurrent layer with weights shared across time steps. The input and the preceding hidden state are used to update the hidden state at each step. Its efficacy in complex tasks is limited because, although it can capture short-term dependencies, it has trouble with long-term dependencies because of vanishing gradients.

2. Long Short-Term Memory (LSTM)

By adding input, forget, and output gates that control the information flow, LSTM solves the vanishing gradient issue. The status of its cells acts as a memory buffer, selectively retaining or forgetting information over time. This makes LSTM effective for tasks requiring long-range dependency modeling, as time-series prediction, speech recognition, and language translation.

3. Gated Recurrent Unit (GRU)

GRU is a condensed form of LSTM that substitutes the hidden state for the cell state and merges the input and forget gates into a single update gate. This preserves the capacity to identify long-term relationships while lowering computational complexity. Because of their effectiveness, GRUs are frequently used, particularly for real-time applications like chatbots and streaming data processing.

4. Bidirectional RNN

By merging two RNNs, bidirectional RNNs process input sequences both forward and backward. As a result, the model can concurrently access past and future context. Bidirectional RNNs are very helpful for tasks like named entity recognition, speech recognition, and video captioning when context from both directions is crucial.

5. Deep RNN

Deep RNNs stack multiple RNN layers to form a deep network, enabling the extraction of hierarchical features from sequences. The additional layers enhance The ability of the model to recognize intricate patterns but also increase computational requirements. Deep RNNs are widely used in advanced applications like handwriting generation and multi-modal learning.

6. Vanilla RNN

Vanilla RNN refers to the basic form of RNN without any enhancements like gating mechanisms. While it is conceptually simple, it is highly prone to vanishing gradients and is generally unsuitable for tasks involving long-term dependencies. It serves as a baseline model for understanding RNNs.

7. Echo State Network (ESN)

Echo State Networks are a type of RNN that addresses training difficulties by using a sparsely connected recurrent layer with fixed weights. Only the output weights are trained. ESNs are computationally efficient and are used in tasks like dynamic system modeling and real-time processing, though they lack the flexibility of LSTMs or GRUs.

8. Attention-Based RNN

Attention mechanisms allow RNNs to concentrate on certain segments of the input sequence, enhancing performance in alignment-demanding activities like machine translation. Instead of relying solely on the hidden state, attention assigns weights to different time steps, enabling the model to prioritize relevant information and bypass long-term dependency issues.

9. Transformer RNN

Transformers revolutionized RNNs by entirely replacing recurrence with self-attention mechanisms. Though technically not recurrent, they are often grouped under RNN advancements. Transformers parallelize sequence processing and excel in handling long-range dependencies, forming the backbone of models like GPT and BERT. They dominate tasks like language modeling, summarization, and question-answering.

10. GRU with Attention

This hybrid combines GRUs with attention mechanisms, leveraging the efficiency of GRUs and the focus of attention. It is particularly effective in sequence-to- chores involving sequence, such text summarization and image captioning, where both computational efficiency and alignment between input and output sequences are crucial.

Evaluation preference: Prediction Accuracy, Model Robustness, Learning Efficiency, Training Time, Complexity.

1. Prediction Accuracy

Prediction accuracy refers to the model's ability to correctly predict outcomes based on input data. It is one of the most common metrics used to evaluate model performance, typically measured as the percentage of correct predictions. However, in certain tasks like classification or regression, other The F1 score, mean squared error (MSE), recall, precision, and others may be more relevant, depending on the problem context.

High Accuracy: Metrics like precision show how well the model works on unseen data and how well it has learned patterns from the training set.call, mean squared error (MSE), or F1 score

Challenges: Overfitting can occasionally result from achieving high accuracy, particularly if the dataset is unbalanced or the model is overly complex.

2. Model Robustness

The ability of a model to continue operating in the presence of hostile, noisy, or incomplete data is known as model resilience. A robust model can generalize well across different datasets and resist overfitting, ensuring that small changes in input data do not significantly affect its predictions.

High Robustness: Means the model performs consistently across various real-world conditions, even in the presence of data anomalies or unexpected input variations.

Challenges: In order to expose the model to a variety of inputs during training, rigorous regularization approaches and data augmentation strategies are frequently necessary to achieve resilience.

3. Learning Efficiency

Learning efficiency refers to how quickly and effectively a model can learn from the data. It involves the number of iterations or epochs required for the model to reach a certain level of performance and the computational resources needed. Efficient learning means that with fewer training samples and less effort, the model can generalize well. computational power.

High Efficiency: Indicates the model can learn in fewer iterations and with less data, which is particularly beneficial when data or computational resources are limited.

Challenges: Overfitting can occur if learning efficiency is prioritized at the cost of over-simplifying the model or underfitting the data.

4. Training Time

The term "training time" describes how long it takes to train a model from scratch, including the data preprocessing, model optimization, and validation steps. This factor is critical for deploying machine learning models in real-time or production environments where speed is essential. Training time can vary significantly depending on the quantity of the dataset, the model's intricacy, and the training apparatus.

Shorter Training Time: Means the model is faster to deploy and iterate upon. However, this can sometimes sacrifice model performance if not balanced properly.

Challenges: Long training periods may be necessary for complex models like deep neural networks, particularly when working with huge or high-dimensional datasets.

5. Complexity

Model complexity refers to the quantity of the model's layers and parameters, as well as the overall architecture's sophistication. More complex models can capture more intricate patterns but may require more data to train effectively and are more prone to overfitting.

High Complexity: Enables the model to learn highly detailed patterns and representations, but may suffer from longer training times, the need for more data, and increased susceptibility to overfitting.

Challenges: Striking a balance between underfitting and overfitting is key. Models ought to be intricate enough to identify the relevant patterns in the data, but not so intricate that they are hard to train or have poor generalization capabilities.

