Journal of Information Sciences and Application

Volume No. 13

Issue No. 2

May - August 2025

ENRICHED PUBLICATIONS PVT. LTD

S-9, IInd FLOOR, MLU POCKET,
MANISH ABHINAV PLAZA-II, ABOVE FEDERAL BANK,
PLOT NO-5, SECTOR-5, DWARKA, NEW DELHI, INDIA-110075,
PHONE: - + (91)-(11)-47026006

ISSN No. 2349-8412

Journal of Information Sciences and Application

Aims and Scope

Journal of Information Sciences and Applications has become very important with the ever-increasing demands of the software development to serve the millions of applications across various disciplines. For large software projects, innovative software development approaches are of vital importance. In order to gain higher software standard and efficiency, software process adaptation must be derived from social behavior, planning, strategy, intelligent computing, etc., based on various factors. International Journal of Software Engineering address the state of the art of all aspects of software engineering, highlighting the all tools and techniques for the software development process.

Journal of Information Sciences and Application

(Volume No. 13, Issue No. 2, May - August 2025)

Contents

Advanced Intrusion Detection in Software-Defined Networks through Ensemble Modeling

Ibrahim M. Elezmazy 1 and Walid Abdullah 1,*
1 Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Sharqiyah, Egypt.

A B S T RT A C

In software-defined networks (SDNs), effective intrusion detection is crucial for maintaining network safety and integrity. Traditional intrusion detection systems (IDS) have often failed to identify sophisticated threats due to their limited detection capabilities. To address this challenge, this study introduces an ensemble model that integrates Convolutional Neural Networks (CNN), Support Vector Machines (SVM), and XGBoost. This ensemble approach aims to enhance intrusion detection in SDNs by leveraging the strengths of each model. Using the InSDN dataset for training, our proposed model demonstrates superior performance and significantly outperforms a set of state-of-the-art models achieving a performance of 95% for accuracy, precision, recall, and F1 score exceeding the performance of other methods. Additionally, it significantly reduces false positive rates, highlighting its effectiveness in detecting complex intrusions in SDNs.

***Keywords**: IDS; Ensemble Learning; Network Security; Deep Learning; CNN; SVM; Anomaly Detection.*

1. Introduction

The rapid expansion of computer networks has greatly transformed the Internet and telecommunications sectors, leading to remarkable advancements in connectivity and efficiency [1]. Within this landscape, Software-Defined Networks (SDNs) have emerged as a groundbreaking innovation, offering unprecedented levels of flexibility, control, and centralization that exceed traditional network architectures. However, the rise of SDNs has also introduced new security challenges, making them vulnerable to various forms of unauthorized access and cyber threats. As a result, the development of robust and effective Intrusion Detection Systems (IDS) for SDNs has become a critical area of research [2].

IDS plays a vital role in detecting illicit activities, mitigating potential risks, and safeguarding the integrity and confidentiality of network information [3]. Traditional IDS approaches, which rely on predefined signatures and anomaly detection, have struggled to keep pace with the evolving landscape of cyber threats. The emergence of sophisticated cyber-attacks necessitates more advanced and adaptive solutions. Consequently, there is a pressing need for the integration of machine learning and deep learning techniques to enhance the effectiveness and resilience of IDS in the context of SDNs. This transition to more advanced methodologies aims to address the limitations of conventional systems and improve overall network security [4].

In recent years, the use of machine learning models for intrusion detection has significantly increased due to their ability to identify patterns in data and make accurate predictions [5]. Among the most popular machine learning techniques employed for this purpose are Convolutional Neural Networks (CNNs) [6], Support Vector Machines (SVMs) [7], and XGBoost [8], a prominent example of gradient boosting algorithms. Each of these models offers unique strengths: CNNs excel in feature extraction and pattern recognition, SVMs are highly effective at classifying data in high-dimensional spaces, and XGBoost leverages gradient boosting to construct powerful ensemble models. Thesediverse capabilities make them well-suited for the complex task of detecting intrusions in network systems.

Although these individual models have shown considerable success in detecting attacks, their performance can be further enhanced by combining them into ensemble models. Ensemble learning leverages the strengths of different models to improve overall accuracy, robustness, and generalizability. The objective of this study is to develop a hybrid methodology for intrusion detection in SDNs by integrating CNN, SVM, and XGBoost classifiers using a weighted soft voting algorithm. The proposed mode was compared against a set of state-of-arts models, the results of this approach demonstrate impressive performance, it achieved the best performance of 95% for accuracy, precision, recall, and F1 score exceeding the performance of other methods.

The rest of this paper is structured as follows: Section 2 provides the literal review and background needed for this study. Section 3 presents the methodology of this study. The proposed work is shown in Section 4. Section 5 presents experimental results. The conclusion and future directions are presented in Section 6.

2. Related Work

In this section, we provide a literature review on DL-based models for intrusion detection which are summarized in Table 1.

Elsayed et al. [9] proposed a hyper approach based on a Long Short Term Memory (LSTM) autoencoder and One-class Support Vector Machine (OC-SVM) to train the models exclusively with instances of typical classes to identify anomaly-based assaults in an imbalanced dataset. After being trained to recognize the typical traffic pattern and the input data's compressed representation, or latent features, the LSTM-autoencoder feeds the information to an OC-SVM algorithm. The disadvantages of the independent OC-SVM, such as its limited capacity to function in massive and high-dimensional datasets, are addressed by the hybrid model. Furthermore, they used the most recent Intrusion Detection System (IDS) dataset for SDN settings (InSDN) to conduct our tests.

Elsayed et al. [10] investigated the use of CNN for IDSs and proposed a technique to improve its performance by addressing the overfitting issue using two well-liked regularization strategies. The

method enhances IDSs' capacity to identify invisible intrusion events. They trained our approach and assessed its performance using the InSDN benchmark dataset. Based on the experimental results, it can be shown that regularization techniques can enhance CNN-based anomaly detection models' performance in an SDN environment.

Thakur et al. [11] proposed a model that classifies the incursions using a deep learning technique after extracting valuable features from the provided features. It should be emphasized that the underlying data points are not representative of the same distribution; rather, they are drawn from two distinct distributions, one of which is specific to the domain and the other general to all network intrusions. Considering this, they developed a novel Generic-particular autoencoder architecture, in which the autoencoder learns features relevant to that domain, while the generic autoencoder learns features common to all types of network intrusions.

Said et al. [12] created a hybrid Convolutional Neural Network (CNN) and bidirectional long short-term memory (BiLSTM) network to improve the use of binary and multiclass classification in network intrusion detection. Using the most widely used datasets (UNSW-NB15 and NSL-KDD), the efficacy of the suggested model was evaluated. We also made use of the InSDN dataset, which is devoted only to SDN.

Liu et al. [13] proposed an edge-intelligent intrusion detection system that can be applied when a WSN encounters a DoS attack. They implemented a parallel technique to increase the population variety in iterations and Lévy flight to enhance the AOA algorithm's capacity to leap out of the local optimum. Then, using the KNN machine learning classifier in conjunction with the enhanced PLAOA optimization process.

Balyan et al. [14] proposed HNIDS, A PSO technique enhances the vector. A multi-objective function is added to GA to increase its performance. It chooses the best features and produces better fitness outcomes to investigate the important characteristics, minimizing dimensions, increasing the true positive rate (TPR), and reducing the false positive rate (FPR). The following step involves an IRF that removes the less important features, adds a list of decision trees to each iterative process, monitors the classifier's performance, and guards against overfitting problems.

Another recent work [15], proposed an abnormal traffic detection system based on a hybrid deep learning model. The system uses a form of hierarchical detection. To achieve the fine detection of abnormal traffic from the surface, it first completes the rough detection of abnormal traffic in the network based on statistical data from switch ports. Next, it uses deep learning technology and wavelet transform to extract multi-dimensional features of all traffic data flowing through suspicious switches. The experimental findings demonstrate how fast the source of anomalous traffic may be found using the suggested port-information-based detection technique

Table 1. Summarization of some recent intrusion detection studies based on DL.

Ref.	Year	Dataset	Model	Evaluation (AUC.)
[9]	2020	InSDN	OC-SVM	87.5
			LSTM-Autoencoder-OC-SVM	90.5
[10]	2021	InSDN	Model-Based CNN Technique	93.01
[12]	2023	InSDN	CNN-LSTM	95.03
			LeNet5	92.09
[13]	2022	WSN-DS	KNN	91.16
			kNN_{PSO}	92.89
[14]	2022	NSL-KDD	HNIDS	88.149

3. Methodology

3.1 Convolutional Neural Network

CNN, also known as regularized feed-forward neural networks, uses filter (or kernel) optimization to automatically learn features. By applying regularized weights over fewer connections, backpropagation issues with disappearing gradients and expanding gradients which were observed in previous neural networks are avoided [6]

An input layer, hidden layers, and an output layer compose a CNN. One or more convolutionperforming layers are included in the hidden layers of a convolutional neural network. This usually consists of a layer that uses the layer's input matrix to perform a dot product of the convolution kernel. ReLU is typically used as the activation function for this product, which is typically the Frobenius inner product. The convolution procedure creates a feature map as the convolution kernel moves along the layer's input matrix; this feature map then feeds into the input of the layer after it. Other layers including pooling layers, completely connected layers, and normalization layers come after this. The degree to which a convolutional neural network resembles a matched filter should be highlighted here[16]. The CNN architecture is shown in Figure 1.

Figure 1. CNN model architecture.

3.2 Support-Vector Machin

The concept behind the support-vector network was previously implemented for the restricted case where the training data can be separated without errors. Here, we extend this result to nonseparable training data. The support-vector network is a new learning machine for two-group classification problems. The machine conceptually implements the following idea: input vectors are non-linearly mapped to a very high-dimension feature space. In this feature space, a linear decision surface is constructed. Special properties of the decision surface ensure the high generalization ability of the learning machine [17].

Linear SVM for Binary Classification is used for separating data into two classes using a linear decision boundary. Here's a detailed explanation. Is used to find a hyperplane that best separates two classes in a feature space. The hyperplane is chosen such that the margin between the closest points of the two classes (known as support vectors) is maximized. Given a set of training data $\{(x_i, y_i)\}$, where $x_i \in \mathbb{R}^n$ represents the feature vectors and $y_i \in \{-1, +1\}$ represents the class labels, the hyperplane can be represented as follows:

$$f(x) = w \cdot x + b = 0 \tag{1}$$

Here w is the weight vector and b is the bias. And the distance from a point x_i to the hyperplane can be calculated as follows:

$$\frac{|w \cdot x + b|}{\|w\|}$$

The SVM model for a binary classification problem with a linear separator is shown in Figure 2.

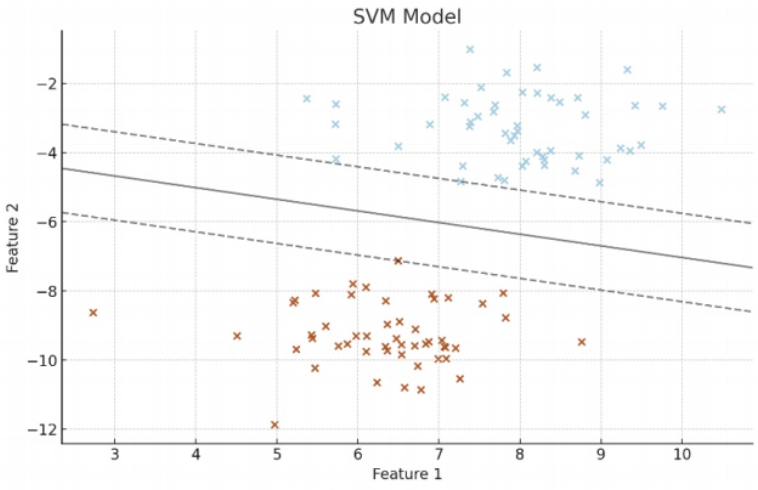

Figure 2. 2D plot illustrating the SVM model.

Where $\hat{y}^{(t)}$ is the prediction at iteration t, $f_t(x)$ is the prediction from the t tree, η is the learning rate, l is a differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i, Ω is a regularization term for the complexity of the model, γ is the penalty for adding a new tree, λ is the L2 regularization term on weights, T is the number of leaves in the tree, w_j is the weight on the j-th leaf.

XGBoost uses second-order Taylor approximation for the objective function to efficiently calculate the gain and cover for the decision tree splits. For each split, it calculates the gradients (g) and Hessians (h) as follows:

$$g_i = \frac{\partial l(\hat{y}_i, y_i)}{\partial(\hat{y}_i)} \tag{6}$$

$$h_i = \frac{\partial^2 l(\hat{y}_i, y_i)}{\partial(\hat{y}_i)^2} \tag{7}$$

Finally, the gain from a split is:

$$Gain = \frac{1}{2}\left[\frac{(\sum_{i\epsilon I_L} g_i)^2}{\sum_{i\epsilon I_L} h_i + \lambda} + \frac{(\sum_{i\epsilon I_R} g_i)^2}{\sum_{i\epsilon I_R} h_i + \lambda} - \frac{(\sum_{i\epsilon I} g_i)^2}{\sum_{i\epsilon I} h_i + \lambda}\right] - \gamma \tag{8}$$

where I_L and I_R are the instances in the left and right nodes after the split, respectively.

4. Proposed Work

To develop this problem detection method, we incorporate some machine learning approaches: CNN, SVM, and XGBoost. By fusing these models, we hope to use their different properties as well as enhance the net detection rate. The proposed model architecture is shown in Figure 3. The following is the description of each component of the proposed model.

4.1 Connected Neural Network (CNN)

CNNs can recognize patterns and features in data, especially useful for processing data sets or time series data in intrusion detection. the CNN part consists of four main layers including the Conv2D Layer which uses convolution operations to extract local features from input data, Batch Normalization to normalize the output of previous layers to stabilize and train faster, MaxPooling layer which is commonly utilized after the CNN layers to reduce the space size of the data, focusing on the most relevant elements, and finally dense layer: A fully connected layer that provides the final output of the CNN, it generates a probabilistic output pCNN(x) of the probability of an attack. The CNN model processes the input x through multiple layers to produce a probabilistic output, this operation is mathematically represented as follows:

on the most relevant elements, and finally dense layer: A fully connected layer that provides the final output of the CNN, it generates a probabilistic output pCNN(x) of the probability of an attack. The CNN model processes the input X through multiple layers to produce a probabilistic output, this operation is mathematically represented as follows:

$$f_{CNN}(X) = \sigma[w_d \cdot Flatten\left(Conv2D_n\left(..\left(Conv2D_1(X)\right)\right)\right)] \tag{10}$$

Where $Conv2D_i$ represents the i-th convolutional layer followed by Batch Normalization, max pooling, and Dropout, W_d represents the weights of the dense layer, and σ represents the sigmoid activation function.

4.2 Support Vector Machine (SVM)

SVM performs well in high-level environments and is used for its robustness and accuracy in classification tasks. One known problem with this model is that the values of the variables must be specified precisely. To overcome this problem, the Grid Search method is used for Hyperparameter tuning to find the optimal values for the model's Hyperparameter, which increases the performance of SVM. The SVM model maps the input x to a binary classification output as defined in Eq. (11):

$$f_{SVM}(X) = sign(w \cdot X + b) \tag{11}$$

Where w represents the weights learned by the SVM and b represents the bias term.

4.3 XGBoost

XGBoost is a robust gradient-enhancing algorithm that excels in processing structured data and can capture complex patterns through multiple decision trees. XGBoost model uses the Trees Growmethod to aggregate predictions from multiple weak learners (decision trees) and outputs the weighted sum of predictions from multiple decision trees which helps to improve overall accuracy. The mathematical equation of the Trees Grow method is defined as follows:

The mathematical equation of the Trees Grow method is defined as follows:

$$f_{XGB}(X) = \sum_{K=1}^{K} \alpha_k h_k(X) \tag{12}$$

Where α_k represents the weight of the k-th tree and h_k represents the k-th decision tree.

4.4 Ensemble Voting Classifier

The final stage of our approach involves a weighted voting process to aggregate the outputs from the three models: CNN, SVM, and XGBoost. Initially, each model generates a probability estimate indicating the likelihood of an intrusion. Following this, a weighted soft voting mechanism is employed to determine the final prediction. In this process, the individual probability estimates from each model are combined into a single probability score. This is achieved through a weighted sum, where each model's contribution is adjusted according to its assigned weight. The ensemble model thus integrates the probabilistic outputs from all constituent models using this weighted soft voting strategy. Mathematically, this can be represented as follows:

$$p_{ensemble}(X) = \alpha_{CNN} \cdot p_{CNN}(X) + \alpha_{SVM} \cdot p_{SVM}(X) + \alpha_{XGB} \cdot p_{XGB}(X) \tag{13}$$

Where: p_{CNN} , p_{SVM}, p_{XGB} are the predicted probabilities from CNN, SVM, and XGBoost respectively.

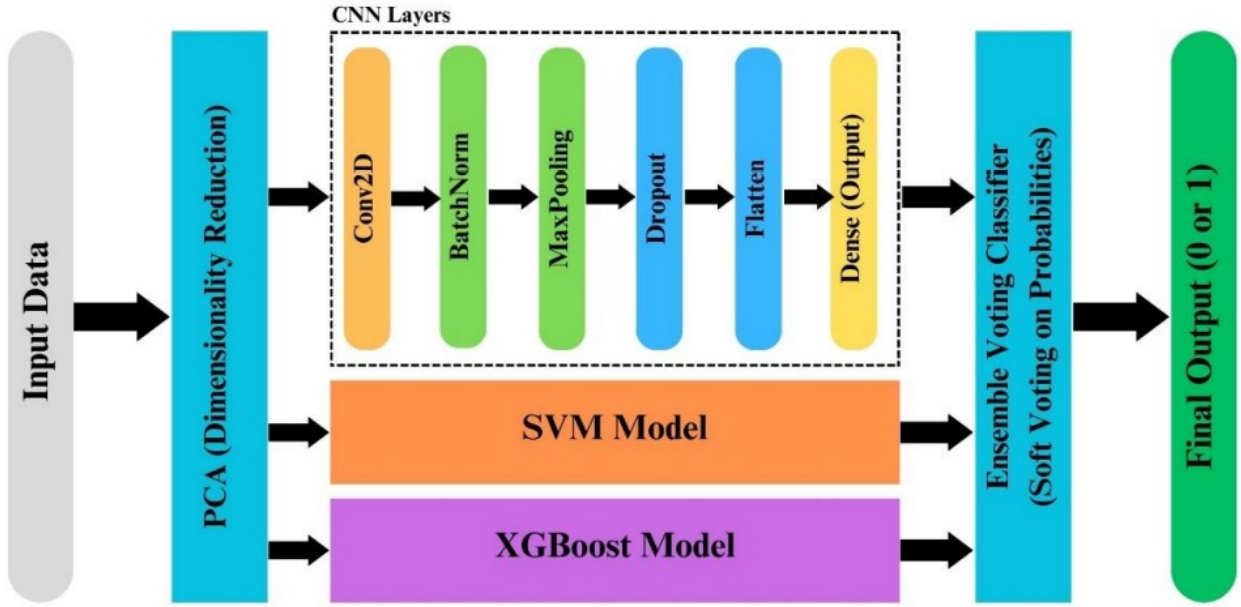

Figure 3. The proposed model architecture.

The proposed model was compiled to determine the loss function, Adam optimizer, and metrics for evaluating performance. The Categorical cross-entropy loss function is used to optimize the initial weights of the proposed model to increase classification accuracy, in addition, it was trained for 50epochs. In addition, in our experiments, and applied a mini-batch gradient descent technique todecrease the error calculated from the loss function (Binary Cross Entropy). In each epoch, the data is divided into 64 batches so that the weights in each batch are updated. This means that in every epoch, the weights change 64 times, corresponding to the number of batches.

5. Results and Discussion

This section provides a detailed description foe the utilized dataset, evaluation metrics used to assess the proposed model, the key finding, and experimental results for this study.