Grey Relational Analysis (GRA): Grey Relational Analysis (GRA), also known as Gray Correlation Analysis, is a technique used to address issues related to data envelopment analysis in facility management. It is especially valuable for decision-making involving layout and dispatch rules. The practical application of GRA involves a systematic comparison of different performance metrics, known as sequential translation. The process starts with the formation of associated data, where alternative performances are evaluated sequentially. Grey values for these performances are then compared to a reference set, allowing for the calculation of correlation coefficients. These grey values are ultimately linked through coefficients, reference sequences, and quality assessments for each comparison sequence. [21] GRA can be utilized to enhance drilling process parameters, particularly focusing on improving the surface roughness and burr height of workpieces. This involves using grey relational analysis combined with an orthogonal experimental design. By applying grey analysis to machining parameters, various performance characteristics, such as surface hardness and burr height, can be evaluated and optimized according to established standards. It's important to mention that there is a lack of published research on how cutting parameters affect different performance characteristics through grey relational analysis. [22] In 1989, Deng introduced grey relational analysis, which has become a popular tool among researchers for optimizing process parameters. This method has been applied to a range of areas, including die-sinking EDM machining, shape analysis, and finding the best parameters for Polycarbonate yield stress and elongation composite injection molding. Researchers frequently combine using the Taguchi method in conjunction with gray relational analysis to enhance and present analysis results, particularly in turn functions. The technique has also proven effective in optimizing processes such as extrusion for particle-reinforced materials and final dry grinding for high-purity graphite, highlighting its utility in improving parameters for machining. [23] One technique used in grey correlation analysis is the weighted average method, which takes into account multiple criteria in practical decision-making scenarios like ordering goods. This technique involves comparing data sets on both global and local scales. Its major advantage is its flexibility in accommodating different parameters within the model, which helps minimize potential negative impacts on the system. By using ordered pairs and linking resulting domains, this paper presents a domain-combination approach specifically designed for the Grey correlation analysis model. [24] The Istanbul Stock Exchange (ISE) employs gray correlation analysis (GRA) to rank shares of companies listed in the financial sector index. GRA is well-regarded worldwide for its effectiveness in preserving hierarchical structures while allowing for comparability across various markets. To ensure impartiality, all criteria are equally weighted as decision factors. However, in complex decision-making models where performance characteristics are weighted across different hierarchical levels, adjustments may be needed to uphold both accuracy and fairness. [25-26] Gray correlation analysis (GRA) is essential for improving wastewater treatment options and is closely tied to selection analysis. It proves particularly useful in challenging scenarios involving incomplete or uncertain data. As a key element in gray system applications, GRA is instrumental in addressing complex relationships among multiple performance factors. By optimizing these relationships and managing interdependencies, GRA effectively enhances overall performance. [27] Gray relational analysis is used to tackle problems related to turning functions by identifying optimal cutting parameters. This process involves using the Taguchi method to evaluate gray relational quality. Initially, the Taguchi method and gray relational analysis are applied for optimization. Subsequently, a detailed investigation is carried out to select the best cutting and turning parameters, which helps in assessing machine performance during operational tasks. [28] In gray correlation analysis, electrode wear begins at a baseline of zero and is then adjusted to a standardized gray level, a process known as correlation formation.

Systematically identifying the best machining parameters using gray relational analysis involves evaluating various performance criteria to ultimately determine the most effective parameters. [29] The benefits of employing the Grey Relational Analysis (GRA) approach, which is backed by primary data, are mostly found in correlation analysis (GRA) and multi-attribute decision making (MADM). The calculations involved are simple and understandable. GRA is considered to be one of the best methods for supporting managerial decision-making in corporate settings. Gray relational analysis is used to identify characteristics related to several functional elements, such as surface removal rate, maximum surface area, and hardness across 203 particles, in order to improve wire electrical discharge machining (WEDM) for processing reinforced materials. In addition to assessing shear stress, this approach considers critical variables such cutting speed, feed rate, depth of cut, and machining time. [30]

Step 1. Decision matrix and weight matrix design

Let D=x_{ij} be a decision matrix for an MCDM issue with m choices and n criteria, where x_{ij}∈R.

$$D = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \quad (1)$$

Step 2: The decision matrix is normalized

Equations 2 and 3 are used to evaluate the normalization of two categories of data, i.e., better when higher type or better when lower. The data ranges from 0 to 1 after normalization.

$$M_{ij} = \frac{N_{ij} - \min(N_{ij})}{\max(N_{ij}) - \min(N_{ij})} \quad (2)$$

$$M_{ij} = \frac{\max(N_{ij}) - N_{ij}}{\max(N_{ij}) - \min(N_{ij})} \quad (3)$$

Where $i, j = 1,2,3, \cdots, n$

Step 3. $Deviation = the \max value\ after\ normalization - value\ of\ the\ current\ row \quad (4)$

Step 4: Gray relation coefficient calculation

$$C_{ij} = \frac{\Delta_{min} - \xi\Delta_{max}}{Current\ value - \xi\Delta_{max}}, where\ \xi\ is\ distinguishing\ coefficient \quad (5)$$

Step 5: Determine the Gray connection grade. It is the Gray relation coefficient average.

3. RESULT AND DISCUSSION

TABLE 1. Recurrent Neural Network (RNN)

	Prediction Accuracy	Model Robustness	Learning Efficiency	Training Time	Complexity
Simple RNN	70	65	75	60	55
LSTM	90	85	80	95	85
GRU	80	80	85	70	70
Bidirectional RNN	85	80	75	90	80
Deep RNN	78	72	65	100	95
Vanilla RNN	60	55	85	50	60
Echo State Network	72	75	70	45	50
Attention-based RNN	92	90	88	120	110
Transformer RNN	95	90	70	140	130
GRU with Attention	88	85	80	110	100

Table 1 provides a comparative analysis of various types of Recurrent Neural Networks (RNNs) based on five performance metrics: Prediction Accuracy, Model Robustness, Learning Efficiency, Training Time, and Complexity. These metrics help evaluate each model's strengths and trade-offs in different application scenarios. Simple RNN offers moderate performance across all metrics, but its lower robustness and complexity make it suitable for basic tasks. LSTM (Long Short-Term Memory) excels in prediction accuracy (90%) and robustness (85%), making it ideal for long-sequence data. However, it requires significant training time (95%) and complexity (85%). GRU (Gated Recurrent Unit) balances accuracy (80%) and efficiency (85%) with moderate training time (70%), making it a lightweight alternative to LSTM. Bidirectional RNN improves accuracy (85%) and robustness (80%) by learning from both past and future data but demands higher training time (90%). Deep RNN trades efficiency (65%) for high complexity (95%) and training time (100%), making it suitable for deep learning tasks. Vanilla RNN has the lowest overall performance, especially in robustness (55%) and training time (50%). Echo State Network offers moderate robustness (75%) but has lower complexity (50%), making it ideal for fast, lightweight tasks. Attention-based RNN and Transformer RNN dominate in accuracy (92%, 95%) and robustness (90%) but demand high training time (120%, 140%) and complexity (110%, 130%). GRU with Attention combines high accuracy (88%) and robustness (85%) with a reasonable balance in training time (110%).

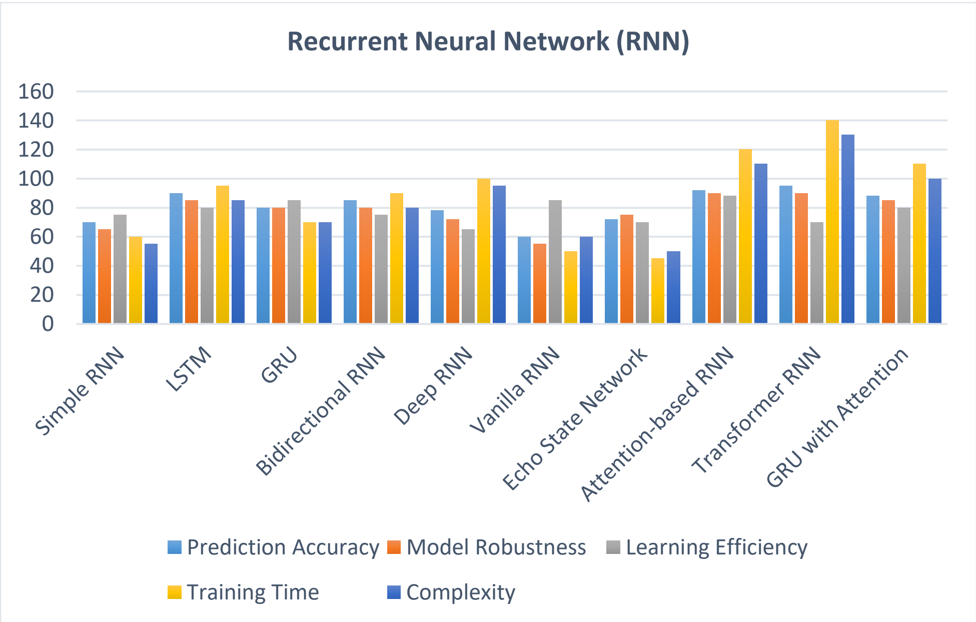


FIGURE 1. Recurrent Neural Network (RNN)

The graph 1 compares various Recurrent Neural Networks (RNNs) based on four key metrics: Prediction Accuracy, Model Robustness, Learning Efficiency, and Complexity. The models represented include Simple RNN, LSTM (Long Short-Term Memory), GRU (Gated Recurrent Unit), Bidirectional RNN, Deep RNN, Vanilla RNN, Echo State Network, Attention-based RNN, Transformer RNN, and GRU with Attention. Prediction Accuracy: Models like LSTM, GRU, and Transformer RNN show the highest prediction accuracy, suggesting these architectures are particularly adept at learning complex sequential patterns in data. These networks tend to perform well on tasks involving long-term dependencies or complex sequences. Model Robustness: LSTM and GRU, along with Bidirectional RNNs, demonstrate high model robustness. This indicates that these models are more stable and able to generalize well across a variety of tasks, coping effectively with variations in input data. Learning Efficiency: Simple RNN and Vanilla RNN exhibit lower learning efficiency. These models are limited in their ability to capture long-term dependencies, leading to slower learning and performance degradation on more complex tasks. More sophisticated models like LSTM, GRU, and Attention-based RNNs, on the other hand, show much higher learning efficiency, allowing them to learn faster and more effectively. Complexity: Transformer RNN, Deep RNN, and Attention-based RNN rank high in complexity, which corresponds to their more intricate architectures that enable improved performance but require greater computational resources. These models are typically used for tasks that require capturing long-range dependencies or intricate features in sequential data. The graph visually reflects the trade-offs between these RNN models. While simpler models like Vanilla RNN and Simple RNN are less complex and have faster training times, they fall short in prediction accuracy and model robustness. More complex models like Transformer RNN and LSTM are more powerful but come with greater computational demands.

TABLE 2. Normalized Data

Prediction Accuracy	Model Robustness	Learning Efficiency	Training Time	Complexity
0.2857	0.2857	0.4348	0.8421	0.9375
0.8571	0.8571	0.6522	0.4737	0.5625
0.5714	0.7143	0.8696	0.7368	0.7500
0.7143	0.7143	0.4348	0.5263	0.6250
0.5143	0.4857	0.0000	0.4211	0.4375
0.0000	0.0000	0.8696	0.9474	0.8750
0.3429	0.5714	0.2174	1.0000	1.0000
0.9143	1.0000	1.0000	0.2105	0.2500
1.0000	1.0000	0.2174	0.0000	0.0000
0.8000	0.8571	0.6522	0.3158	0.3750

Table 2 presents normalized data for five evaluation metrics—Prediction Accuracy, Model Robustness, Learning Efficiency, Training Time, and Complexity—across multiple models. The values, scaled between 0 and 1, facilitate easy comparison by standardizing the metrics. The table reveals notable trends in performance trade-offs. Models with higher normalized scores in Prediction Accuracy and Model Robustness (e.g., Row 9 with values of 1.0000 each) tend to have lower scores in Training Time and Complexity (both 0.0000), indicating significant computational costs for achieving high accuracy. Conversely, models in Row 6 and Row 7 exhibit low complexity and training time (e.g., Row 7 scores 1.0000 in Training Time but lower in other metrics), making them lightweight but less robust. Row 8 (Attention-based RNN) achieves the highest normalized values for Learning Efficiency (1.0000) and Model Robustness (1.0000) with a lower training time (0.2105), suggesting an efficient balance. The table highlights the inherent trade-offs in model design, emphasizing the need for tailored model selection based on application requirements.