5.1 Utilized Dataset

In this work, we utilized A public and widely used InSDN dataset designed for intrusion detection research [19]. This dataset contains various objects representing network traffic, which can be used to detect malicious activity. The dataset contains one label that denotes either normalcy or an attack for each of the 30 features that make up its 10000 samples. This set is meant to enhance unique characteristics, and lower repetition hence making it quality for machine learning evaluation. The dataset description is summarized in Table 2.

Table 2. InSDN dataset description.

Number of records	**10000**
Number of features	30
Number of classes	2

5.2 Evaluation Metrics

Our proposed work was evaluated using a set of comprehensive set of evaluation metricsincluding accuracy, precision, recall, and F1-score.

Accuracy (ACC) measures the proportion of correctly classified instances among all instances:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

where:

- TP = True Positives (correctly identified attacks).
- TN = True Negatives (correctly identified normal instances).
- FP = False Positives (normal instances incorrectly classified as attacks).
- FN = False Negatives (attacks incorrectly classified as normal instances).

- **Precision** measures the proportion of true positive predictions among all positive predictions.

$$Precision = \frac{TP}{TP + FP} \quad (15)$$

- **Recall (Sensitivity or True Positive Rate)**: measures the proportion of true positive instances that are correctly identified.

$$Recall = \frac{TP}{TP + FN} \quad (16)$$

- **F1-score**: **this metric** provides a balanced measure between Recall and Precision

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (17)$$

Together, these metrics examine how well the model can classify network traffic into common and attack categories. Higher contrasts indicate stronger performance in terms of identification while reducing false positives and false negatives. For the Proposed model, these metrics provide quantitative insight into the efficiency and reliability of the IDS.

5.3 Implementation Settings

Python programming language was utilized to develop our deep learning models. The development and experimentation were conducted on the Google Colab platform [20], leveraging its computational

resources. In addition, models were built using the Keras API [21]. Table 3 describes the models' configurations and hyperparameters used in training DL models.

Table 3. Implementation settings.

Optimizer	Adam
Epochs	50
Batch size	64
Dropout	0.3

5.4 Statistical Analysis

In this section, we present the comprehensive results of our investigation of six deep learning models: Convolutional Neural Networks (CNNs), Deep Neural Networks (DNN), Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), Deep Belief Networks (DBN), Gated Recurrent Units (GRU), and Long Short-Term Memory networks (LSTM). We evaluate and compare these models based on a set of key performance metrics, namely accuracy, precision, recall, and F1-score. Table 4. Offers significant insights into the classification model's performance, demonstrating its accuracy correctly.

Table 4. Performance of different DL models for intrusion detection.

Model	Accuracy	Precision	Recall	F1-score
ANN	0.92	0.92	0.92	0.92
DNN	0.92	0.90	0.93	0.91
LSTM	0.91	0.91	0.90	0.91
RNN	0.90	0.90	0.90	0.90
DBN	0.90	0.91	0.92	0.91
CNN	0.91	0.91	0.90	0.91
GRU	0.91	0.91	0.90	0.90
Proposed model	0.95	0.95	0.95	0.95

The comparison between the various deep learning models against our proposed model showed that the proposed model consistently outperforms the others across all evaluation metrics. With accuracy, precision, recall, and F1-score for the proposed model are all 0.95, highlighting its balanced and robust performance. In contrast, the other models show slight variations and low performance compared to the proposed model. These results indicate that while the traditional models exhibit competitive performance, particularly ANN and DNN, our proposed model's superior efficiency proves its potential for more accurate and reliable predictions in practical applications.

6. Conclusion and Future Work

This work proposed an enhanced deep learning model for complicated threat detection in software-defined networks (SDNs). This model combines three powerful models named CNN, SVM, and XGBoost classifiers. These models were combined with a weighted soft voting system for the best results. However, before any training could take place various activities and steps were followed during data processing. This included standardization and transformation into a 2D image dummy for the sake of CNN and fully connected or dense layers, batch normalization, and Conv2D layers in step 1&2 with max pooling. Max pooling was then applied on these convolutions after which there was a merge layer between everything else until output nodes where ReLU followed by softmax activation would come into play. In addition, the GridSearch method was utilized to adjust SVM model hyperparameters optimally to increase the model's performance, the ensemble model showed great efficiency in terms of accuracy, precision, recall, F1 score, and AUC, thus being perceived as a stable and well-performing model for identifying intrusions. Improvements in the future will not only enhance the efficiency of the model but also promote its extendibility and trustworthiness in real-world network security environments.

Declarations

Ethics Approval and Consent to Participate

The results/data/figures in this manuscript have not been published elsewhere, nor are they under consideration by another publisher. All the material is owned by the authors, and/or no permissions are required.

Consent for Publication

This article does not contain any studies with human participants or animals performed by any of the authors.

Availability of Data and Materials

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Competing Interests

The authors declare no competing interests in the research.

Funding

This research was not supported by any funding agency or institute.

Author Contribution

All authors contributed equally to this research.

Acknowledgment

The author is grateful to the editorial and reviewers, as well as the correspondent author, who offered

assistance in the form of advice, assessment, and checking during the study period.

References

[1] Evans, P.C. and M. Annunziata, Industrial internet: Pushing the boundaries. General Electric Reports, 2012: p. 488-508.

[2] Abdulganiyu, O.H., T. Ait Tchakoucht, and Y.K. Saheed, A systematic literature review for network intrusion detection system (IDS). International journal of information security, 2023. 22(5): p. 1125-1162.

[3] Ashoor, A.S. and S. Gore, Importance of intrusion detection system (IDS). International Journal of Scientific and Engineering Research, 2011. 2(1): p. 1-4.

[4] Saranya, T., et al., Performance analysis of machine learning algorithms in intrusion detection system: A review. Procedia Computer Science, 2020. 171: p. 1251-1260.

[5] Ahmad, Z., et al., Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 2021. 32(1): p. e4150.

[6] Vinayakumar, R., K. Soman, and P. Poornachandran. Applying convolutional neural network for network intrusion detection. in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). 2017. IEEE.

[7] Mohammadi, M., et al., A comprehensive survey and taxonomy of the SVM-based intrusion detection systems. Journal of Network and Computer Applications, 2021. 178: p. 102983.

[8] Bhati, B.S., et al., An improved ensemble based intrusion detection technique using XGBoost. Transactions on emerging telecommunications technologies, 2021. 32(6): p. e4076.

[9] Elsayed, M.S., et al., Network Anomaly Detection Using LSTM Based Autoencoder, in Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks. 2020, Association for Computing Machinery: Alicante, Spain. p. 37–45.

[10] Elsayed, M.S., et al. The Role of CNN for Intrusion Detection Systems: An Improved CNN Learning Approach for SDNs. 2021. Cham: Springer International Publishing.

[11] Thakur, S., et al., Intrusion detection in cyber-physical systems using a generic and domain specific deep autoencoder model. Computers & Electrical Engineering, 2021. 91: p. 107044.

[12] Said, R.B., Z. Sabir, and I. Askerzade, CNN-BiLSTM: A Hybrid Deep Learning Approach for Network Intrusion Detection System in Software-Defined Networking With Hybrid Feature Selection. IEEE Access, 2023. 11: p. 138732-138747.

[13] Liu, G., et al., An Enhanced Intrusion Detection Model Based on Improved kNN in WSNs. Sensors, 2022. 22(4): p. 1407.

[14] Balyan, A.K., et al., A Hybrid Intrusion Detection Model Using EGA-PSO and Improved Random Forest Method. Sensors, 2022. 22(16): p. 5986.

[15] Wang, K., et al., Abnormal traffic detection system in SDN based on deep learning hybrid models. Computer Communications, 2024. 216: p. 183-194.

[16] Desai, R. and T. Venkatesh. Robust Network Intrusion Detection Systems for Outlier Detection. in 2022 IEEE

27th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD). 2022. IEEE.

[17] Cortes, C. and V. Vapnik, Support-vector networks. Machine learning, 1995. 20: p. 273-297.

[18] Chen, T. and C. Guestrin. Xgboost: A scalable tree boosting system. in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.

[19] Yan, M., et al., Bearing remaining useful life prediction using support vector machine and hybrid degradation tracking model. ISA transactions, 2020. 98: p. 471-482.

[20] Sukhdeve, D.S.R. and S.S. Sukhdeve, Google Colaboratory, in Google Cloud Platform for Data Science: A Crash Course on Big Data, Machine Learning, and Data Analytics Services. 2023, Springer. p. 11-34.

[21] Gulli, A. and S. Pal, Deep learning with Keras. 2017: Packt Publishing Ltd.

Sleep Apnea Detection

Mohamed Elkholy 1 , Maryam Abdelfattah 2 , Maryam Amer 2 and Hussam Elbehiery 3,*

Faculty of Computer Science & Engineering, Alamein International University, Egypt;

ABSTRACT

Sleep apnea is one of the most common sleep disorders, posing a significant health risk. The proposed paper provides a prospective technique by applying machine learning to enable the early prediction of sleep apnea. Two methodologies were used for this study: Firstly, the hybrid model was used in analyzing the electrocardiogram records by combining VGG16 and Long Short-Term Memory (LSTM) networks for the extraction of meaningful features from the ECG signals for modeling the process of aiding detection. Second, lifestyle patterns were assessed in their relationship with the disorder. Many lifestyle factors were analyzed in search of critical indicators that may, at an early age, indicate the onset of the case of sleep apnea. The fusion of deep learning methods and analysis of lifestyle patterns offers a comprehensive framework for the solution to the complexity of sleep apnea detection. More exactly, this kind of interdisciplinary approach should permit improvement in effectiveness and accuracy for early diagnosis, therefore enabling timely intervention and treatment of that effect. Proposed here is an innovative methodology that could lead to better management and recovery of the affected.

Keywords: *Artificial Intelligence; Electrocardiogram (ECG); Sleep Diseases; Deep learning.*

1. Introduction

One-third of human life is spent in the sleeping process. Hence, sleep has a huge impact on maintaining health and overall well-being [1, 2]. The sleep cycle includes two primary stages. Firstly, there is rapid eye movement, which is characterized by high speed in eye movement and includes dreams. Secondly, there is the non-rapid eye movement stage in sleep [3]. REM sleep includes an increased peak activity in the sympathetic nervous system. Increased hemodynamic changes are also a part of REM. On the other hand, NREM sleep reduces oxygen consumption, significantly decreases heart rate, and causes a decrement in blood pressure [5, 6].

Sleep disorders are increasingly becoming common, with approximately seventy million adults within the United States affected. Such disorder in the sleeping process leads to higher rates of mortality. Among the various sleep disorders are insomnia, central disorders of hypersomnolence, parasomnias, and movement disorders to mention a few. Sleeping Breath Disorder (SBD) is also considered one of the popular sleeping disorders. There are several types of SBDs: obstructive sleep apnea, sleep hypoventilation, and central sleep apnea. OSA affects one billion individuals all over the world.

Advanced technologies have made it feasible to develop a few techniques for sleep apnea detection. Most of the techniques use ECG signals in the detection of certain abnormal features while one is

sleeping. Initially, conventional machine-learning techniques were used, but there has been a shift to advanced Deep-Learning models since they are good at feature extraction. However, this complication makes it very hard to compare the performance of such algorithms due to differences in data sets, physiological signals used, training, and evaluation criteria.

2. Background

Accurate and efficient detection of sleep apnea is critical to treat and reduce its impact. Several approaches are used in monitoring sleep apnea, which include questionnaires, medical imaging, and signal-processing techniques [16]. As much as questionnaires are cost-effective in determining patients with sleep disorders, medical imaging could be efficient in serious conditions by monitoring several activities of anatomies while sleeping. Currently, the gold standard for the diagnosis of sleep apnea is PSG, since it involves multiple biological signals. The method of PSG, however, tends to be restricted to special sleep laboratories and thus is not appropriate for home application.

There is a rising interest in wearable, non-invasive technologies that autonomously can monitor and manage sleep [18, 19]. Several signals connected with physiological activities have been discussed for the detection of sleep apnea, and among these, ECG has emerged as one of the potential candidates in wearable devices. In this respect, machine learning and deep learning techniques are used for the analysis of different human signals. Machine learning depends on manually defined features; on the other hand, deep learning methods do an auto feature extraction.

Deep learning, due to recent developments, has transformed the detection technique of sleep apnea. Many authors have reported high performance in this area using Convolutional Neural Networks (CNNs) [20, 21]. Other studies have assessed the performance of various deep learning architectures and feature extraction methods to increase accuracy in detection

Some innovative technologies in motion detection methods and single-lead ECG wearable devices have been proposed for real-time sleep apnea monitoring and detection. The healthcare system of the future is likely to include these kinds of wearable devices, which provide the capability of automatic sleep condition monitoring and management.

3. Experimental Setup

3.1 Dataset

Two different datasets were used in the proposed experiments:

3.1.1 ECG Records

For ECG records, a dataset containing seventy records is used. Such data is divided into a learning set, containing thirty-five records, while the rest are used for the testing set. The records were chosen in

different lengths for up to 10 hours. Each record has only a single continuous ECG signal. In addition, there is an annotation about sleep apnea that is based on specialized human decisions. Every record is supplemented with a collection of automated annotations to distinguish between normal and up-normal signals. A collection of eight records is supplemented with additional signals associated with abdominal respiratory signal airflow rate and oxygen saturation.

3.1.2 Tabular Data

In the work, the Sleep Health and Lifestyle Dataset has been used, with four hundred entries that denote sleep health. The thirteen columns represent sleeping behavior and daily habits; it also contains personal data, such as age, gender, amount of activity and stress levels one undergoes, the total number of sleeping hours, heart rate, and blood pressure—features deemed important in predicting sleeping apnea.

Dataset Features:

Sleeping metrics tell about the quality of sleep and sleeping duration, among other factors classifying sleep patterns. Personal data analyzes physical activities and the stress levels of different people. Cardiovascular health is in charge of measuring changes in blood pressure and heart rate. Sleep disorder measurements are used to detect the occurrence of sleep disorders.

3.2 Preprocessing

Two different datasets were used in the proposed experiments:

3.2.1 ECG Records

The ECG signals were divided into 1-minute intervals after which R-R Intervals were extracted using the Hamilton R-peak algorithm to detect normality. Chen et al used a median filter in reducing physiologically meaningful points. The subsequent R-R Intervals underwent essential processing before inputting into the machine learning algorithms.

The level of R-peaks was recorded near the R-R Interims and passed towards a profound learning algorithm. In arrange to test both R-peak value as well as R-R interims at a rise to evaluate a cubic introduction at 3 Hz was utilized. Hence, the added R-R Interims and R-peak value are bolstered into profound learning algorithms. To acquire the exact transient highlights of DRNN data while using it, the data was divided into n segments of 60/n seconds.

3.2.2 Tabular Data

This one had an amazingly simple preprocessing step where we cleaned up inconsistent data, replaced missing data, and encoded categorical data to make life easier for the model.

4. Conventional Machine Learning

Traditional machine learning methods include the following three steps. As depicted in Figure1, the proposed steps in this work are classifying a set of selected features. Hence, the proposed work uses feature engineering methodologies to enhance algorithm performance. In this work, we presented a unified feature engineering framework, specially designed for conventional machinelearning algorithms. Preprocessing was first applied to ECG signals. Then time, frequency, and nonlinear features were extracted. Dimensionality was reduced by PCA, followed by feature extraction. Afterward, the classification of ECG signals into apnea and normal episodes was realized.

4.1 Steps of Extracting Features

In detecting apnea by ECG signals, it has been proven to be HRV parameter dependent. The proposed work presents various machine learning models that efficiently detect sleep apnea by making use of different preprocessed data sets. Extracted features include time domain, frequency domain, and HRV features.

4.1.1 Time Domain

The statistical, time-domain features are divided into two main categories. The first is the longterm time domain while the second is the short-term time domain. Both features are extracted from R-R intervals. The extracted features include the minimum value, range value, and standard deviation value.

4.1.2 Frequency-Domain Features

Frequency-domain features provide information related to power frequency, for instance, total power, low-frequency power, and high frequency, and their ratio.

4.1.3 Nonlinear Features

Seven nonlinear features, that were SD1, SD2, the ratio of SD2/SD1, CVI, CSI, modified CVI, and permutation entropy, were extracted.

4.2 Dimension Reduction

We applied PCA and performed dimensionality reduction to get rid of the curse of dimensionality, classifying only important features.

4.3 Classification

Following this, several machine-learning techniques were applied to the detection and classification of sleep apnea. One major challenge that remains is the lack of a common and fair comparison framework between these many techniques. The proposed study will, therefore, address this gap by performing a

comparative evaluation of some prominent machine learning methods.

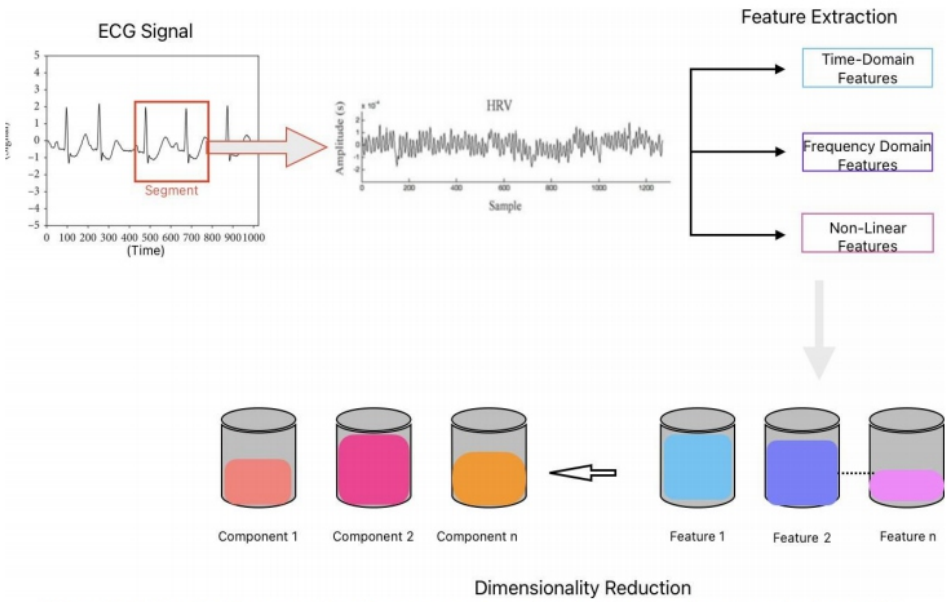

Figure 1. A Flowchart representing the process of using ML in detecting sleep disorder from ECG.

1) Linear Discriminant Analysis: LDA offers the most basic method of classification but very essential in creating linear decision boundaries by fitting Gaussian density per class, ruled by Bayes' rule. The major hyperparameters for LDA entail a solver model and parameters associated with tolerance; it sets the threshold of significance of singular Values.

2) Quadratic Discriminant Analysis: QDA through quadratic decision bound- emanates from the generalization of LDA. Ordinarily, using the SVD solver, the main parameter for QDA is the statement of tolerance, which is essential for efficient training.

3) Logistic Regression: LR includes the solver algorithm and choices for convergence tolerance.4) Gaussian Naïve Bayes (GNB): GNB is an application of Bayes' theorem assuming independence between features. The key hyperparameter for GNB would be that of variance tuning for better stability, which is sensitive to the ratio of the biggest variance across different features.

5) Gaussian Process (GP): It is a non-parametric approach. GP uses kernel functions to identify class labels. Important hyperparameters for GP include the choice of kernel and algorithm for optimization.

6) Support Vector Machine: SVM uses kernel functions to project the data into higherdimensional spaces. The key hyperparameters are about the type of kernel to be used and their associated parameters

7) K-Nearest Neighbors: KNN will enable the classification of the data by neighbors; several distance metrics are at one's disposal.

8) Decision Tree: DT is another nonparametric classifier that predicts classes by learning simple decisions based on input features.Extract rules from features. The most important hyperparameters include the maximum depth of the tree, the strategy to perform the splits, and the criterion to evaluate the split quality.