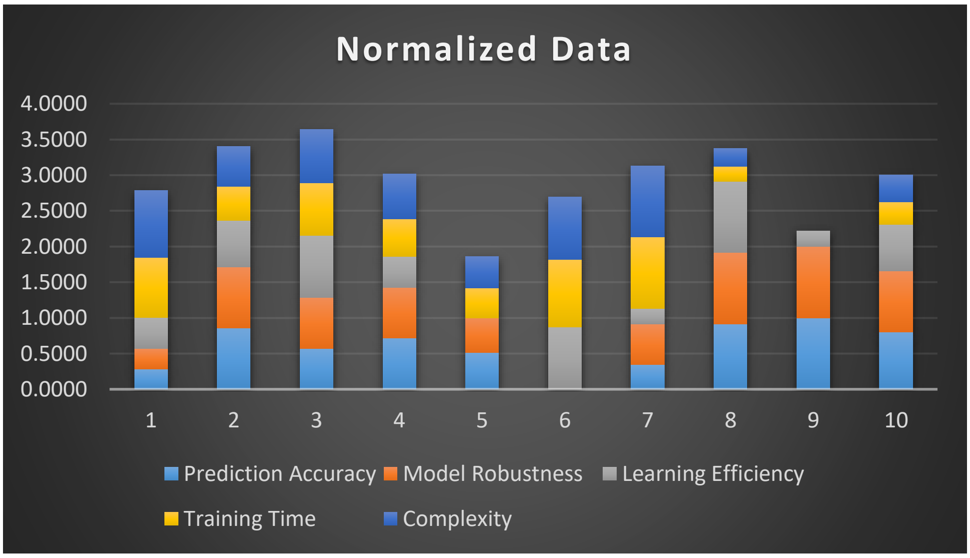


FIGURE 2. Normalized Data

The graph 2 labeled "Normalized Data" displays normalized values for various performance metrics: Prediction Accuracy, Model Robustness, Learning Efficiency, Training Time, and Complexity, across 10 different models or scenarios, represented on the x-axis. The metrics are color-coded for clarity: Prediction Accuracy is represented by blue bars. Model Robustness is represented by orange bars. Learning Efficiency is represented by gray bars. Training Time is represented by yellow bars. Complexity is represented by teal bars. The height of each stacked bar shows the combined values of all five metrics for each scenario, with individual metrics being stacked on top of one another. The use of normalized data means that the values are scaled, likely for easier comparison across models or scenarios, with each metric scaled to a common range, such as between 0 and 3. From the graph, we can observe: Prediction Accuracy tends to dominate in most scenarios, especially for models 2, 6, 7, and 10, where its value is noticeably higher than others. Model Robustness and Learning Efficiency also follow significant patterns but are usually lower in comparison to Prediction Accuracy. Training Time and Complexity are generally lower, though they can peak for specific scenarios like 7 and 9. This normalized presentation helps highlight the balance and trade-offs between these factors across different models or setups.

TABLE 3. Deviation sequence

Prediction Accuracy	Model Robustness	Learning Efficiency	Training Time	Complexity
0.7143	0.7143	0.5652	0.1579	0.0625
0.1429	0.1429	0.3478	0.5263	0.4375
0.4286	0.2857	0.1304	0.2632	0.2500
0.2857	0.2857	0.5652	0.4737	0.3750
0.4857	0.5143	1.0000	0.5789	0.5625
1.0000	1.0000	0.1304	0.0526	0.1250
0.6571	0.4286	0.7826	0.0000	0.0000
0.0857	0.0000	0.0000	0.7895	0.7500
0.0000	0.0000	0.7826	1.0000	1.0000
0.2000	0.1429	0.3478	0.6842	0.6250

Table 3 outlines the deviation sequence for five metrics—Prediction Accuracy, Model Robustness, Learning Efficiency, Training Time, and Complexity calculated as the absolute differences between normalized data values and their ideal reference points. These deviations help identify the extent to which each metric differs from optimal performance. Smaller deviation values indicate closer alignment with ideal performance. For instance, Row 9 demonstrates zero deviation for Prediction Accuracy and Model Robustness, suggesting a model that excels in these metrics but exhibits significant deviations in Training Time (1.0000) and Complexity (1.0000), highlighting high computational demands. Conversely, Row 6 shows high deviations for Prediction Accuracy (1.0000) and Model Robustness (1.0000), indicating trade-offs for reduced complexity. Notably, Row 8 (Attention-based RNN) shows minimal deviations in Prediction Accuracy (0.0857) and Model Robustness (0.0000), achieving a balanced performance despite moderate deviations in Training Time (0.7895) and Complexity (0.7500). These insights highlight how deviation sequences pinpoint strengths and weaknesses, aiding model optimization.

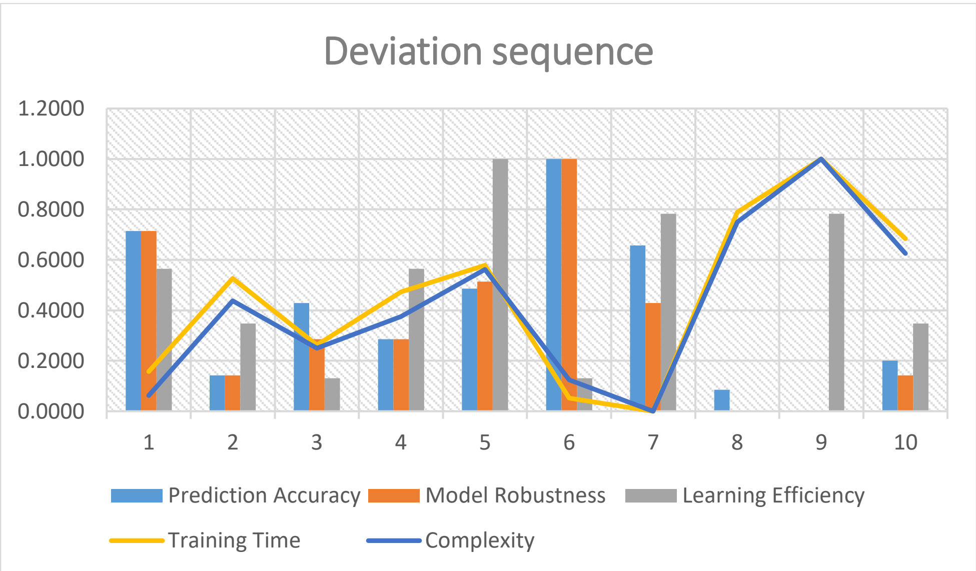


FIGURE 3. Deviation sequence

The "Deviation Sequence" chart illustrates the comparative evaluation of five key metrics across ten test cases or scenarios. These metrics are Prediction Accuracy, Model Robustness, Learning Efficiency, Training Time, and Complexity, represented by distinct bars and lines in the graph. The blue and orange bars indicate Prediction Accuracy and Model Robustness, respectively, highlighting how well the models perform and adapt to varying conditions. The gray bars show Learning Efficiency, It illustrates the models' capacity for generalization effectively within shorter durations. The yellow and blue lines depict Training Time and Complexity, demonstrating trends in computational resource utilization and overall design intricacy. The graph shows a dynamic relationship between these metrics. For instance, higher prediction accuracy and robustness may sometimes correlate with increased complexity and training time, as seen in cases like 6 and 8. Conversely, certain scenarios, such as 3 and 9, demonstrate reduced training time but lower efficiency or accuracy. These deviations highlight trade-offs inherent in optimizing models for balanced performance. Overall, the graph provides insights into the multi-dimensional evaluation of models, emphasizing the need for prioritizing specific parameters based on practical constraints and application requirements.

TABLE 4. Grey relation coefficient

Prediction Accuracy	Model Robustness	Learning Efficiency	Training Time	Complexity
0.4118	0.4118	0.4694	0.7600	0.8889
0.7778	0.7778	0.5897	0.4872	0.5333
0.5385	0.6364	0.7931	0.6552	0.6667
0.6364	0.6364	0.4694	0.5135	0.5714
0.5072	0.4930	0.3333	0.4634	0.4706
0.3333	0.3333	0.7931	0.9048	0.8000
0.4321	0.5385	0.3898	1.0000	1.0000
0.8537	1.0000	1.0000	0.3878	0.4000
1.0000	1.0000	0.3898	0.3333	0.3333
0.7143	0.7778	0.5897	0.4222	0.4444

Table 4 presents the Grey Relation Coefficients (GRC) for five metrics Prediction Accuracy, Model Robustness, Learning Efficiency, Training Time, and Complexity. The GRC values measure the closeness of each metric's performance to the ideal reference point, with higher values indicating stronger relationships and better alignment with optimal performance. For instance, Row 9 demonstrates perfect GRC scores (1.0000) for Prediction Accuracy and Model Robustness, indicating outstanding performance in these areas. However, the same row shows lower coefficients for Training Time (0.3333) and Complexity (0.3333), reflecting significant trade-offs in computational efficiency. Similarly, Row 8 (Attention-based RNN) achieves high GRC scores across Prediction Accuracy (0.8537), Model Robustness (1.0000), and Learning Efficiency (1.0000), emphasizing its balanced and strong performance. However, its lower scores in Training Time (0.3878) and Complexity (0.4000) highlight areas of computational intensity. Overall, the GRC values in Table 4 provide a comprehensive understanding of the performance balance across models, allowing for nuanced evaluation and informed decision-making in model selection.

TABLE 5. GRG & Rank

	GRG	**Rank**
Simple RNN	0.5884	8
LSTM	0.6332	4
GRU	0.6580	3
Bidirectional RNN	0.5654	9
Deep RNN	0.4535	10
Vanilla RNN	0.6329	5
Echo State Network	0.6721	2
Attention-based RNN	0.7283	1
Transformer RNN	0.6113	6
GRU with Attention	0.5897	7

Table 5 highlights the Grey Relational Grades (GRG) and their respective Ranks for various Recurrent Neural Network (RNN) models. GRG values represent the overall performance of each model, derived from the Grey Relational Coefficient (GRC) calculations. Higher GRG values indicate better performance in achieving an optimal balance across multiple evaluation criteria, such as Prediction Accuracy, Model Robustness, Learning

Efficiency, Training Time, and Complexity. The Attention-based RNN ranks first with the highest GRG score of 0.7283, signifying its superior ability to balance accuracy and robustness while maintaining reasonable efficiency. The Echo State Network ranks second with a GRG score of 0.6721, showcasing its competitive performance in these metrics despite being a simpler model. The GRU model, with a GRG of 0.6580, takes third place, reflecting its strong performance, especially in learning efficiency and training time. The LSTM model, another popular variant, ranks fourth with a GRG of 0.6332, indicating reliable performance across various tasks. At the lower end, the Deep RNN has the lowest GRG score of 0.4535, ranking tenth, due to its high training time and complexity. Similarly, the Bidirectional RNN ranks ninth with a GRG score of 0.5654, underperforming in certain metrics. These rankings provide a clear hierarchy of model effectiveness, aiding in selecting the most suitable architecture for specific applications.