9) Random Forest RF — this method generalizes DT by averaging several trees on sub-sampling with replacement. The main hyperparameters are maximum depth, the lowest count of samples needed to split an internal node, the criterion by which the quality of the split is measured, and the count of trees, or estimators.

10) Extra Trees: ET is a variant of RF that uses the whole original sample (rather than bootstrapping it), and it also randomly selects the cut points. This makes ET a great deal more computationally efficient than RF.

11) AdaBoost (AB): AB is another ensemble technique that combines multiple weak learners to come up with a strong classifier. Contrary to RF's bagging technique, AB works on boosting, where the weights of the misclassified samples are increased in every iteration. The most important hyperparameters are the count of estimators and the rate of learning.

12) Gradient Boosting, GB: Generalizes boosting as an optimization problem. In this process of sequentially adding weak learners—mostly DTs, GB minimizes model loss by using a gradient descent-like technique. Some major hyperparameters are the loss function, highest value of depth, count of estimators, sample fraction for base learner fitting, and split criterion.

13) Majority Voting (MV): Being itself an ensemble method, this process enforces the combination of classifiers by a vote. In this way, MV pools predictions from different conventional machine-learning algorithms to bolster classification performance.

5. Deep Learning

We have used the VGG16 architecture in its hybrid version in our ECG model. In the proposed model, multiple convolutional layers are combined with pooling operations as follows:

In the first stage, two convolutional layers are using 8*8 filters with size of 3. After that a maxpooling layer with two size windows.

At the second stage, 128 extractor filter is used followed by the same max-pooling layer as the first stage.

The third stage involves the application of two convolutional layers, by increasing the number of filters to 265.

The fourth stage consists of two convolutional layers and the number of filters is also duplicated to be 512.

After that, the output from these convolutional and pooling layers was fed into a one-layer stacked Deep Recurrent Neural Network with Long Short-Term Memory cells. Each LSTM cell processed a 2-D input of size 512 × 5 512×5. The DRNN produced an output with dimensions 128 × 2 128×2.The algorithms used in the proposed work are listed below.

Algorithm 1 Main function

```
MAIN ()
    specific path /to file location
    LOAD data from path (data folder path)
        Process data from the rr interval
folder and rp folder.
    if ¬rp data ∨ ¬rr_intervals data
then
        return
    end if
    A, B ← ...      % properties
    B←tf. Keras. utils. categorical (Y, num_
classes = 2)
    kfold ← StratifiedKFold (n_
splits = 5, shuffle = True,
random state = 7)
for data set (training set) i n kfold. split (A, B. argmax (1))
do
        m ← create new_model ()
        TRAIN_MODEL (m, A [train], B [train])
        EVALUATE_MODEL (model, A [test], B [test])
end for
```

Algorithm 2 Load and preprocess data

```
    Function LOAD data (data folder_path)

    rpeaks data ← {}
    rr_intervals data ← {}
for filename found in os. listdir do
        file_path ←
            os. join path.
        ...
end for
    return r_peaks_list, rr_intervals_list
```

Algorithm 3 Create a deep learning model.

```
Function CREATE_new model ()
    model ← Sequential ()
    ...
    return model
```

Algorithm 4 Train and evaluate the model.

```
Function TRAIN_MODEL (model, A, B)
    ...
Function EVALUATE_MODEL (model, A, B)
    ...
```

A. Tabular Data

Algorithm 5 Random Forest Classifier

```
Input: Data A, target B
Output: Feature categories
Random Forest initialization
Data fitting
Get feature categories
Create a Data Frame to store feature categories
Sort the data frame by categories
Print top features
```

Algorithm 6 DBSCAN Clustering

```
Input: Data A
Output: Clusters
Extract features and target
Perform DBSCAN clustering
Visualize clusters
```

Algorithm 7 Regression Models

```
Input: Data A, target B
Output: Model MSEs
Split the data into training and testing sets
Initialize regression models (Linear, Ridge, Lasso, Polynomial)
Fit models to training data
Predict on testing data
Calculate MSE for each model
```

5.1 Results Comparison and Evaluation

Since difficulties were used in getting results from the ECG model, the expected results are compared rather than the real ones. The experimental results are listed in Table 1.

Table 1. Proposed experimental results.

	metrics	ECG model Tabular data	
1	Accuracy	88.02%	93%
2	Recall	83.34%	91%
3	F-score	84.24%	92%

6. Conclusion

Unfortunately, in the case of the ECG model, problems extracting R-peaks made it hard to feed the model with our data since it would not derive actual results, hence the failure of our attempt in trying to detect sleep apnea using ECG records. After several trials, it was concluded that the R-peaks files were corrupted and could not be processed. Off these tabular data, processing of this data can lead us to the conclusion that high-stress jobs and high blood pressure are two of the biggest risk factors for sleep apnea. Overall, we couldn't derive real comparison results since we couldn't even get results from the ECG model, while the tabular model results are not indicative of better performance compared to the expected ECG, since we had two different kinds of data, which makes it hard to give any objective conclusion.

Declarations

Ethics Approval and Consent to Participate

The results/data/figures in this manuscript have not been published elsewhere, nor are they under consideration by another publisher. All the material is owned by the authors, and/or no permissions are required.

Consent for Publication

This article does not contain any studies with human participants or animals performed by any of the authors.

Availability of Data and Materials

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Competing Interests

The authors declare no competing interests in the research.

Funding

This research was not supported by any funding agency or institute.

Author Contribution

All authors contributed equally to this research.

Acknowledgment

The author is grateful to the editorial and reviewers, as well as the correspondent author, who offered assistance in the form of advice, assessment, and checking during the study period.

References

[1] M. Bahrami and M. Forouzanfar. Detection of sleep apnea from single-lead ECG: Comparison of deep learning algorithms. IEEE Int. Symp. Med. Meas. Appl., pages 1–5, Jun 2021.

[2] Mahsa Bahrami and Mohamad Forouzanfar. Sleep Apnea Detection from Single-Lead ECG: A Comprehensive Analysis of Machine Learning and Deep Learning Algorithms. IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 71, 2022.

[3] Amaral L. Glass L. Hausdorff-J. Ivanov P. C. Mark R. ... Stanley H. E. Goldberger, A. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online], 101, 2000.

[4] Herbert F Jelinek. A typical ECG signal showing the RR interval. ResearchGate.

[5] RG Mark AL Goldberger JH Peter T Penzel, GB moody. The Apnea ECG Database. Computers In Cardiology, 27:255–258, 2000.

[6] LAKSIKA THARMALINGAM. Sleep Health and Lifestyle Dataset. Kaggle.

[7] Sung-Hyun Yang Zubaer Md. Abdullah Al, Keshav Thapa. Improving R Peak Detection in ECG Signal Using Dynamic Mode Selected Energy and Adaptive Window Sizing Algorithm with Decision Tree Algorithm. MDPI.

Ensemble RF-KNN Model for Accurate Prediction of Drought Levels

Walid Abdullah 1,* Nebojsa Bacanin 2 and K Venkatachalam 3

ABSTRACT

The increasing frequency and severity of droughts represent a critical threat to agricultural systems worldwide, disrupting food production, and supply chains. Accurate and timely prediction of drought conditions is essential for ensuring agricultural sustainability and enabling proactive mitigation strategies. This study proposes a novel ensemble model that combines Random Forest (RF) and K-Nearest Neighbors (KNN) using soft voting to predict drought conditions based on meteorological data. The dataset consists of drought classifications for six levels, ranging from no drought to five drought severity levels using meteorological indicators from various U.S. counties. The performance of the proposed model was evaluated against several state-of-the-art machine learning models, including Logistic Regression, Decision Tree, and Artificial Neural Networks, using various evaluation metrics including accuracy, precision, recall, and F1-score. The results demonstrate the effectiveness of the proposed ensemble approach, achieving superior accuracy and reliability in predicting drought severity. This research highlights the transformative potential of machine learning in supporting agricultural systems and addressing climate change challenges through data-driven drought monitoring and mitigation strategies.

Keywords: *Drought Prediction; Machine Learning; Ensemble Models; Random Forest; K-Nearest*

1. Introduction

Drought is one of the most significant natural disasters affecting agriculture, with widespread impacts on crop production, water availability, and food security [1, 2]. The increasing occurrence of droughts, driven by climate change, has put immense pressure on agricultural systems, making it critical to predict and mitigate their effects [3]. When crops fail due to water scarcity, it disrupts local food supplies, global trade, and livelihoods. Thus, timely and accurate prediction of drought conditions is vital for reducing risks and enabling farmers and policymakers to prepare effectively [4]. Traditional methods for monitoring drought, such as the Standardized Precipitation Index (SPI) or the Palmer Drought Severity Index (PDSI), rely heavily on region-specific climatic data. While these indices have been effective in some areas, their limitations include the inability to generalize to diverse geographic regions and their limited ability to capture complex relationships among droughtrelated factors [5]. This gap necessitates the development of advanced tools that can process vast datasets and identify intricate patterns to predict drought with greater accuracy.

The advancements in artificial intelligence (AI) and machine learning (ML) provide a promising avenue to address these challenges. It enables machines to simulate human intelligence, learn from data, adapt to new inputs, and perform tasks that traditionally require human expertise [6]. Machine learning is a subset of AI, that focuses on developing algorithms that allow computers to identify patterns and make decisions without explicit programming. These models excel in handling large, complex datasets, making them ideal for analyzing meteorological indicators and predicting drought [7].

Machine learning models, such as Random Forest (RF), K-Nearest Neighbors (KNN), and Artificial Neural Networks (ANN), are particularly powerful due to their ability to process high dimensional data, uncover nonlinear relationships, and adapt to diverse data distributions. They can leverage historical meteorological data to predict future drought conditions, even in the presence of missing or noisy data [8]. Furthermore, the integration of advanced techniques like ensemble learning enhances their accuracy and robustness, as it combines the strengths of multiple models to mitigate individual weaknesses [9]. Unlike traditional approaches, which often rely on predefined equations and region-specific assumptions, ML models can generalize well across different regions and climatic conditions. This adaptability makes them highly scalable, offering solutions that can be applied to diverse agricultural systems worldwide [10, 11].

This paper presents a data-driven framework to predict drought conditions, emphasizing its critical importance for agriculture. A new ensemble model, which combines Random Forest and KNearest Neighbors using soft voting is proposed to improve predictive performance. This model utilized the meteorological data across six drought classification levels to ensure high accuracy and adaptability. To validate its effectiveness, the ensemble model is compared against state-of-the-art machine learning algorithms, including Logistic Regression, Decision Tree, and Artificial Neural Networks. The goal of this study is to provide a practical and robust approach to drought prediction that supports agricultural decision-making and enhances resilience to climate-related risks. By utilizing meteorological data and advanced ML techniques, this research contributes to the development of global drought monitoring systems that prioritize food security and sustainable agricultural practices.

The remainder of this paper is structured as follows: Section 2 reviews the literature and discusses related work. Section 3 outlines the methodology, including the proposed ensemble model and its components. Section 4 presents experimental analysis, covering the dataset, preprocessing methods, experimental setup, and evaluation metrics. Section 5 discusses the results, comparing the performance of the proposed model with other machine learning approaches. Finally, Section 6 concludes the paper with key insights and recommendations for future research.

2. Related Work

The prediction of drought conditions is a critical aspect of environmental management, agricultural planning, and disaster preparedness. Over the years, various machine learning techniques have been employed to predict droughts with varying degrees of success. In this section, we review the existing literature on drought prediction using machine learning models, highlighting the strengths and limitations, and better understanding the effectiveness of the existing methods.

The use of machine learning for drought prediction in Pakistan was explored in [12], where Support Vector Machine (SVM), ANNs, and KNN models were applied to predict drought severity levels

(moderate, severe, and extreme) during two major cropping seasons. Data from the NCEP/NCAR reanalysis database was used, and Recursive Feature Elimination (RFE) enhanced predictor accuracy. SVM outperformed other models by effectively capturing temporal and spatial drought patterns, identifying key predictors such as relative humidity, temperature, and wind speed. Short-term drought forecasting has also been explored. In Ethiopia's Awash River [13], the authors focused on short-term drought forecasting using the Standardized Precipitation Index (SPI). They compared ANNs, support vector regression (SVR), and coupled wavelet-ANN models. The coupled wavelet-ANN model delivered the most accurate SPI 3 and SPI 6 predictions over 1- and 3-month lead times, demonstrating the benefits of integrating wavelet transforms with ANN for enhanced forecasting.

In [14], researchers investigated groundwater levels in drought-prone areas of northwestern Bangladesh using historical data from 1981 to 2017. Seven machine learning models, including Random Tree (RT) and Random Forest (RF), were evaluated with metrics such as RMSE and correlation coefficient (CC). Ensemble methods like Bagging-RT and Bagging-RF achieved the most accurate predictions, showcasing their potential for sustainable groundwater resource management. Another study [15] applied machine learning models, including Random Forest (RF), Extreme Gradient Boost (XGB), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM), to estimate drought events on the Tibetan Plateau. Using the Standardized Precipitation Evapotranspiration Index (SPEI), XGB and RF excelled at SPEI-3 estimation, while LSTM and XGB performed best for SPEI-6. These findings highlight the adaptability of these models for decisionmaking in water resource management.

The authors in [8] examined drought indices (SPI and SPEI) at multiple timescales, employing Random Forest, Voting Regressor, AdaBoost Regressor, and K-Nearest Neighbors Regressor. Random Forest and Voting Regressor achieved high accuracy, with NSE values ranging from 0.74 to 0.93, while KNN showed weaker performance. This study underscores the need for advanced algorithms and improved data collection for precise drought prediction. In addition, Agricultural drought vulnerability in Bangladesh's Barind Tract was analyzed in [16], utilizing Landsat satellite imagery and multiple indices like NDVI and VHI. The Cellular Automata-Artificial Neural Network (CA-ANN) model forecasted significant increases in extreme drought conditions by 2026 and 2031, driven by reduced vegetation and rising temperatures. The study emphasizes the need for proactive measures to enhance agricultural resilience.

Hydrological drought prediction was addressed in [17] by modeling three drought indices (SPI, SSI, and SPEI) using SVR, Gene Expression Programming (GEP), and M5 model trees (MT). The MT model excelled in SSI predictions with a correlation coefficient (CC) of 0.8195 and RMSE of 0.8186, demonstrating its effectiveness for hydrological drought modeling. The effectiveness of hybrid approaches, such as wavelet-boosting ANN (WBS-ANN) and wavelet-boosting SVR (WBS-SVR), has also been demonstrated in [18], the authors combine wavelet transforms with ensemble for drought

prediction in Ethiopia. The results showed that hybrid models like wavelet-boosting ANN (WBS-ANN) and wavelet-boosting SVR (WBS-SVR) provided the most accurate SPI predictions, highlighting the potential of hybrid approaches in enhancing drought forecasting.

These studies highlight the rapid advancements in applying machine learning models to drought prediction across diverse regions and contexts. While individual models such as SVM, ANN, and RF have shown strong performance, ensemble techniques, and hybrid models provide further improvements by leveraging the strengths of multiple approaches. This body of research opens the way for developing novel ensemble models, such as the proposed RF-KNN approach, to enhance accuracy and scalability in drought prediction tailored to agricultural systems.

3. Methodology

This section presents the methodological framework employed in this study to predict drought using machine learning. Initially, five machine learning models were implemented and evaluated, including Random Forest (RF), Decision Tree (DT), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Artificial Neural Networks (ANN). Subsequently, the two best-performing models, RF and KNN, were selected to develop an ensemble model using a soft voting mechanism to enhance prediction accuracy. The methodology is divided into two subsections: the first provides an overview of the individual ML models used, and the second elaborates on the proposed ensemble model.

3.1 Machine Learning Models

To identify the most effective algorithms for drought prediction, five machine learning models were employed. Each model offered unique strengths that addressed specific aspects of the prediction task. The first model is the decision tree (DT) [19], this algorithm served as a starting point, creating a tree-like structure by splitting the dataset into subsets based on feature values. While its simplicity and interpretability are notable advantages, DT models are prone to overfitting, especially without proper pruning. To overcome these limitations, the Random Forest (RF) model, an advanced ensemble learning technique, was employed [20]. RF generates multiple decision trees during training and combines their predictions, either by averaging (for regression) or by voting (for classification). Configured with 50 estimators and a max depth of 80, RF demonstrated robustness in handling non-linear relationships and reducing overfitting, making it a strong contender for drought prediction.

Next, the K-Nearest Neighbors (KNN) algorithm which, is a non-parametric technique also tested in this study [21]. KNN classifies data points based on the majority class of their k-nearest neighbors, with k=5 chosen for this study. Its simplicity and effectiveness in capturing local patterns were advantageous, although the model's performance is sensitive to the choice of k and the distance metric used. The Logistic Regression (LR) model, commonly employed for binary classification, was adapted for this

study to handle multi-class drought prediction. As a statistical model estimating theprobability of categorical outcomes based on input features, LR provided a solid baseline for comparison with more complex approaches.

Lastly, the Artificial Neural Network (ANN) model, inspired by biological neural structures, was evaluated [22]. The ANN configuration consisted of input, hidden, and output layers. Known for their ability to model complex, non-linear patterns, ANNs require careful parameter tuning but proved valuable in exploring the dataset's intricate relationships. This exploration of machine learning techniques provided a foundation for selecting the two most effective models, RF and KNN. Their complementary strengths were subsequently combined in a novel ensemble model to enhance predictive accuracy.

3.2 Proposed Ensemble Model

Based on the evaluation of individual models, RF and KNN emerged as the top-performing algorithms. To capitalize on their complementary strengths, an ensemble model was developed using a Voting Classifier with soft voting. The ensemble approach integrates the predictive capabilities of RF and KNN, resulting in a more accurate and generalized model. The concept of soft voting involves averaging the predicted probabilities of individual models, giving more weight to confident predictions, as opposed to hard voting, which relies on majority decisions. This allows the ensemble model to make predictions that reflect the confidence levels of its constituent algorithms. The mathematical formulation of the soft voting mechanism is given by:

$$P(y_k) = \frac{1}{n} \sum_{i=1}^{n} P_i(y_k) \quad (1)$$

where $P_i(y_k)$ is the predicted probability of class y_k By the *i-th* model, and n is the total number of models. This approach ensures a more balanced prediction when dealing with imbalanced datasets. Fig 1 illustrates the architecture of the proposed Voting Classifier, showing how the outputs of Random Forest and K-Nearest Neighbors are combined to produce the final prediction.

This ensemble model combines RF's strength in capturing global patterns and relationships with KNN's ability to detect local data structures. By integrating these strengths, the ensemble model demonstrated superior predictive performance compared to its components. This approach underscores the potential of ensemble learning to enhance the accuracy and reliability of machine learning models in drought prediction.

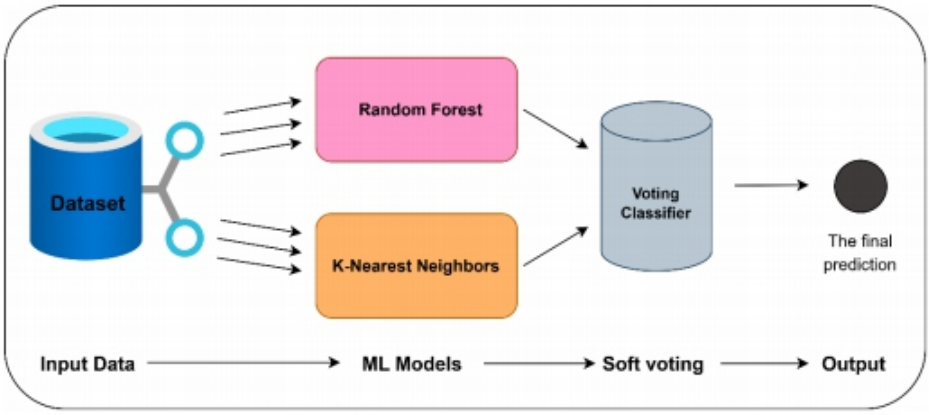

Figure 1. The architecture of the proposed RF-KNN ensemble model.

4. Experimental Analysis

4.1 Dataset

The dataset used in this study is derived from the U.S. Drought Monitor and incorporates meteorological data provided by the NASA POWER Project and the U.S. Drought Monitor [23]. It is a classification dataset aimed at predicting six levels of drought severity, ranging from "No Drought" (None) to "Exceptional Drought" (D4). It includes 18 meteorological indicators, such as precipitation, temperature, humidity, and wind speed, which are essential for capturing the conditions leading to drought. Each data entry represents the drought level at a specific point in time for a given U.S. County, the drought severity levels are classified as shown in Table 1. The dataset serves as a valuable resource for investigating the feasibility of predicting drought conditions using meteorological data. Its insights could potentially lead to generalized drought prediction models applicable beyond the U.S

Table 1. Drought severity levels in the dataset

Category	Description	Possible impacts
(None)	No Drought	- Short-term dryness slows planting, growth of crops or pastures
(D0)	Abnormally Dry	- Some lingering water deficits - Pastures or crops not fully recovered
(D1)	Moderate Drought	- Some damage to crops, pastures - Streams, reservoirs, or wells low, some water shortages developing or imminent
(D2)	Severe Drought	- Crop or pasture losses likely - Water shortages are common - Water restrictions imposed
(D3)	Extreme Drought	- Major crop/pasture losses - Widespread water shortages or restrictions
(D4)	Exceptional Drought	- Exceptional and widespread crop/pasture losses - Shortages of water in reservoirs, streams, and wells create water emergencies.

4.2 Experimental Setup

The meteorological indicator values in the dataset were normalized using standard scaling to ensure that all features had a similar scale, improving the performance of distance-based algorithms like K-Nearest

Neighbors [24]. Then the dataset is divided into training and testing subsets, with 80% of the data used for training and 20% reserved for testing. The experiments were conducted using Python (version 3.10.13) with the Scikit-learn library [25]. Each machine-learning model was initially trained and evaluated using default hyperparameters. Subsequently, fine-tuning was performed to optimize their performance. For the ensemble model, Scikit-learn's Voting Classifier was employed to integrate the predictions of the top-performing models. All experiments were executed on the Kaggle platform, leveraging an Nvidia Tesla P100 GPU with 30 GB of RAM, ensuring efficient computation for both training and testing.

4.3 Evaluation Metrics

To assess the performance of the machine learning models and the proposed ensemble approach, four evaluation metrics were employed: accuracy, precision, recall, and F1-score. These metrics provide a comprehensive understanding of the models' capabilities, particularly in addressing the challenges posed by imbalanced datasets.

Accuracy measures the proportion of correctly classified instances to the total number of instances in the dataset. It is calculated as:

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{FP} + \text{TN} + \text{FN})} \tag{2}$$

Precision evaluates the correctness of positive predictions by determining the ratio of true positive predictions to the total number of positive predictions. It is expressed as:

$$\text{Precision} = \frac{TP}{(TP + FP)} \tag{3}$$

Recall, also known as sensitivity or true positive rate, measures the ability of the model to correctly identify all positive instances. It is defined as:

$$\text{Recall} = \frac{TP}{(TP + FN)} \tag{4}$$

F1-Score is the harmonic means of precision and recall, providing a balanced evaluation of the model's performance, especially when dealing with imbalanced data. It is calculated as:

$$F1\ \text{Score} = 2 \times \frac{\text{recall} \times \text{Precision}}{\text{recall} + \text{Precision}} \tag{5}$$

While accuracy offers an overall performance indicator, it can be misleading in cases where one class dominates the dataset. So, we need the other metrics. In the context of Precision, high precision indicates that the model has a low false-positive rate, making it suitable for tasks where false alarms carry significant costs. High recall indicates that the model minimizes false negatives, making it valuable in scenarios where missing positive cases is critical. Finally, A higher F1 score reflects a better trade-off between precision and recall. These metrics collectively offer a nuanced evaluation of the models, highlighting their strengths and limitations in drought prediction tasks.

5. Results and Discussion

This section presents and analyzes the performance of the machine learning models employed for drought prediction, as well as the proposed ensemble model. The experiments aimed to evaluate the models across key metrics including accuracy, precision, recall, and F1-score while highlighting the strengths and limitations of each algorithm. The proposed model, combining Random Forest and K-Nearest Neighbors, was expected to outperform individual models due to its ability to leverage the complementary strengths of its components. The results are summarized in Table 2, which provides a comparative overview of the models' performance.

Table 2. Performance metrics of machine learning models for drought prediction.

Model	Accuracy	Precision	Recall	F1-Score
Decision tree (DT)	0.7754	0.7743	0.7754	0.7749
Random Forest (RF)	0.8126	0.8083	0.8126	0.8104
K-Nearest Neighbors (KNN)	0.7987	0.7983	0.7987	0.7985
Logistic Regression (LR)	0.6579	0.5545	0.6579	0.6024
Artificial Neural Networks (ANN)	0.7180	0.7245	0.6921	0.7079
Proposed model (RF- KNN)	**0.8252**	**0.8155**	**0.8221**	**0.8188**

The performance of the machine learning models varied significantly, reflecting their strengths and limitations in handling the drought prediction task. The Decision Tree (DT), with an accuracy of 77.54%, offered simplicity and interpretability but struggled with overfitting, which Random Forest (RF) addressed effectively. RF outperformed DT, achieving 81.26% accuracy due to its ensemble approach that better handles non-linear relationships and reduces overfitting. Similarly, K-Nearest Neighbors (KNN) demonstrated its ability to capture local patterns with an accuracy of 79.87%, though its reliance on parameter tuning, such as the choice of k and distance metrics, may limit its robustness in higher dimensions.

Logistic Regression (LR) exhibited the weakest performance, with an accuracy of 65.79%, due to its inability to model complex, non-linear patterns in the data. In contrast, Artificial Neural Networks (ANN) showed moderate success, achieving 71.80% accuracy, highlighting its potential to handle non-linear relationships but revealing the need for further optimization to improve its performance. The proposed ensemble model, which combines RF and KNN, delivered the highest accuracy of 82.52% and the best F1-score of 0.8188, demonstrating its effectiveness in leveraging the complementary strengths of its components to provide a balanced and robust solution for drought prediction. Figure 2 provides a visual comparison of the accuracy of all models, highlighting the superior performance of the proposed ensemble model.

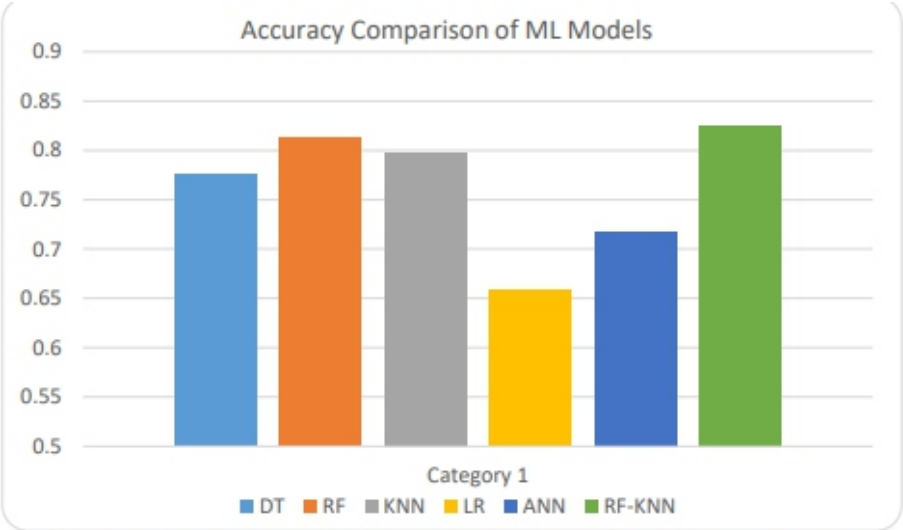

Figure 2. Comparison of performance metrics across machine learning models.

The proposed model's performance highlights the potential of ensemble learning to improve the accuracy of drought prediction. By integrating two complementary algorithms, the ensemble model provides a more generalized and robust solution. This underscores the importance of leveraging diverse machine-learning techniques for complex environmental problems.

6. Conclusion and Future Work

This study explored the use of machine learning models to predict drought severity levels using meteorological data. Five individual algorithms including Decision Tree, Random Forest, K-Nearest Neighbors, Logistic Regression, and Artificial Neural Networks were initially evaluated to identify their suitability for this task. Among these, Random Forest and K-Nearest Neighbors emerged as the top-performing models, demonstrating robust predictive capabilities and complementing each other in terms of capturing both global and local data patterns. To enhance accuracy, an ensemble model was developed by integrating Random Forest and K-Nearest Neighbors through a soft voting mechanism. The ensemble approach effectively leveraged the strengths of its constituent models, achieving superior performance compared to individual models. The experiments, conducted using a standardized dataset normalized for consistency, demonstrated the reliability of the proposed methodology. The evaluation metrics, including accuracy, precision, recall, and F1-score, provided a comprehensive analysis of the model's predictive abilities and highlighted the advantages of ensemble learning in addressing the complexities of drought prediction. This research contributes to the growing field of data-driven drought prediction, emphasizing the potential of machine learning in mitigating the impacts of drought through early detection. However, the study primarily relied on historical meteorological data, and further work is needed to incorporate additional variables such as soil moisture and vegetation indices. Future research should also explore the generalizability of the proposed ensemble model to other geographical regions and its adaptability to real-time prediction systems.

Declarations

Ethics Approval and Consent to Participate

The results/data/figures in this manuscript have not been published elsewhere, nor are they under consideration by another publisher. All the material is owned by the authors, and/or no permissions are required.

Consent for Publication

This article does not contain any studies with human participants or animals performed by any of the authors.

Availability of Data and Materials

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Competing Interests

The authors declare no competing interests in the research.

Funding

This research was not supported by any funding agency or institute.

Author Contribution

All authors contributed equally to this research.

Acknowledgment

The author is grateful to the editorial and reviewers, as well as the correspondent author, who offered assistance in the form of advice, assessment, and checking during the study period.

References

[1] Wilhite, D.A. (2016). Drought as a natural hazard: concepts and definitions Droughts (pp. 3-18): Routledge.

[2] McWilliam, J. (1986). The national and international importance of drought and salinity effects on agricultural production. Functional plant biology, 13(1), 1-13.

[3] Qiu, J., Z. Shen, and H. Xie. (2023). Drought impacts on hydrology and water quality under climate change. Science of The Total Environment, 858, 159854.

[4] Balti, H., et al. (2020). A review of drought monitoring with big data: Issues, methods, challenges and

research directions. Ecological Informatics, 60, 101136.

[5] Bazrafshan, J., S. Hejabi, and J. Rahimi. (2014). Drought monitoring using the multivariate standardized precipitation index (MSPI). Water resources management, 28, 1045-1060.

[6] Orosz, T., et al. (2021). Evaluating human versus machine learning performance in a legaltech problem. Applied Sciences, 12(1), 297.

[7] Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and

Research (IJSR).[Internet], 9(1), 381-386.

[8] En-Nagre, K., et al. (2024). Assessment and prediction of meteorological drought using machine learning algorithms and climate data. Climate Risk Management, 45, 100630.

[9] Prodhan, F.A., et al. (2022). Projection of future drought and its impact on simulated crop yield over South Asia using ensemble machine learning approach. Science of The Total Environment, 807, 151029.

[10] Potla, R.T. (2022). Scalable Machine Learning Algorithms for Big Data Analytics: Challenges and Opportunities. Journal of Artificial Intelligence Research, 2(2), 124-141.

[11] Cravero, A., et al. (2022). Challenges to use machine learning in agricultural big data: a systematic literature review. Agronomy, 12(3), 748.

[12] Khan, N., et al. (2020). Prediction of droughts over Pakistan using machine learning algorithms . Advances in Water Resources, 139, 103562.

[13] Belayneh, A. and J. Adamowski. (2013). Drought forecasting using new machine learning methods. Journal of Water and Land Development, 18(9), 3-12.

[14] Pham, Q.B., et al. (2022). Groundwater level prediction using machine learning algorithms in a droughtprone area. Neural Computing and Applications, 34(13), 10751-10773.

[15] Mokhtar, A., et al. (2021). Estimation of SPEI meteorological drought using machine learning algorithms. IEEe Access, 9, 65503-65523.

[16] Kafy, A.-A., et al. (2023). Assessment and prediction of index based agricultural drought vulnerability using machine learning algorithms. Science of The Total Environment, 867, 161394.

[17] Shamshirband, S., et al. (2020). Predicting standardized streamflow index for hydrological drought using machine learning models. Engineering Applications of Computational Fluid Mechanics, 14(1), 339-350.

[18] Belayneh, A., et al. (2016). Coupling machine learning methods with wavelet transforms and the bootstrap and boosting ensemble approaches for drought prediction. Atmospheric research, 172, 37-47.

[19] Suthaharan, S. and S. Suthaharan. (2016). Decision tree learning. Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning, 237-269.

[20] Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

[21] Cover, T. and P. Hart. (1967). Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27.

[22] Zou, J., Y. Han, and S.-S. So. (2009). Overview of artificial neural networks. Artificial neural networks: methods and applications, 14-22.

[23] Monitor, U.D. (Accessed 2024). U.S. Drought Monitor. https://droughtmonitor.unl.edu/.

[24] Raju, V.G., et al. (2020). Study the influence of normalization/transformation process on the accuracy of supervised classification. Paper presented at the 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT).

[25] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.

Interval graphs and proper interval graphs in Fuzzy and Neutrosophic Graphs

Takaaki Fujita1 and Florentin Smarandache2
Independent Researcher, Shinjuku, Shinjuku-ku, Tokyo, Japan; t171d603@gunma-u.ac.jp.
University of New Mexico, Gallup Campus, NM 87301, USA; smarand@unm.edu

ABSTRACT

: Interval graphs represent vertices as intervals on the real line, with edges denoting overlapping intervals, while proper interval graphs prevent one interval from being fully contained within another. This paper explores interval and proper interval graphs within the frameworks of fuzzy, neutrosophic, and Turiyam Neutrosophic graphs. We examine how these types of graphs can represent relationships involving uncertainty and imprecision, focusing on their properties and relationships.

Keywords: Neutrosophic Graph; Interval Graphs; Proper Interval Graphs; Fuzzy Graph; Intersection Graphs.

1 Introduction

1.1 Interval graphs and proper interval graphs

Graph theory is a fundamental area of mathematics that examines networks made up of nodes (vertices) and connections (edges), essential for analyzing paths, structures, and properties of these networks [28]. A notable example in graph theory is the intersection graph, which represents sets where vertices correspond to these sets, and edges exist between vertices if their corresponding sets intersect [82, 53, 100]. Variants such as intersection digraphs[23, 121], random intersection graphs[56, 122, 88] , Intersection hypergraphs [87, 72, 7], and geometric intersection graphs[31, 58] have also been studied. And several related graph classes have been extensively studied, including interval graphs [57, 46], proper interval graphs [60, 65, 16], Mixed interval hypergraphs[12], interval hypergraphs[94, 22, 85, 54], almost interval graphs [13], weighted interval graphs [9, 17, 118], semi-proper interval graphs [97], mixed interval graphs[61, 63, 62], Unit mixed interval graphs[99], Rigid interval graphs[77], Minimum proper interval graphs[60], circular arc graphs [52, 114, 59], and polygon-circle graphs [73].

In this paper, we focus on interval and proper interval graphs. Interval and proper interval graphs are well known types of intersection graphs. In interval graphs, each vertex corresponds to an interval on the real line, with edges between vertices whose intervals overlap [57, 46]. Proper interval graphs, a subclass of interval graphs, ensure that no interval is fully contained within another, thereby avoiding nested intervals [60, 65, 16]. A graph is an interval graph if and only if it is chordal and AT-free [74]. Additionally, key graph parameters such as the interval number, pathwidth [70, 27], and boxicity [15] are associated with interval graphs. Interval graphs have various applications, including in food webs [21, 78, 18], scheduling problems [67, 51, 50], and DNA analysis [68, 84, 116, 83].

1.2 Fuzzy Graphs and Neutrosophic Graphs

A fuzzy graph assigns a membership value between 0 and 1 to each vertex and edge, representing the degree of uncertainty or imprecision [95, 39, 89]. Essentially, fuzzy graphs are a graphical representation of fuzzy sets [76, 120, 119]. They are widely used in fields like social networks, decision-making, and transportation systems,

where relationships are uncertain or not clearly defined [95, 86]. Neutrosophic graphs [1, 33, 44], based on neutrosophic set theory [3, 101, 111], extend classical and fuzzy logic by incorporating three components: truth, indeterminacy, and falsity, providing greater flexibility in handling uncertainty.

Building on these ideas, Turiyam Neutrosophic graphs were introduced as an extension of neutrosophic and fuzzy graphs, where each vertex and edge is assigned four attributes: truth, indeterminacy, falsity, and a liberal state [47, 48]. Plithogenic graphs have also emerged as a more generalized form, and are actively being researched [112, 38, 43, 36, 103, 110].

While significant progress has been made in studying fuzzy and neutrosophic graphs, including their intersection variants (e.g., fuzzy intersection graphs [98, 81, 20, 55] and neutrosophic intersection graphs [11]), there has been limited exploration of interval graphs and proper interval graphs within the context of fuzzy, neutrosophic, and Turiyam Neutrosophic graphs.

1.3 Our Contribution

Based on the above, this paper defines interval graphs and proper interval graphs within the context of fuzzy, neutrosophic, and Turiyam Neutrosophic graphs, and examines their properties as well as the relationships between these graph classes.

2 Preliminaries and definitions

In this section, we present a brief overview of the definitions and notations used throughout this paper.

2.1 Basic Graph Concepts

Here are a few basic graph concepts listed below. For more foundational graph concepts and notations,

please refer to [28].

Definition 1 (Graph). [28] A graph is G a mathematical structure consisting of a set of vertices $V(G)$ and a set of edges $E(G)$ that connect pairs of vertices, representing relationships or connections between them. Formally, a graph is defined as $G=(V, E)$, where V is the vertex set and E is the edge set.

Definition 2 (Degree). [28] Let $G=(V, E)$ be a graph. The degree of a vertex $v \in V$, denoted deg (v), is the number of edges incident to v. Formally, for undirected graphs:

$$\deg(v) = |\{e \in E \mid v \in e\}|$$

In the case of directed graphs, the in-degree $\deg^{-}(v)$ is the number of edges directed into v_I, and the out-degree $\deg^{*}(v)$ is the number of edges directed out of v.

Definition 3 (Subgraph). [28] A subgraph of G is a graph formed by selecting a subset of vertices and edges from G.

Definition 4 (Connected graph). (cf.[115, 64]) A graph $G = (V, E)$ is said to be a **connected graph** if for any two distinct vertices $u, v \in V$, there exists a path in G that connects u and v. In other words, every pair of vertices in the graph is reachable from each other, meaning there is a sequence of edges that allows traversal between any two vertices.

Mathematically, for all $u, v \in V$, there exists a sequence of vertices $v_1 = u, v_2, \ldots, v_k = v$ such that $(v_i, v_{i+1}) \in E$ for all $1 \le i < k$.

Definition 5 (Induced subgraph). [75, 66] Let $G = (V, E)$ be a graph, where V is the set of vertices and E is the set of edges. For a subset $V' \subseteq V$, the *induced subgraph* $G[V']$ is the graph whose vertex set is V' and whose edge set consists of all edges from E that have both endpoints in V'. Formally, the induced subgraph $G[V'] = (V', E')$ is defined as follows:

$$E' = \{(u, v) \in E \mid u \in V', v \in V'\}.$$

In other words, $G[V']$ is the subgraph of Gthat contains all vertices in V' and all edges from G' whose endpointsare both in V'

'

Definition 6 (Complete Graph). (cf.[29, 8]) A complete graph is a graph $G=(V, E)$ in which every pair of distinct vertices is connected by a unique edge. Formally, a graph $G=(V, E)$ is complete if for every pair of vertices $u, v \in V$ with $u \neq v$, there exists an edge $\{u, v\} \in E$.