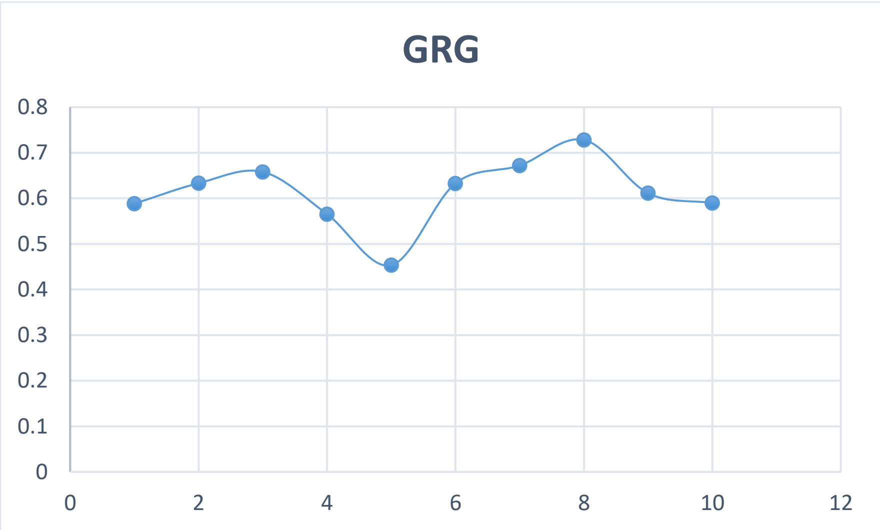


FIGURE 4. Grey Relational Grades (GRG)

The "Grey Relational Grades (GRG)" chart provides a visual representation of the GRG values for ten scenarios or test cases, which assess the overall performance of a system or model based on multiple criteria. GRG is a crucial component in Grey Relational Analysis (GRA), a technique used to evaluate and rank alternatives in decision-making problems by considering their closeness to the ideal solution. The plot reveals a fluctuating trend in GRG values, with the range spanning from approximately 0.5 to 0.8. Higher GRG values indicate better overall performance, as the evaluated system aligns more closely with the ideal solution. For example, cases around the 8th test exhibit higher GRG values, suggesting superior performance compared to other scenarios. Conversely, cases such as the 4th show a dip in GRG, reflecting lower relative performance. The chart highlights the variability in system effectiveness across different scenarios, driven by trade-offs in the underlying metrics (e.g., accuracy, robustness, efficiency). These insights guide stakeholders in identifying strengths and weaknesses and prioritizing optimizations to improve the overall system's alignment with desired outcomes. The GRG plot effectively summarizes multi-dimensional evaluations into a single comparative metric.

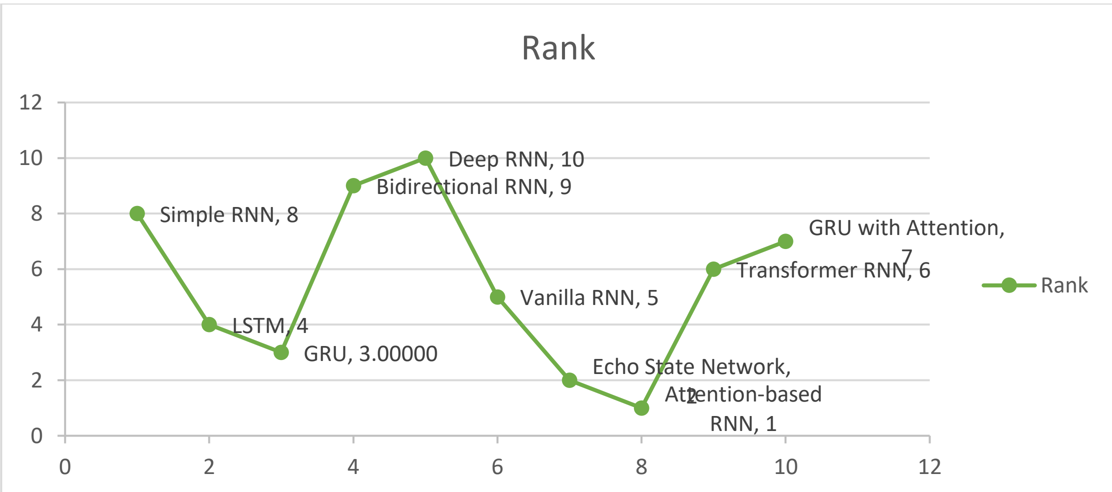


FIGURE 5. Rank

This graph, titled "Rank," compares the ranks of various recurrent neural network (RNN) models and architectures. The x-axis, labeled "Axis Title," appears to represent different models, while the y-axis, also labeled "Axis Title," indicates the ranks assigned to these models, ranging from 0 to 12. The data points represent specific RNN variants such as Simple RNN (rank 8), LSTM (rank 4), GRU (rank 3), and Attention-based RNN (rank 1). The graph shows a fluctuating trend, with certain models like the Deep RNN receiving a high rank (10), indicating strong performance, while others such as the Echo State Network are ranked lower (2). There is a noticeable peak with the Bidirectional RNN at rank 9 and a dip around Vanilla RNN (rank 5) and Echo State Network (rank 2). The line chart visually connects these points, making it easy to Compare the models' respective performances or preferences. The GRU with Attention and Transformer RNN are ranked 7 and 6, respectively, suggesting moderate effectiveness. This visualization effectively highlights the comparative performance of various RNN architectures in a given context.

4. CONCLUSION

An artificial neural network class called Recurrent Neural Networks (RNNs) is made to handle sequential data. RNNs are especially useful for jobs where context and order are essential because, in contrast to standard feedforward neural networks, they have the unusual capacity to retain information over time through recurrent connections. RNNs can represent relationships in sequential data, including time series, natural language, or video frames, because to this temporal dynamic. RNNs' primary strength is their feedback loops, which give them the ability to retain a kind of memory. They can efficiently capture both short-term and long-term dependencies in data thanks to this property. RNNs, for example, can understand both individual words and their contextual meanings in natural language processing by identifying patterns within phrases. Notwithstanding their promise, RNNs have several drawbacks, especially when managing long-term dependencies because to problems like vanishing and exploding gradients. Standard RNNs have trouble learning associations across long sequences because of these issues. Advanced versions like Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) networks were created to overcome these drawbacks. These architectures greatly enhance the network's capacity to store and process pertinent data over time by including features like gates to control the information flow. RNNs have transformed a number of fields, such as video analysis, language modeling, and speech recognition. They are also a key component of machine learning applications in sectors like banking, healthcare, and entertainment due to their capacity to interpret temporal and sequential data. RNNs are used, for instance, to create realistic text or music, diagnose illnesses from sequential medical data, and forecast stock market trends. However, the supremacy of RNNs has been challenged by the introduction of Transformer topologies. Transformers use attention mechanisms to examine whole input sequences at once, doing away with the necessity for sequential processing. They are therefore the better option for jobs like language translation and generative AI since they can scale more efficiently and handle data more quickly than RNNs. In summary, despite many drawbacks, RNNs continue to be a key and innovative technology in the deep learning space. Unquestionably, they have contributed to sequential data processing, and their development into more sophisticated forms like LSTMs and GRUs guarantees their ongoing applicability. As the area develops, RNNs will probably be crucial in resolving challenging issues involving sequential data, in conjunction with complementary and competitive architectures.