The complete graph on n vertices is denoted by K_n, and it has the following properties:

- The number of vertices is $|V| = n$
- The number of edges is $|E| = \binom{n}{2} = \frac{n(n-1)}{2}$.
- Each vertex has degree $\deg(v) = n - 1$ for all $v \in V$.

2.2 Intersection graph and Interval graphs

In this paper, we focus on Interval graphs, which are known as intersection graphs. Intersection graphs have been extensively studied[82, 53, 100]. The definition is provided below[82, 53, 100].

Definition 7 (Intersection graph)**.** [82, 53, 100] A *intersection graph* is a graph that represents the intersection relationships between sets. Formally, let $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ be a collection of sets. The *intersection graph* $G = (V, E)$ associated with $\mathcal{S}$ is a graph where:

- The vertex set V corresponds to the sets in $\mathcal{S}$, i.e., $V = \{v_1, v_2, \ldots, v_n\}$, where each vertex v_i represents the set $S_i \in \mathcal{S}$.
- There is an edge $(v_i, v_j) \in E$ if and only if the corresponding sets S_i and S_j have a non-empty intersection, i.e., $S_i \cap S_j \neq \emptyset$.

Next, we will consider interval graphs and proper interval graphs. The definitions are provided below[57, 46].

Definition 8. [57, 46] An *interval graph* is an undirected graph $G = (V, E)$ that can be represented by a family of intervals on the real line. For each vertex $v \in V$, there exists a corresponding interval I_v on the real line. Two vertices $u, v \in V$ are adjacent, i.e., $(u, v) \in E$, if and only if their corresponding intervals I_u and I_v overlap. Formally, the edge set E of the graph G is defined as:

$$E(G) = \{(u, v) \mid I_u \cap I_v \neq \emptyset\}.$$

Definition 9. [60, 65, 16] A *proper interval graph* is a special case of an interval graph where no interval is strictly contained within another. That is, for any two intervals I_u and I_v corresponding to vertices u and v, neither $I_u \subset I_v$ nor $I_v \subset I_u$. This restriction ensures that no interval is nested within another. A graph is a proper interval graph if and only if it has a proper interval representation.

The above graphs have been the subject of numerous published papers and studies [80, 96, 61]. An interval graph is both chordal (i.e., every induced cycle has length 3 [32]) and AT-free (i.e., it contains no asteroidal triple, a set of three vertices such that any two are connected by a path that avoids the neighborhood of the third vertex[19]). Below are examples of interval graphs and proper interval graphs.

Example 10 (Interval Graph)**.** Consider a set of intervals on the real line:

$$I_1 = [1, 5], \quad I_2 = [4, 8], \quad I_3 = [6, 9], \quad I_4 = [2, 3]$$

The vertices $V = \{v_1, v_2, v_3, v_4\}$ correspond to the intervals I_1, I_2, I_3, I_4, and the edges are drawn between vertices whose intervals overlap:

- $I_1 \cap I_2 \neq \emptyset$, so $(v_1, v_2) \in E$,
- $I_2 \cap I_3 \neq \emptyset$, so $(v_2, v_3) \in E$,
- $I_1 \cap I_4 \neq \emptyset$, so $(v_1, v_4) \in E$,
- $I_1 \cap I_3 = \emptyset$, so $(v_1, v_3) \notin E$,
- $I_3 \cap I_4 = \emptyset$, so $(v_3, v_4) \notin E$,
- $I_2 \cap I_4 = \emptyset$, so $(v_2, v_4) \notin E$.

The resulting interval graph $G = (V, E)$ is:

$$V = \{v_1, v_2, v_3, v_4\}, \quad E = \{(v_1, v_2), (v_1, v_4), (v_2, v_3)\}$$

Example 11 (Proper Interval Graph)**.** Consider a set of intervals that do not nest:

$$I_1 = [1, 3], \quad I_2 = [4, 6], \quad I_3 = [7, 9], \quad I_4 = [2, 4]$$

None of these intervals are strictly contained within another. The vertices $V = \{v_1, v_2, v_3, v_4\}$ correspond to the intervals, and the edges are drawn between overlapping intervals:

- $I_1 \cap I_4 \neq \emptyset$, so $(v_1, v_4) \in E$,
- $I_2 \cap I_4 \neq \emptyset$, so $(v_2, v_4) \in E$,
- $I_3 \cap I_2 = \emptyset$, so $(v_2, v_3) \notin E$,
- No other intervals overlap.

The proper interval graph G is:

$$V = \{v_1, v_2, v_3, v_4\}, \quad E = \{(v_1, v_4), (v_2, v_4)\}$$

Since no interval is strictly contained within another, this forms a proper interval graph.

Let p and q be integers. The generalized concepts of p-Proper Interval Graphs [91, 24] and q-Improper Interval Graphs[10] are well known. Like standard interval graphs, they have been the subject of various studies. The definitions of these graphs are provided below.

Definition 12 (p-Proper Interval Graph)**.** [91] A graph $G = (V, E)$ is called a *p-proper interval graph* if there exists an interval representation of G such that no interval in the representation is properly contained within more than p other intervals. Formally, let $\mathcal{I} = \{I_v \mid v \in V\}$ be a set of intervals corresponding to the vertices of G. The graph G is p-proper if for every interval $I_u \in \mathcal{I}$, the number of other intervals $I_v \in \mathcal{I}$ such that $I_v \subset I_u$ is at most p:

$$\forall u \in V, \quad |\{v \in V \mid I_v \subset I_u\}| \leq p.$$

Definition 13 (q-Improper Interval Graph)**.** [10] A graph $G = (V, E)$ is called a *q-improper interval graph* if there exists an interval representation of G such that no interval in the representation properly contains more than q other intervals. Formally, let $\mathcal{I} = \{I_v \mid v \in V\}$ be a set of intervals corresponding to the vertices of G. The graph G is q-improper if for every interval $I_u \in \mathcal{I}$, the number of other intervals $I_v \in \mathcal{I}$ such that $I_u \subset I_v$ is at most q:

$$\forall u \in V, \quad |\{v \in V \mid I_u \subset I_v\}| \leq q.$$

p-Proper Interval Graph generalizes the concept of proper interval graphs, where a 0-proper interval graph is a proper interval graph (i.e., no interval is properly contained within any other interval). q-Improper Interval Graph extends the notion of proper interval graphs, where a 0-improper interval graph is a proper interval graph (i.e., no interval properly contains any other interval).

2.3 Interval graph in Fuzzy Graphs

Now, we explore interval graphs within the context of fuzzy graphs. Fuzzy graphs extend classical graph theory by incorporating the principles of fuzzy sets [119, 14, 117, 25]. Extensive research has been conducted on fuzzy graphs [95]. The definition of a fuzzy graph is given below.

Definition 14. [95] A *fuzzy graph* $\psi = (V, \sigma, \mu)$ is defined as follows:

- V is a set of vertices.
- $\sigma : V \to [0, 1]$ is a function that assigns a membership degree to each vertex $v \in V$, indicating the degree of membership of v in the fuzzy graph.
- $\mu : V \times V \to [0, 1]$ is a fuzzy relation that represents the strength of the connection between each pair of vertices $(u, v) \in V \times V$, such that $\mu(u, v) \leq \min\{\sigma(u), \sigma(v)\}$.

In this definition, the following properties hold:

- The fuzzy function μ is symmetric, meaning $\mu(u, v) = \mu(v, u)$ for all $u, v \in V$.
- Additionally, $\mu(v, v) = 0$ for all $v \in V$, meaning that there is no self-loop in the fuzzy graph.

The fuzzy graph ψ allows for the representation of uncertainty in the presence or strength of connections between vertices, making it a valuable tool for modeling complex systems with ambiguous or imprecise relationships.

Next, we define the fuzzy interval graph, which combines the concepts of a fuzzy graph and an interval graph, as follows.

Definition 15 (Fuzzy Interval Graph)**.** Let V be a finite set of vertices, and let

$$\mathcal{F} = \{\mu_1, \mu_2, \ldots, \mu_n\}$$

be a finite family of fuzzy intervals on the real line $\mathbb{R}$. Each fuzzy interval $\mu_i : \mathbb{R} \to [0, 1]$ is a normal, convex fuzzy subset of $\mathbb{R}$, meaning there exists a point $x_i \in \mathbb{R}$ such that $\mu_i(x_i) = 1$, and for any $y, z \in \mathbb{R}$ with $y \leq z$, the following holds:

$$\mu_i(w) \geq \min\{\mu_i(y), \mu_i(z)\} \quad \text{for all } w \text{ between } y \text{ and } z.$$

The *fuzzy interval graph* $G = (V, \mu_V, \rho)$ is defined as follows:

- The vertex membership function $\mu_V : V \to [0, 1]$ is given by:

$$\mu_V(v_i) = h(\mu_i) = \sup_{x \in \mathbb{R}} \mu_i(x) = 1,$$

  where $h(\mu_i)$ denotes the height of the fuzzy interval μ_i.

- The fuzzy adjacency relation $\rho : V \times V \to [0, 1]$ is defined by:

$$\rho(v_i, v_j) = \begin{cases} h(\mu_i \cap \mu_j) = \sup_{x \in \mathbb{R}} \min\{\mu_i(x), \mu_j(x)\}, & \text{if } i \neq j, \\ 0, & \text{if } i = j. \end{cases}$$

In this definition, the edge membership function $\rho(v_i, v_j)$ measures the degree of overlap between the fuzzy intervals μ_i and μ_j. The fuzzy interval graph captures the intersection properties of fuzzy intervals, extending the concept of interval graphs to the fuzzy context.

Definition 16 (Fuzzy Proper Interval Graph)**.** A *fuzzy proper interval graph* is a fuzzy interval graph $G = (V, \mu_V, \rho)$ where the family of fuzzy intervals $\mathcal{F} = \{\mu_1, \mu_2, \ldots, \mu_n\}$ satisfies the additional condition that no fuzzy interval is strictly contained within another.

Formally, for any two distinct fuzzy intervals μ_i and μ_j in $\mathcal{F}$, neither of the following holds:

- $\mu_i(x) \leq \mu_j(x)$ for all $x \in \mathbb{R}$ and there exists $x_0 \in \mathbb{R}$ such that $\mu_i(x_0) < \mu_j(x_0)$.
- $\mu_j(x) \leq \mu_i(x)$ for all $x \in \mathbb{R}$ and there exists $x_0 \in \mathbb{R}$ such that $\mu_j(x_0) < \mu_i(x_0)$.

This restriction ensures that no fuzzy interval is nested within another. The vertex membership function μ_V and the fuzzy adjacency relation ρ are defined similarly to those in the fuzzy interval graph, but the non-nesting condition on fuzzy intervals ensures that no interval is entirely contained within another.

2.4 interval graph in Intuitionistic fuzzy Graphs

Next, we consider Interval graphs in Intuitionistic Fuzzy Graphs. Intuitionistic fuzzy graphs are an extended version of fuzzy graphs and have been the subject of extensive study for over 15 years [92, 71, 90, 49]. Intuitionistic fuzzy graphs are related to the concept of intuitionistic fuzzy sets [113, 30, 6, 5]. The definitions of intuitionistic fuzzy graphs and intuitionistic fuzzy Interval graphs are provided below.

Definition 17 (Intuitionistic Fuzzy Graph (IFG))**.** [90] Let $G = (V, E)$ be a classical graph where V denotes the set of vertices and E denotes the set of edges. An *Intuitionistic Fuzzy Graph* (IFG) on G, denoted $G_{IF} = (A, B)$, is defined as follows:

(1) (μ_A, v_A) is an *Intuitionistic Fuzzy Set (IFS)* on the vertex set V. For each vertex $x \in V$, the degree of membership $\mu_A(x) \in [0, 1]$ and the degree of non-membership $v_A(x) \in [0, 1]$ satisfy:

$$\mu_A(x) + v_A(x) \leq 1$$

The value $1 - \mu_A(x) - v_A(x)$ represents the hesitancy or uncertainty regarding the membership of x in the set.

(2) (μ_B, v_B) is an *Intuitionistic Fuzzy Relation (IFR)* on the edge set E. For each edge $(x, y) \in E$, the degree of membership $\mu_B(x, y) \in [0, 1]$ and the degree of non-membership $v_B(x, y) \in [0, 1]$ satisfy:

$$\mu_B(x, y) + v_B(x, y) \leq 1$$

Additionally, the following constraints must hold for all $x, y \in V$:

$$\mu_B(x, y) \leq \mu_A(x) \wedge \mu_A(y)$$

$$v_B(x, y) \leq v_A(x) \vee v_A(y)$$

In this definition:

- $\mu_A(x)$ and $v_A(x)$ represent the degree of membership and non-membership of the vertex x, respectively.
- $\mu_B(x, y)$ and $v_B(x, y)$ represent the degree of membership and non-membership of the edge (x, y), respectively.
- If $v_A(x) = 0$ and $v_B(x, y) = 0$ for all $x \in V$ and $(x, y) \in E$, then the Intuitionistic Fuzzy Graph reduces to a Fuzzy Graph.

We define the Intuitionistic Fuzzy Interval Graph as follows. he Intuitionistic Fuzzy Interval Graph is a concept that combines the ideas of an Intuitionistic Fuzzy Graph and Interval graphs.

Definition 18 (Intuitionistic Fuzzy Interval Graph)**.** Let V be a finite set of vertices, and let

$$\mathcal{J} = \{(\mu_1, v_1), (\mu_2, v_2), \ldots, (\mu_n, v_n)\}$$

be a finite family of intuitionistic fuzzy intervals on the real line $\mathbb{R}$. Each pair (μ_i, v_i) consists of a membership function $\mu_i : \mathbb{R} \to [0,1]$ and a non-membership function $v_i : \mathbb{R} \to [0,1]$, such that:

$$\mu_i(x) + v_i(x) \leq 1 \quad \text{for all } x \in \mathbb{R},$$

where μ_i is convex and normal, meaning there exists a point $x_i \in \mathbb{R}$ such that $\mu_i(x_i) = 1$, and for any $y, z \in \mathbb{R}$ with $y \leq z$, the following holds:

$$\mu_i(w) \geq \min\{\mu_i(y), \mu_i(z)\} \quad \text{for all } w \text{ between } y \text{ and } z.$$

The *Intuitionistic Fuzzy Interval Graph* $G = (V, \mu_V, v_V, \rho_\mu, \rho_v)$ is defined as follows:

- The vertex membership function $\mu_V : V \to [0,1]$ and non-membership function $v_V : V \to [0,1]$ are given by:

$$\mu_V(v_i) = \sup_{x \in \mathbb{R}} \mu_i(x), \quad v_V(v_i) = \inf_{x \in \mathbb{R}} v_i(x).$$

- The fuzzy adjacency relations $\rho_\mu : V \times V \to [0,1]$ and $\rho_v : V \times V \to [0,1]$ are defined by:

$$\rho_\mu(v_i, v_j) = \begin{cases} \sup_{x \in \mathbb{R}} \min\{\mu_i(x), \mu_j(x)\}, & \text{if } i \neq j, \\ 0, & \text{if } i = j, \end{cases}$$

$$\rho_v(v_i, v_j) = \begin{cases} \inf_{x \in \mathbb{R}} \max\{v_i(x), v_j(x)\}, & \text{if } i \neq j, \\ 0, & \text{if } i = j. \end{cases}$$

This definition extends the classical interval graph by incorporating intuitionistic fuzzy intervals, allowing for both membership and non-membership degrees for vertices and edges.

Definition 19 (Intuitionistic Fuzzy Proper Interval Graph)**.** An *Intuitionistic Fuzzy Proper Interval Graph* is an Intuitionistic Fuzzy Interval Graph $G = (V, \mu_V, v_V, \rho_\mu, \rho_v)$ where the family of intuitionistic fuzzy intervals $\mathcal{J} = \{(\mu_1, v_1), (\mu_2, v_2), \ldots, (\mu_n, v_n)\}$ satisfies the additional condition that no intuitionistic fuzzy interval is strictly contained within another.

Formally, for any two distinct intuitionistic fuzzy intervals (μ_i, v_i) and (μ_j, v_j) in $\mathcal{J}$, neither of the following holds:

- $\mu_i(x) \leq \mu_j(x)$ for all $x \in \mathbb{R}$, and there exists $x_0 \in \mathbb{R}$ such that $\mu_i(x_0) < \mu_j(x_0)$,
- $v_i(x) \geq v_j(x)$ for all $x \in \mathbb{R}$, and there exists $x_0 \in \mathbb{R}$ such that $v_i(x_0) > v_j(x_0)$,

ensuring that no interval is nested within another in terms of both membership and non-membership degrees. The vertex membership function μ_V, non-membership function v_V, and fuzzy adjacency relations ρ_μ and ρ_v are defined similarly to those in the Intuitionistic Fuzzy Interval Graph.

2.5 interval graph in Neutrosophic Graphs

First, the definition of a neutrosophic graph is provided. As mentioned in the introduction, neutrosophic graphs are an extension of fuzzy graphs and Intuitionistic Fuzzy Graphs. A Neutrosophic Graph assigns truth, indeterminacy, and falsity membership degrees to each vertex and edge, representing uncertainty. Similar to fuzzy graphs, neutrosophic graphs have been the subject of extensive research [69, 2, 108, 106]. Neutrosophic graphs are related to the concept of Neutrosophic sets [4, 26, 109, 79, 105]. The definition is provided below[108].

Definition 20. [108] A neutrosophic graph $G = (V, E, \sigma = (\sigma_T, \sigma_I, \sigma_F), \mu = (\mu_T, \mu_I, \mu_F))$ is a graph where:

- $\sigma : V \to [0,1]^3$ assigns a triple $(\sigma_T(v), \sigma_I(v), \sigma_F(v))$ representing the truth, indeterminacy, and falsity membership degrees to each vertex $v \in V$.
- $\mu : E \to [0,1]^3$ assigns a triple $(\mu_T(e), \mu_I(e), \mu_F(e))$ representing the truth, indeterminacy, and falsity membership degrees to each edge $e \in E$.
- For every edge $e = v_i v_j \in E$, the following condition holds:

$$\mu_T(e) \leq \min(\sigma_T(v_i), \sigma_T(v_j)).$$

(1) σ is called the *neutrosophic vertex set*.

(2) μ is called the *neutrosophic edge set*.

(3) The number of vertices $|V|$ is the *order* of G, denoted by $O(G)$.

(4) The sum of the truth values over all vertices, $\sum_{v\in V}\sigma_T(v)$, is the *neutrosophic order* of G, denoted by $On(G)$.

(5) The number of edges $|E|$ is the *size* of G, denoted by $S(G)$.

(6) The sum of the truth values over all edges, $\sum_{e\in E}\mu_T(e)$, is the *neutrosophic size* of G, denoted by $Sn(G)$.

We define the Neutrosophic Interval Graph as follows. The Neutrosophic Interval Graphis a concept that combines the ideas of an Neutrosophic Graph and a Interval Graph.

Definition 21 (Neutrosophic Interval Graph)**.** Let V be a finite set of vertices, and let

$$\mathcal{N} = \{(\mu_1, \tau_1, \zeta_1), (\mu_2, \tau_2, \zeta_2), \dots, (\mu_n, \tau_n, \zeta_n)\}$$

be a finite family of neutrosophic intervals on the real line $\mathbb{R}$. Each triple (μ_i, τ_i, ζ_i) represents the truth-membership function $\mu_i : \mathbb{R} \to [0,1]$, the indeterminacy-membership function $\tau_i : \mathbb{R} \to [0,1]$, and the falsity-membership function $\zeta_i : \mathbb{R} \to [0,1]$ such that:

$$\mu_i(x) + \tau_i(x) + \zeta_i(x) = 1 \quad \text{for all } x \in \mathbb{R}.$$

The *Neutrosophic Interval Graph*

$$G = (V, \mu_V, \tau_V, \zeta_V, \rho_\mu, \rho_\tau, \rho_\zeta)$$

is defined as follows:

- The vertex membership functions $\mu_V : V \to [0,1]$, $\tau_V : V \to [0,1]$, and $\zeta_V : V \to [0,1]$ are given by:

$$\mu_V(v_i) = \sup_{x\in\mathbb{R}} \mu_i(x), \quad \tau_V(v_i) = \sup_{x\in\mathbb{R}} \tau_i(x), \quad \zeta_V(v_i) = \sup_{x\in\mathbb{R}} \zeta_i(x).$$

- The neutrosophic adjacency relations $\rho_\mu : V \times V \to [0,1]$, $\rho_\tau : V \times V \to [0,1]$, and $\rho_\zeta : V \times V \to [0,1]$ are defined by:

$$\rho_\mu(v_i, v_j) = \begin{cases} \sup_{x\in\mathbb{R}} \min\{\mu_i(x), \mu_j(x)\}, & \text{if } i \neq j, \\ 0, & \text{if } i = j, \end{cases}$$

$$\rho_\tau(v_i, v_j) = \begin{cases} \sup_{x\in\mathbb{R}} \min\{\tau_i(x), \tau_j(x)\}, & \text{if } i \neq j, \\ 0, & \text{if } i = j, \end{cases}$$

$$\rho_\zeta(v_i, v_j) = \begin{cases} \sup_{x\in\mathbb{R}} \min\{\zeta_i(x), \zeta_j(x)\}, & \text{if } i \neq j, \\ 0, & \text{if } i = j, \end{cases}$$

This definition extends classical interval graphs to the neutrosophic framework, accounting for truth, indeterminacy, and falsity degrees for both vertices and edges.

Definition 22 (Neutrosophic Proper Interval Graph)**.** A *Neutrosophic Proper Interval Graph* is a Neutrosophic Interval Graph

$$G = (V, \mu_V, \tau_V, \zeta_V, \rho_\mu, \rho_\tau, \rho_\zeta)$$

where the family of neutrosophic intervals

$$\mathcal{N} = \{(\mu_1, \tau_1, \zeta_1), (\mu_2, \tau_2, \zeta_2), \dots, (\mu_n, \tau_n, \zeta_n)\}$$

satisfies the condition that no neutrosophic interval is strictly contained within another.

Formally, for any two distinct neutrosophic intervals (μ_i, τ_i, ζ_i) and (μ_j, τ_j, ζ_j) in $\mathcal{N}$, neither of the following holds:

- $\mu_i(x) \leq \mu_j(x)$ for all $x \in \mathbb{R}$, and there exists $x_0 \in \mathbb{R}$ such that $\mu_i(x_0) < \mu_j(x_0)$,
- $\tau_i(x) \leq \tau_j(x)$ for all $x \in \mathbb{R}$, and there exists $x_0 \in \mathbb{R}$ such that $\tau_i(x_0) < \tau_j(x_0)$,
- $\zeta_i(x) \leq \zeta_j(x)$ for all $x \in \mathbb{R}$, and there exists $x_0 \in \mathbb{R}$ such that $\zeta_i(x_0) < \zeta_j(x_0)$,

ensuring that no neutrosophic interval is nested within another in terms of truth, indeterminacy, and falsity degrees. The vertex and edge membership functions μ_V, τ_V, ζ_V and adjacency relations ρ_μ, ρ_τ, ρ_ζ are defined similarly to those in the Neutrosophic Interval Graph.

2.6 Interval Graph in Turiyam Neutrosophic Graph

This section explores interval graphs and proper interval graphs within the framework of Turiyam Neutrosophic Graphs. A Turiyam Neutrosophic Graph extends classical graph theory by assigning four distinctparameters—truth, indeterminacy, falsity, and liberal state—to each vertex and edge.

Recent research on Turiyam Neutrosophic Graphs, which build upon and extend Neutrosophic Graphs byintroducing additional parameters, has gained significant attention [47, 45, 40, 34]. It is also known that Turiyam Neutrosophic Graphs can be generalized to Quadripartitioned Neutrosophic Graphs and related frameworks (cf.[107]).

The formal definition is presented below.

Definition 23 (Turiyam Neutrosophic Graph)**.** [47, 40, 34] Let $G = (V, E)$ be a classical graph with a finite set of vertices $V = \{v_i : i = 1, 2, \dots, n\}$ and edges $E = \{(v_i, v_j) : i, j = 1, 2, \dots, n\}$. A *Turiyam Neutrosophic Graph* of G, denoted $G^T = (V^T, E^T)$, is defined as follows:

(1) *Turiyam Neutrosophic Vertex Set*: For each vertex $v_i \in V$, the Turiyam Neutrosophic graph assigns the following mappings:

$$t(v_i), iv(v_i), fv(v_i), lv(v_i) : V \to [0, 1],$$

where:

- $t(v_i)$ is the truth value (tv) of the vertex v_i,
- $iv(v_i)$ is the indeterminacy value (iv) of v_i,
- $fv(v_i)$ is the falsity value (fv) of v_i,
- $lv(v_i)$ is the Turiyam Neutrosophic state (or liberal value) (lv) of v_i,

for all $v_i \in V$, such that the following condition holds for each vertex:

$$0 \leq t(v_i) + iv(v_i) + fv(v_i) + lv(v_i) \leq 4.$$

(2) *Turiyam Neutrosophic Edge Set*: For each edge $(v_i, v_j) \in E$, the Turiyam Neutrosophic graph assigns the following mappings:

$$t(v_i, v_j), iv(v_i, v_j), fv(v_i, v_j), lv(v_i, v_j) : E \to [0, 1],$$

where:

- $t(v_i, v_j)$ is the truth value of the edge (v_i, v_j),
- $iv(v_i, v_j)$ is the indeterminacy value of (v_i, v_j),
- $fv(v_i, v_j)$ is the falsity value of (v_i, v_j),
- $lv(v_i, v_j)$ is the Turiyam Neutrosophic state (or liberal value) of (v_i, v_j),

for all $(v_i, v_j) \in E$, such that the following condition holds for each edge:

$$0 \leq t(v_i, v_j) + iv(v_i, v_j) + fv(v_i, v_j) + lv(v_i, v_j) \leq 4.$$

In this case, V^T represents the Turiyam Neutrosophic vertex set of the graph G^T, and E^T represents the Turiyam Neutrosophic edge set of G^T.

We define the Turiyam Neutrosophic Interval Graph as follows. The Turiyam Neutrosophic Interval Graph is a concept that combines the ideas of an Turiyam Neutrosophic Graph and a Interval Graph.

Definition 24 (Turiyam Neutrosophic Interval Graph). Let V be a finite set of vertices, and let

$$\mathcal{T} = \{(\mu_1, iv_1, fv_1, lv_1), (\mu_2, iv_2, fv_2, lv_2), \dots, (\mu_n, iv_n, fv_n, lv_n)\}$$

be a family of Turiyam Neutrosophic intervals on the real line $\mathbb{R}$. Each quadruple

$$(\mu_i, iv_i, fv_i, lv_i)$$

$$0 \leq \mu_i(x) + iv_i(x) + fv_i(x) + lv_i(x) \leq 4 \quad \text{for all } x \in \mathbb{R}.$$

The *Turiyam Neutrosophic Interval Graph* $G^T = (V, \mu_V, iv_V, fv_V, lv_V, \rho_\mu, \rho_i v, \rho_f v, \rho_l v)$ is defined as follows:

- The vertex membership functions $\mu_V, iv_V, fv_V, lv_V : V \to [0,1]$ are defined as:

$$\mu_V(v_i) = \sup_{x \in \mathbb{R}} \mu_i(x), \quad iv_V(v_i) = \sup_{x \in \mathbb{R}} iv_i(x),$$

$$fv_V(v_i) = \sup_{x \in \mathbb{R}} fv_i(x), \quad lv_V(v_i) = \sup_{x \in \mathbb{R}} lv_i(x).$$

- The Turiyam Neutrosophic adjacency relations $\rho_\mu, \rho_i v, \rho_f v, \rho_l v : V \times V \to [0,1]$ are defined as:

$$\rho_\mu(v_i, v_j) = \begin{cases} \sup_{x \in \mathbb{R}} \min\{\mu_i(x), \mu_j(x)\}, & \text{if } i \neq j, \\ 0, & \text{if } i = j, \end{cases}$$

$$\rho_i v(v_i, v_j) = \begin{cases} \sup_{x \in \mathbb{R}} \min\{iv_i(x), iv_j(x)\}, & \text{if } i \neq j, \\ 0, & \text{if } i = j, \end{cases}$$

$$\rho_f v(v_i, v_j) = \begin{cases} \sup_{x \in \mathbb{R}} \min\{fv_i(x), fv_j(x)\}, & \text{if } i \neq j, \\ 0, & \text{if } i = j, \end{cases}$$

$$\rho_l v(v_i, v_j) = \begin{cases} \sup_{x \in \mathbb{R}} \min\{lv_i(x), lv_j(x)\}, & \text{if } i \neq j, \\ 0, & \text{if } i = j. \end{cases}$$

This definition extends the classical interval graph to the Turiyam Neutrosophic framework, allowing for truth,indeterminacy, falsity, and liberal state values for both vertices and edges.

Definition 25 (Turiyam Neutrosophic Proper Interval Graph). A Turiyam Neutrosophic Proper Interval Graph is a Turiyam Neutrosophic Interval Graph

$$G^T = (V, \mu_V, iv_V, fv_V, lv_V, \rho_\mu, \rho_i v, \rho_f v, \rho_l v)$$

where the family of Turiyam Neutrosophic intervals

$$\mathcal{T} = \{(\mu_1, iv_1, fv_1, lv_1), (\mu_2, iv_2, fv_2, lv_2), \dots, (\mu_n, iv_n, fv_n, lv_n)\}$$

satisfies the additional condition that no interval is strictly contained within another.

Formally, for any two distinct Turiyam Neutrosophic intervals $(\mu_i, iv_i, fv_i, lv_i)$ and $(\mu_j, iv_j, fv_j, lv_j)$ in $\mathcal{T}$, neither of the following holds:

- $\mu_i(x) \leq \mu_j(x)$ for all $x \in \mathbb{R}$, and there exists $x_0 \in \mathbb{R}$ such that $\mu_i(x_0) < \mu_j(x_0)$,
- $iv_i(x) \leq iv_j(x)$ for all $x \in \mathbb{R}$, and there exists $x_0 \in \mathbb{R}$ such that $iv_i(x_0) < iv_j(x_0)$,
- $fv_i(x) \leq fv_j(x)$ for all $x \in \mathbb{R}$, and there exists $x_0 \in \mathbb{R}$ such that $fv_i(x_0) < fv_j(x_0)$,
- $lv_i(x) \leq lv_j(x)$ for all $x \in \mathbb{R}$, and there exists $x_0 \in \mathbb{R}$ such that $lv_i(x_0) < lv_j(x_0)$,

ensuring that no Turiyam Neutrosophic interval is nested within another in terms of truth, indeterminacy, falsity, and liberal state values. The vertex membership functions μ_V, iv_V, fv_V, lv_V and the adjacency relations $\rho_\mu, \rho_i v, \rho_f v, \rho_l v$ are defined similarly to those in the Turiyam Neutrosophic Interval Graph.

3 Result in this paper

In this section, we present the results of this paper.

3.1 Property of neutrosophic interval raphs

We consider about neutrosophic interval graph. These properties also hold similarly for fuzzy graphs, intuitionistic fuzzy graphs and Turiyam Neutrosophic graphs.

Theorem 26. a Neutrosophic Interval Graph can be transformed into a classic Interval Graph.

Proof: To transform a Neutrosophic Interval Graph into a Classic Interval Graph, we can set all membership functions related to truth, indeterminacy, and falsity to specific values. Specifically, by setting the truth membership μ_V to 1 and both indeterminacy τ_V and falsity ζ_V to 0 for all vertices and edges, we eliminate the need for representing uncertainty or partial truth in the graph.

1. Set the membership values for vertices:

$$\mu_V(v) = 1, \quad \tau_V(v) = 0, \quad \zeta_V(v) = 0 \quad \forall v \in V.$$

This implies that every vertex fully belongs to the graph, with no indeterminacy or falsity.

2. Set the membership values for edges:

$$\rho_\mu(v_i, v_j) = 1, \quad \rho_\tau(v_i, v_j) = 0, \quad \rho_\zeta(v_i, v_j) = 0 \quad \forall (v_i, v_j) \in E.$$

This implies that if two intervals overlap, the edge between the corresponding vertices is fully included in the graph, with no indeterminacy or falsity.

After this transformation, all vertices and edges in the Neutrosophic Interval Graph behave exactly as in a classic Interval Graph. Specifically:

- The adjacency of vertices depends solely on the overlap of their intervals.
- There is no longer any notion of partial membership, uncertainty, or falsity affecting the structure of the graph.

Thus, the transformed graph is equivalent to a *classic Interval Graph*.

By setting the *truth membership* $\mu_V(v) = 1$ for all vertices and edges and setting the *indeterminacy* and *falsity memberships* $\tau_V(v) = 0$ and $\zeta_V(v) = 0$, the Neutrosophic Interval Graph is transformed into a classic Interval Graph. The adjacency structure based on interval overlap remains unchanged, and the resulting graph adheres to the traditional definition of an Interval Graph in graph theory. This transformation ensures that all uncertainty and fuzziness are eliminated, leaving a purely classical graph structure.

Therefore, the *Neutrosophic Interval Graph* can indeed be transformed into a *classic Interval Graph* by setting all membership values to specific constants, completing the proof. □

Theorem 27. *A Neutrosophic Proper Interval Graph can be transformed into a Classic Proper Interval Graph by assigning specific values to the truth, indeterminacy, and falsity membership functions.*

Proof: To transform a Neutrosophic Proper Interval Graph into a Classic Proper Interval Graph, we assign the following values to the truth, indeterminacy, and falsity membership functions:

- Set the truth membership $\mu_V(v) = 1$ for all vertices $v \in V$, and set the indeterminacy and falsity memberships to zero:

$$\mu_V(v) = 1, \quad \tau_V(v) = 0, \quad \zeta_V(v) = 0, \quad \forall v \in V.$$

  This implies that each vertex is fully present in the graph without uncertainty or falsity.

- Similarly, set the truth membership $\rho_\mu(v_i, v_j) = 1$ for all edges $(v_i, v_j) \in E$, and set the indeterminacy and falsity memberships for edges to zero:

$$\rho_\mu(v_i, v_j) = 1, \quad \rho_\tau(v_i, v_j) = 0, \quad \rho_\zeta(v_i, v_j) = 0, \quad \forall (v_i, v_j) \in E.$$

  This implies that all edges in the graph represent complete adjacency between vertices without uncertainty

In both Neutrosophic and Classic Proper Interval Graphs, the property of "properness" is essential. This property ensures that no interval is strictly contained within another. In the Neutrosophic context, this condition is enforced by the membership functions μ_V, τ_V, and ζ_V. By setting $\mu_V(v) = 1$ and $\tau_V(v) = \zeta_V(v) = 0$, we preserve this condition, as no vertex or edge will exhibit uncertainty or falsity, and the intervals remain distinct and non-nested.

By setting the truth membership $\mu_V = 1$ and $\rho_\mu = 1$ for all vertices and edges, and setting the indeterminacy and falsity memberships $\tau_V = 0$ and $\zeta_V = 0$ for all vertices and edges, the Neutrosophic Proper Interval Graph is transformed into a Classic Proper Interval Graph. The adjacency structure remains based solely on the overlap of intervals, and the non-nesting condition is preserved. Thus, the transformation is complete. □

Corollary 28. *A Fuzzy Interval Graph, Intuitionistic Fuzzy Interval Graph, or Turiyam Neutrosophic Interval Graph can be transformed into a classic Interval Graph.*

Proof: It can be proven in the same way as above. □

Corollary 29. *A Fuzzy proper Interval Graph, Intuitionistic Fuzzy proper Interval Graph, or Turiyam Neutrosophic proper Interval Graph can be transformed into a classic proper Interval Graph.*

Proof: It can be proven in the same way as above. □

Theorem 30. *Neutrosophic Graph can be represented as a Neutrosophic Interval Graph.*

Proof: We consider about Mapping vertices to intervals. Each vertex $v \in V$ in the Neutrosophic Graph is mapped to a neutrosophic interval on the real line, where:

$$\mu_V(v) = \sigma_T(v), \quad \tau_V(v) = \sigma_I(v), \quad \zeta_V(v) = \sigma_F(v).$$

This mapping ensures that the neutrosophic interval for each vertex reflects the truth, indeterminacy, and falsity degrees as defined in the Neutrosophic Graph.

We consider about mapping edges to intervals. For each edge $(v_i, v_j) \in E$, we define the neutrosophic adjacency relations ρ_μ, ρ_τ, and ρ_ζ in the Neutrosophic Interval Graph. These relations are determined based on the overlap of the neutrosophic intervals corresponding to v_i and v_j:

$$\rho_\mu(v_i, v_j) = \min(\mu_V(v_i), \mu_V(v_j)) = \min(\sigma_T(v_i), \sigma_T(v_j)),$$
$$\rho_\tau(v_i, v_j) = \min(\tau_V(v_i), \tau_V(v_j)) = \min(\sigma_I(v_i), \sigma_I(v_j)),$$
$$\rho_\zeta(v_i, v_j) = \min(\zeta_V(v_i), \zeta_V(v_j)) = \min(\sigma_F(v_i), \sigma_F(v_j)).$$

These relations mirror the conditions for the neutrosophic edge membership function μ in the original Neutrosophic Graph.

Since we have mapped both the vertices and edges of the Neutrosophic Graph to the corresponding neutrosophic intervals and relations in the Neutrosophic Interval Graph, we conclude that any Neutrosophic Graph can be represented as a Neutrosophic Interval Graph by this transformation. □

Corollary 31. *A Fuzzy Graph, Intuitionistic Fuzzy Graph, or Turiyam Neutrosophic Graph can be represented as a Fuzzy Interval Graph, Intuitionistic Fuzzy Interval Graph, or Turiyam Neutrosophic Interval Graph, respectively.*

Proof: It can be proven in the same way as above. □

Theorem 32. *Neutrosophic Proper Interval Graph is special case of Neutrosophic Interval Graph.*

Proof: Obviously holds. □

Theorem 33. *A Neutrosophic Interval Graph can be transformed into a Fuzzy Interval Graph, Intuitionistic Fuzzy Interval Graph, or Turiyam Neutrosophic Interval Graph.*

Proof: An interval graph is an undirected graph $G = (V, E)$ where each vertex corresponds to an interval on the real line, and two vertices are adjacent if and only if their intervals overlap. Formally, the edge set E of an interval graph is defined as:

$$E(G) = \{(u, v) \mid I_u \cap I_v \neq \emptyset\},$$

where I_u and I_v represent intervals on the real line associated with vertices u and v, respectively.

A *Neutrosophic Interval Graph* $G = (V, \mu_V, \tau_V, \zeta_V, \rho_\mu, \rho_\tau, \rho_\zeta)$ extends the traditional interval graph by associating *truth* (μ), *indeterminacy* (τ), and *falsity* (ζ) membership degrees with each vertex and edge. Two vertices u and v are adjacent if and only if their intervals overlap:

$$\text{Adjacency Condition: } (u, v) \in E \quad \text{if and only if } I_u \cap I_v \neq \emptyset.$$

Additionally, the neutrosophic membership functions satisfy:

$$\mu_V(v) + \tau_V(v) + \zeta_V(v) = 1, \quad \forall v \in V,$$

and similarly for edges.

We need to show how a Neutrosophic Interval Graph can be transformed into a *Fuzzy Interval Graph*, *Intuitionistic Fuzzy Interval Graph*, or *Turiyam Neutrosophic Interval Graph*, while preserving the core interval graph structure (i.e., adjacency based on overlapping intervals).