REFERENCES

[1]. Yin, Chuanlong, Yuefei Zhu, Jinlong Fei, and Xinzheng He. "A deep learning approach for intrusion detection using recurrent neural networks." Ieee Access 5 (2017): 21954-21961.

[2]. Peram, S. R. "Advanced Network Traffic Visualization and Anomaly Detection Using PCA-MDS Integration and Histogram Gradient Boosting Regression." Journal of Artificial Intelligence and Machine Learning 1, no. 3 (2023): 281.

[3]. Mou, Lichao, Pedram Ghamisi, and Xiao Xiang Zhu. "Deep recurrent neural networks for hyperspectral image classification." IEEE Transactions on Geoscience and Remote Sensing 55, no. 7 (2017): 3639-3655.

[4]. Murad, Abdulmajid, and Jae-Young Pyun. "Deep recurrent neural networks for human activity recognition." Sensors 17, no. 11 (2017): 2556.

[5]. Banerjee, Imon, Yuan Ling, Matthew C. Chen, Sadid A. Hasan, Curtis P. Langlotz, Nathaniel Moradzadeh, Brian Chapman et al. "Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification." Artificial intelligence in medicine 97 (2019): 79-88.

[6]. Bahad, Pritika, Preeti Saxena, and Raj Kamal. "Fake news detection using bi-directional LSTM-recurrent neural network." Procedia Computer Science 165 (2019): 74-82.

[7]. Kanumarlapudi, Praveen Kumar, Sudhakara Reddy Peram, and Sridhar Reddy Kakulavaram. "Evaluating Cyber Security Solutions through the GRA Approach: A Comparative Study of Antivirus Applications." *International Journal of Computer Engineering and Technology (IJCET)* 15, no. 4 (2024): 1021-1040.

[8]. Peram, S. R. "Automated Label Detection and Recommendation System Using Deep Convolution Neural Networks and SPSS-Based Evaluation." *International Journal of Computer Science and Data Engineering* 1, no. 2 (2024): 258.

[9]. Sehovac, Ljubisa, and Katarina Grolinger. "Deep learning for load forecasting: Sequence to sequence recurrent neural networks with attention." Ieee Access 8 (2020): 36411-36426.
[10]. Peram, S. R. "Automated Label Detection and Recommendation System Using Deep Convolution Neural Networks and SPSS-Based Evaluation." International Journal of Computer Science and Data Engineering 1, no. 2 (2024): 258.
[11]. Yu, Yong, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. "A review of recurrent neural networks: LSTM cells and network architectures." Neural computation 31, no. 7 (2019): 1235-1270.
[12]. Roy, Bipraneel, and Hon Cheung. "A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network." In 2018 28th international telecommunication networks and applications conference (ITNAC), pp. 1-6. IEEE, 2018.
[13]. Kurinjimalar Ramu, M. Ramachandran, Ramya sharma, Prabakaran Nanjundan, "Urban Agriculture Overview of Sustainability Using GRA Methodology", Building Materials and Engineering Structures, 2(3), September 2023, 7-14.
[14]. Sukumaran, C., M. Ramachandran, Chinnasami Sivaji, and Manjula Selvam. "Ranking of Product in E-store using WASPAS method." REST Journal on Banking, Accounting and Business 1, no. 1 (2022): 1-9. https://doi.org/10.46632/jbab/1/1/1
[15]. Venkateswaran, C., D. R. Pallavi, M. Ramachandran, Sathiyaraj Chinnasamy, and Chinnasami Sivaji. "A Study on Weighted Aggregated Sum Product Assessment (WASPAS) wrt Multiple Criteria Decision Making." Data Analytics and Artificial Intelligence 2, no. 1 (2022): 26-33. https://doi.org/10.46632/daai/2/1/5
[16]. Kong, Weicong, Zhao Yang Dong, Youwei Jia, David J. Hill, Yan Xu, and Yuan Zhang. "Short-term residential load forecasting based on LSTM recurrent neural network." IEEE transactions on smart grid 10, no. 1 (2017): 841-851.
[17]. Tortora, Stefano, Stefano Ghidoni, Carmelo Chisari, Silvestro Micera, and Fiorenzo Artoni. "Deep learning-based BCI for gait decoding from EEG with LSTM recurrent neural network." Journal of neural engineering 17, no. 4 (2020): 046011.
[18]. Kanumarlapudi, Praveen Kumar, and S. Ballamudi. "Enhancing Generative AI Shopping Assistants through Advanced Multi-Attribute Decision Making Technique." Journal of Artificial Intelligence and Machine Learning 3, no. 2 (2025): 1-7.
[19]. Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." In 2013 IEEE international conference on acoustics, speech and signal processing, pp. 6645-6649. Ieee, 2013.
[20]. Peram, S. R. "Advanced Network Traffic Visualization and Anomaly Detection Using PCA-MDS Integration and Histogram Gradient Boosting Regression." *Journal of Artificial Intelligence and Machine Learning* 1, no. 3 (2023): 281.
[21]. Thota, Sandeep Kumar, Kumari Gubbala, Ashok Polavarapu, Vikram Narayandas, Hari Suresh Babu Gummadi, Narendra Chennupati, Sreedhar Babu Seshagani, Shivakrishna Deepak Veeravalli, and Manisha Guduri. "Adversarial Training with Attention-Guided DCGAN for Robust Lung Segmentation in Medical Imaging." In *2025 IEEE Region 10 Symposium (TENSYMP)*, pp. 1-6. IEEE, 2025.
[22]. Nayeemuddin, . and Salma, Umme, Analysis of a Drilling Hybrid Aluminum Metal Matrix Composites by Using SPSS (May 7, 2023). Int. J. Adv. Res. 11(01), 82-91, 2023, Available at SSRN: https://ssrn.com/abstract=4440349
[23]. Kanumarlapudi, P. K. "Improving Data Market Implementation Using Gray Relational Analysis in Decentralized Environments." *Journal of Artificial intelligence and Machine Learning* 2, no. 1 (2024): 1-7.
[24]. Raja, Chandrasekar, M. Ramachandran, Kurinjimalar Ramu, and Chinnasami Sivaji. "Solid Waste Management from a systems perspective using GRA methodology." Journal on Applied and Chemical Physics 2 (2023): 2. https://doi.org/10.46632/jacp/2/2/5
[25]. Chaki, Soumi, Yevgeniy Zagayevskiy, Xuebei Shi, Terry Wong, and Zainub Noor. "Machine learning for proxy modeling of dynamic reservoir systems: Deep neural network DNN and recurrent neural network RNN applications." In International Petroleum Technology Conference, p. D022S152R002. IPTC, 2020.
[26]. Sharma, Ramya, M. Ramachandran, Vimala Saravanan, and Chinnasami Sivaji. "Design optimization of Solar Power Inverter using the GRA Method." https://doi.org/10.46632/jame/2/3/3
[27]. Yazdinejad, Abbas, Hamed HaddadPajouh, Ali Dehghantanha, Reza M. Parizi, Gautam Srivastava, and Mu-Yen Chen. "Cryptocurrency malware hunting: A deep recurrent neural network approach." Applied Soft Computing 96 (2020): 106630.
[28]. Nayeemuddina Umme Salma, Green Fabrication of Nio Nano Particles Doped PS-PVDF Nanocomposite Films: Structural, Morphology and Electrical Studies, Journal of Information Systems Engineering and Management, 10(23), 2025, 794-800.
[29]. Jaganathan Rajamanickam, M. Ramachandran, Ramya sharma, Chinnasami Sivaji, "Distributed Generation (DG) system using COPRAS method", REST Journal on Advances in Mechanical Engineering, 2(3) September 2023, 11-22.
[30]. Kanumarlapudi, P. K. "Improving Data Market Implementation Using Gray Relational Analysis in Decentralized Environments." Journal of Artificial intelligence and Machine Learning 2, no. 1 (2024): 1-7.
[31]. Park, S. U., J. H. Park, Mohammed A. Al-Masni, Mugahed A. Al-Antari, Md Z. Uddin, and T-S. Kim. "A depth camera-based human activity recognition via deep learning recurrent neural network for health and social care services." Procedia Computer Science 100 (2016): 78-84.
[32]. Ghosh, Rajib. "A Recurrent Neural Network based deep learning model for offline signature verification and recognition system." Expert Systems with Applications 168 (2021): 114249.