Next, we consider about Transformation into Fuzzy Interval Graph. A *Fuzzy Interval Graph* is a graph where each vertex and edge has a single membership degree, and two vertices are adjacent if their intervals overlap:

$$E(G^{fuzzy}) = \{(u, v) \mid I_u \cap I_v \neq \emptyset\}.$$

To transform a Neutrosophic Interval Graph into a Fuzzy Interval Graph, we focus on the *truth-membership function* $\mu_V(v)$ for each vertex and $\mu_\rho(v_i, v_j)$ for each edge, ignoring the indeterminacy and falsity components.

Define the transformed fuzzy membership functions as:

$$\mu_V^{fuzzy}(v_i) = \mu_V(v_i), \quad \mu_\rho^{fuzzy}(v_i, v_j) = \mu_\rho(v_i, v_j).$$

The resulting fuzzy interval graph $G^{fuzzy} = (V, \mu_V^{fuzzy}, \mu_\rho^{fuzzy})$ retains the interval graph structure, where adjacency is determined by interval overlap, and the edge membership function $\mu_\rho^{fuzzy}(v_i, v_j)$ quantifies the fuzzy strength of the connection.

Thus, this transformation ensures that G^{fuzzy} is a valid *Fuzzy Interval Graph*.

Next, we consider about Transformation into Intuitionistic Fuzzy Interval Graph. An *Intuitionistic Fuzzy Interval Graph* uses both membership and non-membership degrees for vertices and edges, subject to the condition:

$$\mu_V(v_i) + v_V(v_i) \leq 1, \quad \forall v_i \in V.$$

In the Neutrosophic Interval Graph, the truth-membership $\mu_V(v)$ and falsity-membership $\zeta_V(v)$ can be mapped to the intuitionistic fuzzy membership and non-membership degrees, respectively.

Define the transformed intuitionistic fuzzy membership and non-membership functions as:

$$\mu_V^{intuitionistic}(v_i) = \mu_V(v_i), \quad v_V^{intuitionistic}(v_i) = \zeta_V(v_i),$$

$$\mu_\rho^{intuitionistic}(v_i, v_j) = \mu_\rho(v_i, v_j), \quad v_\rho^{intuitionistic}(v_i, v_j) = \zeta_\rho(v_i, v_j).$$

Thus, the resulting Intuitionistic Fuzzy Interval Graph

$$G^{intuitionistic} = (V, \mu_V^{intuitionistic}, v_V^{intuitionistic}, \mu_\rho^{intuitionistic}, v_\rho^{intuitionistic})$$

maintains the interval-based adjacency condition.

Thus, this transformation results in a valid *Intuitionistic Fuzzy Interval Graph.*

Next, we consider about Transformation into Turiyam Neutrosophic Interval Graph. A *Turiyam Neutrosophic Interval Graph* includes four membership functions: truth μ, indeterminacy iv, falsity fv, and liberal lv, satisfying:

$$0 \leq \mu(v) + iv(v) + fv(v) + lv(v) \leq 4, \quad \forall v \in V.$$

In a Neutrosophic Interval Graph, the truth-membership $\mu_V(v)$, indeterminacy $\tau_V(v)$, and falsity $\zeta_V(v)$ can be mapped to the corresponding Turiyam Neutrosophic membership degrees. Define the liberal state lv to be 0 or context-specific.

Define the transformed membership functions as:

$$\mu_V^{turiyam}(v_i) = \mu_V(v_i), \quad iv_V^{turiyam}(v_i) = \tau_V(v_i)$$

$$fv_V^{turiyam}(v_i) = \zeta_V(v_i), \quad lv_V^{turiyam}(v_i) = 0.$$

The edge functions are similarly defined:

$$\mu_\rho^{turiyam}(v_i, v_j) = \mu_\rho(v_i, v_j), \quad iv_\rho^{turiyam}(v_i, v_j) = \tau_\rho(v_i, v_j)$$

$$fv_\rho^{turiyam}(v_i, v_j) = \zeta_\rho(v_i, v_j), \quad lv_\rho^{turiyam}(v_i, v_j) = 0.$$

The resulting *Turiyam Neutrosophic Interval Graph*

$$G^{turiyam} = (V, \mu_V^{turiyam}, iv_V^{turiyam}, fv_V^{turiyam}, lv_V^{turiyam}, \mu_\rho^{turiyam}, iv_\rho^{turiyam}, fv_\rho^{turiyam}, lv_\rho^{turiyam})$$

preserves the interval adjacency condition.

Thus, this transformation results in a valid *Turiyam Neutrosophic Interval Graph.* □

Corollary 34. *A Neutrosophic Interval Graph can be transformed into a Fuzzy Interval Graph, Intuitionistic Fuzzy Interval Graph, or Turiyam Neutrosophic Interval Graph.*

Proof: Obviously holds. □

Theorem 35. *In a Neutrosophic Interval Graph, the neutrosophic adjacency relations ρ_μ, ρ_τ, and ρ_ζ are symmetric.*

Proof: By definition, the neutrosophic adjacency relations are given by:

$$\rho_\mu(v_i, v_j) = \sup_{x \in \mathbb{R}} \min\{\mu_i(x), \mu_j(x)\},$$

$$\rho_\tau(v_i, v_j) = \sup_{x \in \mathbb{R}} \min\{\tau_i(x), \tau_j(x)\},$$

$$\rho_\zeta(v_i, v_j) = \sup_{x \in \mathbb{R}} \min\{\zeta_i(x), \zeta_j(x)\}.$$

For any $v_i, v_j \in V$, the minimum function is symmetric, i.e.,

$$\min\{\mu_i(x), \mu_j(x)\} = \min\{\mu_j(x), \mu_i(x)\}.$$

Similarly, the supremum over $x \in \mathbb{R}$ preserves this symmetry. Therefore,

$$\rho_\mu(v_i, v_j) = \rho_\mu(v_j, v_i).$$

The same argument applies to ρ_τ and ρ_ζ. Thus, all neutrosophic adjacency relations are symmetric. □

Corollary 36. *In a Neutrosophic Proper Interval Graph, the neutrosophic adjacency relations ρ_μ, ρ_τ, and ρ_ζ are symmetric.*

Proof: Obviously holds. □

Theorem 37. *In a Neutrosophic Proper Interval Graph, no neutrosophic interval is properly contained within another with respect to the truth-membership function μ_i.*

Proof: By definition, a Neutrosophic Proper Interval Graph is a Neutrosophic Interval Graph where, for any two distinct neutrosophic intervals (μ_i, τ_i, ζ_i) and (μ_j, τ_j, ζ_j), the following does not hold:

$$\mu_i(x) \leq \mu_j(x) \quad \forall x \in \mathbb{R}, \quad \text{and} \quad \exists x_0 \in \mathbb{R} \text{ such that } \mu_i(x_0) < \mu_j(x_0).$$

This condition explicitly states that no truth-membership function μ_i is entirely within another μ_j with a strict inequality at some point x_0. Therefore, no neutrosophic interval μ_i is properly contained within another μ_j in terms of the truth-membership functions. The same reasoning applies to the indeterminacy τ_i and falsity ζ_i functions. □

Theorem 38. *The class of Neutrosophic Interval Graphs is closed under taking induced subgraphs.*

Proof: Let $G = (V, \mu_V, \tau_V, \zeta_V, \rho_\mu, \rho_\tau, \rho_\zeta)$ be a Neutrosophic Interval Graph, and let $V' \subseteq V$. Consider the induced subgraph $G' = (V', \mu_{V'}, \tau_{V'}, \zeta_{V'}, \rho'_\mu, \rho'_\tau, \rho'_\zeta)$ where:

$$\mu_{V'}(v_i) = \mu_V(v_i), \quad \tau_{V'}(v_i) = \tau_V(v_i), \quad \zeta_{V'}(v_i) = \zeta_V(v_i), \quad \forall v_i \in V',$$

$$\rho'_\mu(v_i, v_j) = \rho_\mu(v_i, v_j), \quad \rho'_\tau(v_i, v_j) = \rho_\tau(v_i, v_j), \quad \rho'_\zeta(v_i, v_j) = \rho_\zeta(v_i, v_j), \quad \forall v_i, v_j \in V'.$$

Since G is a Neutrosophic Interval Graph, there exists a family of neutrosophic intervals $\mathcal{N} = \{(\mu_i, \tau_i, \zeta_i) \mid v_i \in V\}$. The subgraph G' corresponds to the subset of intervals $\mathcal{N}' = \{(\mu_i, \tau_i, \zeta_i) \mid v_i \in V'\}$. The adjacency relations in G' are determined by the overlaps of intervals in $\mathcal{N}'$ using the same definitions as in G.

Therefore, G' is a Neutrosophic Interval Graph corresponding to $\mathcal{N}'$. Thus, the class of Neutrosophic Interval Graphs is closed under taking induced subgraphs. □

3.2 Neutrosophic p-proper interval graph

Let p and q be integers. The definitions of a *neutrosophic p-proper interval graph* and a *neutrosophic q-improper interval graph*, which extend the concepts of p-proper interval graphs and q-improper interval graphs, are provided below.

Definition 39. A *neutrosophic p-proper interval graph* is a neutrosophic graph

$$G = (V, E, \sigma_T, \sigma_I, \sigma_F)$$

where the truth-membership intervals I_v, corresponding to the vertices $v \in V$, satisfy the condition that no interval is properly contained within more than p others. Formally, for each vertex $u \in V$, let $\mathcal{I}_T = \{I_v \mid v \in V\}$ represent the truth-membership intervals. The graph G is p-proper if for every interval $I_u \in \mathcal{I}_T$, the number of intervals $I_v \subset I_u$ is at most p:

$$\forall u \in V, \quad |\{v \in V \mid I_v \subset I_u\}| \leq p.$$

This definition ensures that the intervals corresponding to the truth-membership values in the neutrosophic framework adhere to the p-proper constraint.

Definition 40. A *neutrosophic q-improper interval graph* is a neutrosophic graph

$$G = (V, E, \sigma_T, \sigma_I, \sigma_F)$$

where the truth-membership intervals I_v, corresponding to the vertices $v \in V$, satisfy the condition that no interval properly contains more than q others. Formally, for each vertex $u \in V$, let $\mathcal{I}_T = \{I_v \mid v \in V\}$ represent the truth-membership intervals. The graph G is q-improper if for every interval $I_u \in \mathcal{I}_T$, the number of intervals $I_u \subset I_v$ is at most q:

$$\forall u \in V, \quad |\{v \in V \mid I_u \subset I_v\}| \leq q.$$

This ensures that the truth-membership intervals in the neutrosophic graph follow the q-improper constraint.

Theorem 41. *Neutrosophic p-proper Interval Graph is a special type of neutrosophic proper Interval Graph.*

Proof: Obviously holds. □

Corollary 42. *Neutrosophic p-proper Interval Graph is a special type of neutrosophic Interval Graph.*

Proof: Obviously holds. □

Corollary 43. *Neutrosophic q-improper Interval Graph is a special type of neutrosophic Interval Graph.*

Proof: Obviously holds. □

Theorem 44. *A neutrosophic 0-proper interval graph is a neutrosophic proper interval graph.*

Proof: By definition, a neutrosophic p-proper interval graph is a neutrosophic interval graph where no interval is properly contained within more than p other intervals. When $p = 0$, no interval is properly contained within any other interval. This is precisely the condition for a neutrosophic proper interval graph, where no interval properly contains any other interval. Hence, a neutrosophic 0-proper interval graph is a neutrosophic proper interval graph. □

Theorem 45. *A neutrosophic 0-improper interval graph is a neutrosophic proper interval graph.*

Proof: A neutrosophic q-improper interval graph is defined as a neutrosophic interval graph where no interval properly contains more than q other intervals. For $q = 0$, no interval properly contains any other interval, which is exactly the condition for a neutrosophic proper interval graph. Therefore, a neutrosophic 0-improper interval graph is a neutrosophic proper interval graph. □

Theorem 46. *A Fuzzy p-proper Interval Graph, Intuitionistic Fuzzy p-proper Interval Graph, Neutrosophic p-proper Interval Graph or Turiyam Neutrosophic p-proper Interval Graph can be transformed into a classic p-proper Interval Graph.*

Proof: Obviously holds. □

Corollary 47. *A Fuzzy q-improper Interval Graph, Intuitionistic Fuzzy q-improper Interval Graph, Neutrosophic q-improper Interval Graph or Turiyam Neutrosophic q-improper Interval Graph can be transformed into a classic q-improper Interval Graph.*

Proof: Obviously holds. □

3.3 Fuzzy Intersection Graph and Fuzzy Interval Graph

We will examine the relationship between a Fuzzy Intersection Graph and a Fuzzy Interval Graph. The definition of a Fuzzy Intersection Graph is provided below [98, 81, 93, 20, 55] .

Definition 48 (Fuzzy Intersection Graph)**.** A *Fuzzy Intersection Graph* is a graph $G = (V, E, \sigma, \mu)$ where:

- V is the set of vertices.
- $E \subseteq V \times V$ is the set of edges.
- $\sigma : V \to [0, 1]$ is a membership function that assigns a degree of membership to each vertex $v \in V$.
- $\mu : V \times V \to [0, 1]$ is a fuzzy relation representing the strength of the connection (degree of membership) between each pair of vertices $(u, v) \in V \times V$.

The edge set E of the fuzzy intersection graph is defined based on the membership functions of the vertices and the fuzzy relation. Specifically, for each pair $(u, v) \in V \times V$, the edge (u, v) exists in the fuzzy intersection graph with the membership degree:

$$\mu(u, v) = \min(\sigma(u), \sigma(v))$$

if the Euclidean distance between the corresponding points of u and v satisfies the condition for intersection, and $\mu(u, v) = 0$ otherwise.

In this way, the fuzzy intersection graph generalizes the concept of an intersection graph by incorporating fuzzy set theory, allowing for partial membership and gradual relationships between vertices and edges.

The following theorem is well-known in the context of fuzzy intersection graphs.

Theorem 49. [37] *Any undirected fuzzy graph $G = (V, \sigma, \mu)$ can be represented as a fuzzy intersection graph.*

We will explore the relationship between Fuzzy Interval Graphs and Fuzzy Intersection Graphs. The theorem is presented as follows.

Theorem 50. *A Fuzzy Interval Graph is a Fuzzy Intersection Graph.*

Proof: Let $G = (V, \mu_V, \rho)$ be a Fuzzy Interval Graph where:

- V is the set of vertices.
- $\mu_V(v_i)$ represents the membership function for each vertex $v_i \in V$.
- $\rho(v_i, v_j)$ represents the fuzzy adjacency relation between vertices v_i and v_j.

In a Fuzzy Interval Graph, each vertex v_i is associated with a fuzzy interval $\mu_i : \mathbb{R} \to [0, 1]$ on the real line. The fuzzy adjacency relation $\rho(v_i, v_j)$ measures the degree of overlap between the fuzzy intervals μ_i and μ_j corresponding to the vertices v_i and v_j. Specifically, for any two distinct vertices v_i and v_j, the adjacency relation is given by:

$$\rho(v_i, v_j) = \sup_{x \in \mathbb{R}} \min\{\mu_i(x), \mu_j(x)\},$$

which calculates the maximum degree of overlap between the intervals μ_i and μ_j. The function $\min\{\mu_i(x), \mu_j(x)\}$ captures the intersection of the two intervals, as it represents the smallest membership value at any point $x \in \mathbb{R}$ common to both intervals.

Next, consider a *Fuzzy Intersection Graph* $G' = (V, E, \sigma, \mu)$, defined as follows:

- V is the set of vertices.
- $\sigma : V \to [0, 1]$ is a membership function that assigns a membership degree to each vertex $v \in V$.
- $\mu : V \times V \to [0, 1]$ is the fuzzy relation representing the strength of the connection between any two vertices $v_i, v_j \in V$, defined as:

$$\mu(v_i, v_j) = \min(\sigma(v_i), \sigma(v_j)).$$

To prove that a fuzzy interval graph is a fuzzy intersection graph, we need to show that the fuzzy adjacency relation $\rho(v_i, v_j)$ in the fuzzy interval graph is equivalent to the edge membership function $\mu(v_i, v_j)$ in the fuzzy intersection graph.

By the construction of a fuzzy interval graph, the adjacency relation $\rho(v_i, v_j)$ is based on the intersection of the fuzzy intervals μ_i and μ_j, as given by the formula:

$$\rho(v_i, v_j) = \sup_{x \in \mathbb{R}} \min\{\mu_i(x), \mu_j(x)\}.$$

This represents the highest degree of intersection between the two fuzzy intervals. On the other hand, in a fuzzy intersection graph, the relation $\mu(v_i, v_j) = \min(\sigma(v_i), \sigma(v_j))$ directly uses the membership degrees $\sigma(v_i)$ and $\sigma(v_j)$, which can be interpreted as the maximum heights (i.e., supremum values) of the corresponding fuzzy intervals μ_i and μ_j.

Thus, by setting $\sigma(v_i) = \sup_{x \in \mathbb{R}} \mu_i(x)$ for each vertex $v_i \in V$, we ensure that the fuzzy relation in the fuzzy intersection graph is exactly the same as the fuzzy adjacency relation in the fuzzy interval graph:

$$\rho(v_i, v_j) = \mu(v_i, v_j) = \min(\sigma(v_i), \sigma(v_j)) = \sup_{x \in \mathbb{R}} \min\{\mu_i(x), \mu_j(x)\}.$$

□

Corollary 51. *A Fuzzy Proper Interval Graph is a Fuzzy Intersection Graph.*

Proof: It can be proven in the same way as above. □

4 Future Perspectives of This Research

In this section, we briefly outline the future prospects of this study. Potential directions include exploring the Neutrosophic Interval OffGraph [38, 102], the Neutrosophic Proper Interval OffGraph, and the Interval

SuperHyperGraph (cf. [104, 42, 41, 35, 106]). These avenues offer promising opportunities for further developmentand applications of the concepts introduced in this research.

Funding

This research received no external funding.

Acknowledgments

We humbly extend our heartfelt gratitude to everyone who has provided invaluable support, enabling the successful completion of this paper. We also express our sincere appreciation to all readers who have taken the time to engage with this work. Furthermore, we extend our deepest respect and gratitude to the authors of the references cited in this paper. Thank you for your significant contributions.

Data Availability

This paper does not involve any data analysis.

Ethical Approval

This article does not involve any research with human participants or animals.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Disclaimer

This study primarily focuses on theoretical aspects, and its application to practical scenarios has not yet been validated. Future research may involve empirical testing and refinement of the proposed methods. The authors have made every effort to ensure that all references cited in this paper are accurate and appropriately attributed.However, unintentional errors or omissions may occur. The authors bear no legal responsibility for in accuraciesin external sources, and readers are encouraged to verify the information provided in the references independently.Furthermore, the interpretations and opinions expressed in this paper are solely those of the authors and do not necessarily reflect the views of any affiliated institutions.

References

[1] Muhammad Akram, Hafsa M Malik, Sundas Shahzadi, and Florentin Smarandache. Neutrosophic soft rough graphs with application. Axioms, 7(1):14, 2018.

[2] Muhammad Akram and Gulfam Shahzadi. Operations on single-valued neutrosophic graphs. Infinite Study, 2017.

[3] Shawkat Alkhazaleh. Neutrosophic vague set theory. Critical Review, 10:29–39, 2015.

[4] Adrián Alejandro Alvaracín Jarrín, David Santiago Proaño Tamayo, Salomón Alejandro Montecé Giler, Juan Carlos Arandia Zambrano, and Dante Manuel Macazana. Neutrosophic statistics applied in social science. Neutrosophic Sets and Systems, 44(1):1, 2021.

[5] Krassimir T. Atanassov. Intuitionistic fuzzy sets - theory and applications. In Studies in Fuzziness and Soft Computing, 1999.

[6] Krassimir T. Atanassov. On intuitionistic fuzzy sets theory. In Studies in Fuzziness and Soft Computing, 2012.