[33]. Kuo, Yiyo, Taho Yang, and Guan-Wei Huang. "The use of grey relational analysis in solving multiple attribute decision-making problems." Computers & industrial engineering 55, no. 1 (2008): 80-93.

[34]. Tosun, Nihat. "Determination of optimum parameters for multi-performance characteristics in drilling by using grey relational analysis." The International Journal of Advanced Manufacturing Technology 28, no. 5 (2006): 450-455.

[35]. Tzeng, Chorng-Jyh, Yu-Hsin Lin, Yung-Kuang Yang, and Ming-Chang Jeng. "Optimization of turning operations with multiple performance characteristics using the Taguchi method and Grey relational analysis." Journal of materials processing technology 209, no. 6 (2009): 2753-2759.

[36]. Chan, Joseph WK, and Thomas KL Tong. "Multi-criteria material selections and end-of-life product strategy: Grey relational analysis approach." Materials & Design 28, no. 5 (2007): 1539-1546.

[37]. Hamzaçebi, Coşkun, and Mehmet Pekkaya. "Determining of stock investments with grey relational analysis." Expert Systems with Applications 38, no. 8 (2011): 9186-9195.

[38]. Zeng, Guangming, Ru Jiang, Guohe Huang, Min Xu, and Jianbing Li. "Optimization of wastewater treatment alternative selection by hierarchy grey relational analysis." Journal of environmental management 82, no. 2 (2007): 250-259.

[39]. Lin, C. L. "Use of the Taguchi method and grey relational analysis to optimize turning operations with multiple performance characteristics." Materials and manufacturing processes 19, no. 2 (2004): 209-220.

[40]. Lin, J. L., and C. L. Lin. "The use of the orthogonal array with grey relational analysis to optimize the electrical discharge machining process with multiple performance characteristics." International Journal of machine Tools and manufacture 42, no. 2 (2002): 237-244.

[41]. Wu, Hsin-Hung. "A comparative study of using grey relational analysis in multiple attribute decision making problems." Quality Engineering 15, no. 2 (2002): 209-217.

[42]. Kakulavaram, Sridhar. "Life Insurance Customer Prediction and Sustainbility Analysis Using Machine Learning Techniques." International Journal of Intelligent Systems and Applications in Engineering 10 (2022).

[43]. Çaydaş, Ulaş, and Ahmet Hasçalık. "Use of the grey relational analysis to determine optimum laser cutting parameters with multi-performance characteristics." Optics & laser technology 40, no. 7 (2008): 987-994.

[44]. Abhinav, E. Meher, Sai Naveen Kavuri, Thota Sandeep Kumar, Maragani Thirupathi, M. Chandra Mohan, and A. Suresh Reddy. "Analysis of molecular single-electron transistors using silicene, graphene and germanene." In *Proceedings of the Second International Conference on Computer and Communication Technologies: IC3T 2015, Volume 1*, pp. 77-84. New Delhi: Springer India, 2015.