[7] Keszegh Balázs. Coloring intersection hypergraphs of pseudo-disks. Discrete & Computational Geometry, 64(3):942–964, 2020.

[8] Mehdi Behzad, Gary Chartrand, and John K Cooper Jr. The colour numbers of complete graphs. Journal of the London Mathematical Society, 1(1):226–228, 1967.

[9] Alan A Bertossi and Alessandro Gori. Total domination and irredundance in weighted interval graphs. SIAM journal on discrete mathematics, 1(3):317–327, 1988.

[10] Jeffrey Beyerl and Robert E Jamison. Interval graphs with containment restrictions. arXiv preprint arXiv:1109.6675, 2011.

[11] Said Broumi, D Ajay, P Chellamani, Lathamaheswari Malayalan, Mohamed Talea, Assia Bakali, Philippe Schweizer, and Saeid Jafari. Interval valued pentapartitioned neutrosophic graphs with an application to mcdm. Operational Research in Engineering Sciences: Theory and Applications , 5(3):68–91, 2022.[12] Elena Bulgaru and Vitaly I Voloshin. Mixed interval hypergraphs. Discrete Applied Mathematics, 77(1):29–41, 1997.

[13] Yixin Cao. Linear recognition of almost interval graphs. In Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1096–1115. SIAM, 2016.

[14] Renato M Capocelli and Aldo De Luca. Fuzzy sets and decision theory. Information and control, 23(5):446–473, 1973.

[15] L Sunil Chandran and Naveen Sivadasan. Boxicity and treewidth. Journal of Combinatorial Theory, Series B, 97(5):733–744, 2007.

[16] Chiuyuan Chen, Chin-Chen Chang, and Gerard J Chang. Proper interval graphs and the guard problem. Discrete Mathematics,170(1-3):223–230, 1997.

[17] Mingxia Chen, Jianbo Li, Jianping Li, Weidong Li, and Lusheng Wang. Some approximation algorithms for the clique partition problem in weighted interval graphs. Theoretical computer science, 381(1-3):124–133, 2007.

[18] Joel E Cohen and David W Stephens. Food webs and niche space.(MPB-11), Volume 11, volume 11. Princeton University Press, 2020.

[19] Derek G Corneil, Stephan Olariu, and Lorna Stewart. Asteroidal triple-free graphs. SIAM Journal on Discrete Mathematics, 10(3):399–430, 1997.

[20] WL Craine. Characterizations of fuzzy interval graphs. Fuzzy Sets and Systems, 68(2):181–193, 1994.

[21] Roger E Critchlow Jr and Stephen C Stearns. The structure of food webs. The American Naturalist, 120(4):478–499, 1982.

[22] Elias Dahlhaus. Improved efficient parallel algorithms to recognize interval graphs and interval

hypergraphs. In Proceedings of the Thirtieth Hawaii International Conference on System Sciences, volume 1, pages 172–181. IEEE, 1997.

[23] Sipra Das. Intersection digraphs: an analogue of intersection graphs. PhD thesis, University of North Bengal, 1990.

[24] Shilpa Dasgupta. On Characterizations and Structure of Internal Digraphs and Unit Probe Interval Graphs. PhD thesis, University of Colorado Denver, 2012.

[25] Aldo De Luca and Settimo Termini. A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory. In Readings in fuzzy sets for intelligent systems, pages 197–202. Elsevier, 1993.

[26] Irfan Deli. Refined neutrosophic sets and refined neutrosophic soft sets: theory and applications. In Handbook of research on generalized and hybrid set structures and applications for soft computing, pages 321–343. IGI Global, 2016.

[27] Dariusz Dereniowski. From pathwidth to connected pathwidth. SIAM Journal on Discrete Mathematics, 26(4):1709–1732, 2012.

[28] Reinhard Diestel. Graph theory. Springer (print edition); Reinhard Diestel (eBooks), 2024.

[29] David P Dobkin, Steven J Friedman, and Kenneth J Supowit. Delaunay graphs are almost as good as complete graphs.Discrete & Computational Geometry, 5:399–407, 1990.

[30] P. A. Ejegwa, S. O. Akowe, P. M. Otene, and J. M. Ikyule. An overview on intuitionistic fuzzy sets. International Journal of Scientific & Technology Research, 3:142–145, 2014.

[31] Thomas Erlebach, Klaus Jansen, and Eike Seidel. Polynomial-time approximation schemes for geometric intersection graphs.SIAM Journal on Computing, 34(6):1302–1323, 2005.

[32] Martin Farber. Characterizations of strongly chordal graphs. Discrete Mathematics, 43(2-3):173–189, 1983.

[33] Takaaki Fujita. Survey of trees, forests, and paths in fuzzy and neutrosophic graphs. Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond, page 477.

[34] Takaaki Fujita. Note for neutrosophic incidence and threshold graph. SciNexuses, 1:97–125, 2024.

[35] Takaaki Fujita. Review of some superhypergraph classes: Directed, bidirected, soft, and rough. In Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Second Volume). Biblio Publishing, 2024.

[36] Takaaki Fujita. Superhypergraph neural networks and plithogenic graph neural networks: Theoretical foundations. arXiv preprint arXiv:2412.01176, 2024.

[37] Takaaki Fujita. Unit disk graphs in fuzzy and neutrosophic graphs. June 2024. Available under CC BY 4.0 license.

[38] Takaaki Fujita. Advancing uncertain combinatorics through graphization, hyperization, and uncertainization: Fuzzy, neutrosophic, soft, rough, and beyond. 2025.

[39] Takaaki Fujita. A comprehensive discussion on fuzzy hypersoft expert, superhypersoft, and indetermsoft graphs. Neutrosophic Sets and Systems, 77:241–263, 2025.

[40] Takaaki Fujita and Florentin Smarandache. Antipodal turiyam neutrosophic graphs. Neutrosophic Optimization and Intelligent Systems, 5:1–13, 2024.

[41] Takaaki Fujita and Florentin Smarandache. A concise study of some superhypergraph classes. Neutrosophic Sets and Systems,77:548–593, 2024.

[42] Takaaki Fujita and Florentin Smarandache. Fundamental computational problems and algorithms for superhypergraphs. In Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft,Rough, and Beyond (Second Volume). Biblio Publishing, 2024.

[43] Takaaki Fujita and Florentin Smarandache. Study for general plithogenic soft expert graphs. Plithogenic Logic and Computation, 2:107–121, 2024.

[44] Takaaki Fujita and Florentin Smarandache. Survey of planar and outerplanar graphs in fuzzy and neutrosophic graphs.Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft,Rough, and Beyond, page 366, 2024.

[45] Takaaki Fujita and Florentin Smarandache. A compact exploration of turiyam neutrosophic competition graphs. Neutrosophic Optimization and Intelligent Systems, 5:29–37, 2025.

[46] Delbert Fulkerson and Oliver Gross. Incidence matrices and interval graphs. Pacific journal of mathematics, 15(3):835–855,1965.

[47] GA Ganati, VNS Rao Repalle, MA Ashebo, and M Amini. Turiyam graphs and its applications. Information Sciences Letters,12(6):2423–2434, 2023.

[48] Gamachu Adugna Ganati, VN Srinivasa Rao Repalle, and Mamo Abebe Ashebo. Social network analysis by turiyam graphs.BMC Research Notes, 16(1):170, 2023.

[49] A. Nagoor Gani and S. Shajitha Begum. Perfect intuitionistic fuzzy graphs. Bulletin of Pure & Applied Sciences- Mathematics and Statistics, pages 145–152, 2011.

[50] Frédéric Gardi. Mutual exclusion scheduling with interval graphs or related classes. part ii. Discrete applied mathematics,156(5):794–812, 2008.

[51] Frédéric Gardi. Mutual exclusion scheduling with interval graphs or related classes, part i. Discrete Applied Mathematics,157(1):19–35, 2009.

[52] Fanica Gavril. Algorithms on circular-arc graphs. Networks, 4(4):357–369, 1974.

[53] Fănică Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. Journal of Combinatorial Theory,Series B, 16(1):47–56, 1974.

[54] Julian Gerstenberg. Exchangeable interval hypergraphs and limits of ordered discrete structures. 2020.

[55] Ganesh Ghorai and Madhumangal Pal. G. ghorai, m. pal: Applications of bipolar fuzzy sets in

interval graphs. TWMS Journal of Applied and Engineering Mathematics, 8(2):411–424, 2018.

[56] Erhard Godehardt and Jerzy Jaworski. Two models of random intersection graphs for classification. In Exploratory Data Analysis in Empirical Research: Proceedings of the 25 th Annual Conference of the Gesellschaft für Klassifikation eV,University of Munich, March 14–16, 2001, pages 67–81. Springer, 2003.

[57] Martin Charles Golumbic. Interval graphs. In Annals of Discrete Mathematics, volume 57, pages 171–202. Elsevier, 2004.

[58] Alexander Grigoriev, Athanassios Koutsonas, and Dimitrios M Thilikos. Bidimensionality of geometric intersection graphs. In International Conference on Current Trends in Theory and Practice of Informatics, pages 293–305. Springer, 2014.

[59] Udaiprakash I Gupta, Der-Tsai Lee, and JY-T Leung. Efficient algorithms for interval graphs and circular-arc graphs. Networks,12(4):459–467, 1982.

[60] Marisa Gutierrez and Lía Oubiña. Minimum proper interval graphs. Discrete mathematics, 142(1-3):77–85, 1995.

[61] Grzegorz Gutowski, Konstanty Junosza-Szaniawski, Felix Klesen, Paweł Rzążewski, Alexander Wolff, and Johannes Zink.Coloring and recognizing mixed interval graphs. 2023.

[62] Grzegorz Gutowski, Florian Mittelstädt, Ignaz Rutter, Joachim Spoerhase, Alexander Wolff, and Johannes Zink. Coloring mixed and directional interval graphs. In International Symposium on Graph Drawing and Network Visualization, pages 418–431. Springer, 2022.

[63] Grzegorz Gutowski1, Florian Mittelstädt, Ignaz Rutter, Joachim Spoerhase1, Alexander Wolff, and Johannes Zink. Check for updates coloring mixed and directional interval graphs. In Graph Drawing and Network Visualization: 30th International Symposium, GD 2022, Tokyo, Japan, September 13–16, 2022, Revised Selected Papers, volume 13764, page 418. SpringerNature, 2023.

[64] Rudolf Halin. A theorem on n-connected graphs. Journal of Combinatorial Theory, 7(2):150–154, 1969.

[65] Pinar Heggernes, Daniel Meister, and Yngve Villanger. Induced subgraph isomorphism on interval and proper interval graphs.In International Symposium on Algorithms and Computation, pages 399–409. Springer, 2010.

[66] Robert Hickingbotham. Induced subgraphs and path decompositions. arXiv preprint arXiv:2206.15054, 2022.

[67] Sandy Irani and Vitus Leung. Scheduling with conflicts on bipartite and interval graphs. Journal of Scheduling, 6:287–307,2003.

[68] R JERROLD. Interval graphs and maps of dna. Bulletin ofMothematical Biology Vol, 48(2):189–195.

[69] Vasantha Kandasamy, K Ilanthenral, and Florentin Smarandache. Neutrosophic graphs: a new

dimension to graph theory.Infinite Study, 2015.

[70] Haim Kaplan and Ron Shamir. Pathwidth, bandwidth, and completion problems to proper interval graphs with small cliques.SIAM Journal on Computing, 25(3):540–561, 1996.

[71] Murugan Kaviyarasu, Mohammed Alqahtani, Murugesan Rajeshwari, and Gopikumar Sudalaimuthu. Complex t-intuitionistic fuzzy graph with applications of rubber industrial water wastes. Mathematics, 12(13):1950, 2024.

[72] Balázs Keszegh. Coloring intersection hypergraphs of pseudo-disks. Discrete & Computational Geometry, 64(3):942–964, 2020.

[73] Alexandr Kostochka and Jan Kratochvíl. Covering and coloring polygon-circle graphs. Discrete Mathematics, 1997.

[74] C. G. Lekkerkerker and J. C. Boland. Representation of a finite graph by a set of intervals on the real line. Fundamenta Mathematicae, 51:45–64, 1962.

[75] Benjamin Lévêque, David Y Lin, Frédéric Maffray, and Nicolas Trotignon. Detecting induced subgraphs. Discrete Applied Mathematics, 157(17):3540–3551, 2009.

[76] Hongxing Li and Vincent C Yen. Fuzzy sets and fuzzy decision-making. CRC press, 1995.

[77] Peng Li and Yaokun Wu. Maximal neighborhood search and rigid interval graphs. Journal of Graph Algorithms and Applications, 17(3):245–264, 2013.

[78] J Richard Lundgren. Food webs, competition graphs, competition-common enemy graphs, and niche graphs. Applications of combinatorics and graph theory to the biological and social sciences, pages 221–243, 1989.

[79] Pinaki Majumdar. Neutrosophic sets and its applications to decision making. In Computational Intelligence for Big Data Analysis: Frontier Advances and Applications, pages 97–115. Springer, 2015.

[80] Bubai Manna. Minimum consistent subset in interval graphs and circle graphs. arXiv preprint arXiv:2405.14493, 2024.

[81] MLN McAllister. Fuzzy intersection graphs. Computers & Mathematics with Applications, 15(10):871–886, 1988.

[82] Terry A McKee and Fred R McMorris. Topics in intersection graph theory. SIAM, 1999.

[83] Fred R McMorris, Chi Wang, and Peisen Zhang. On probe interval graphs. Discrete Applied Mathematics, 88(1-3):315–324,1998.

[84] Joao Meidanis and Patrıcia Takaki. Interval graphs with repeats and the dna fragment assembly problem. Technical report,Technical Report 02-14, IC-Unicamp, 2002.

[85] John I Moore Jr. Interval hypergraphs and d-interval hypergraphs. Discrete Mathematics, 17(2):173–179, 1977.

[86] John N Mordeson and Premchand S Nair. Fuzzy graphs and fuzzy hypergraphs, volume 46. Physica, 2012.

[87] Nabil H Mustafa and János Pach. On the zarankiewicz problem for intersection hypergraphs. In International Symposium on Graph Drawing, pages 207–216. Springer, 2015.

[88] S Nikoletseas, Christoforos Raptopoulos, and P Spirakis. Large independent sets in general random intersection graphs.Theoretical Computer Science, 406(3):215–224, 2008.

[89] TM Nishad, Talal Ali Al-Hawary, and B Mohamed Harif. General fuzzy graphs. Ratio Mathematica, 47, 2023.

[90] R Parvathi and MG Karunambigai. Intuitionistic fuzzy graphs. In Computational Intelligence, Theory and Applications: International Conference 9th Fuzzy Days in Dortmund, Germany, Sept. 18–20, 2006 Proceedings, pages 139–150. Springer,2006.

[91] Andrzej Proskurowski and Jan Arne Telle. Classes of graphs with restricted interval models. Discrete Mathematics & Theoretical Computer Science, 3, 1999.

[92] Hossein Rashmanlou, Sovan Samanta, Madhumangal Pal, and Rajab Ali Borzooei. Intuitionistic fuzzy graphs with categorical

properties. Fuzzy information and Engineering, 7(3):317–334, 2015.

[93] Sreenanda Raut and Madhumangal Pal. Fuzzy intersection graph: a geometrical approach. Journal of Ambient Intelligence and Humanized Computing, pages 1–25, 2022.

[94] Arnold L Rosenberg. Interval hypergraphs. Graphs and Algorithms (RB Richter, ed.) Contemporary Mathematics, 89:27–44, 1989.

[95] Azriel Rosenfeld. Fuzzy graphs. In Fuzzy sets and their applications to cognitive and decision processes, pages 77–95. Elsevier, 1975.

[96] Amita Samanta Adhya, Sukumar Mondal, and Sambhu Charan Barman. Edge-vertex domination on interval graphs. Discrete Mathematics, Algorithms and Applications, 16(02):2350015, 2024.

[97] Robert Scheffler. Semi-proper interval graphs. Discrete Applied Mathematics, 360:22–41, 2025.

[98] MK Sen, G Chowdhury, and DS Malik. Fuzzy intersection graphs of fuzzy semigroups. New Mathematics and Natural Computation, 2(01):1–10, 2006.

[99] Alan Shuchat, Randy Shull, Ann N Trenk, and Lee C West. Unit mixed interval graphs. arXiv preprint arXiv:1405.4247, 2014.

[100] Karen B Singer. Random intersection graphs. The Johns Hopkins University, 1996.

[101] Florentin Smarandache. A unifying field in logics: Neutrosophic logic. In Philosophy, pages 1–141. American Research Press,1999.

[102] Florentin Smarandache. Neutrosophic Overset, Neutrosophic Underset, and Neutrosophic Offset. Similarly for Neutrosophic Over-/Under-/Off-Logic, Probability, and Statistics. Infinite Study, 2016.

[103] Florentin Smarandache. Plithogeny, plithogenic set, logic, probability, and statistics. arXiv preprint arXiv:1808.03948, 2018.

[104] Florentin Smarandache. n-superhypergraph and plithogenic n-superhypergraph. Nidus Idearum,

7:107–113, 2019.

[105] Florentin Smarandache. New types of neutrosophic set/logic/probability, neutrosophic over-/under-/off-set, neutrosophic refined set, and their extension to plithogenic set/logic/probability, with applications. MDPI, 2019.

[106] Florentin Smarandache. Extension of HyperGraph to n-SuperHyperGraph and to Plithogenic n-SuperHyperGraph, and Extension of HyperAlgebra to n-ary (Classical-/Neutro-/Anti-) HyperAlgebra. Infinite Study, 2020.

[107] Florentin Smarandache. Ambiguous set is a subclass of the double refined indeterminacy neutrosophic set, and of the refined neutrosophic set in general. Neutrosophic Sets & Systems, 58, 2023.

[108] Florentin Smarandache and Said Broumi. Neutrosophic graph theory and algorithms. IGI Global, 2019.

[109] Florentin Smarandache, Said Broumi, Prem Kumar Singh, Chun-fang Liu, V Venkateswara Rao, Hai-Long Yang, Ion Patrascu,and Azeddine Elhassouny. Introduction to neutrosophy and neutrosophic environment. In Neutrosophic Set in Medical Image Analysis, pages 3–29. Elsevier, 2019.

[110] Florentin Smarandache and Nivetha Martin. Plithogenic n-super hypergraph in novel multi-attribute decision making. Infinite Study, 2020.

[111] Metawee Songsaeng and Aiyared Iampan. Neutrosophic set theory applied to UP-algebras. Infinite Study, 2019.

[112] Fazeelat Sultana, Muhammad Gulistan, Mumtaz Ali, Naveed Yaqoob, Muhammad Khan, Tabasam Rashid, and Tauseef Ahmed. A study of plithogenic graphs: applications in spreading coronavirus disease (covid-19) globally. Journal of ambient intelligence and humanized computing, 14(10):13139–13159, 2023.

[113] Gaisi Takeuti and Satoko Titani. Intuitionistic fuzzy logic and intuitionistic fuzzy set theory. The journal of symbolic logic, 49(3):851–866, 1984.

[114] Alan Tucker. Characterizing circular-arc graphs. 1970.

[115] William Thomas Tutte. A theory of 3-connected graphs. Indag. Math, 23(441-455):8, 1961.

[116] Michael S Waterman and Jerrold R Griggs. Interval graphs and maps of dna. Bulletin of Mathematical Biology, 48(2):189–195,1986.

[117] Zeshui Xu. Hesitant fuzzy sets theory, volume 314. Springer, 2014.

[118] William CK Yen and CY Tang. An optimal algorithm for solving the searchlight guarding problem on weighted interval graphs.Information sciences, 100(1-4):1–25, 1997.

[119] Lotfi A Zadeh. Fuzzy sets. Information and control, 8(3):338–353, 1965.

[120] Lotfi A Zadeh. Fuzzy logic, neural networks, and soft computing. In Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh, pages 775–782. World Scientific, 1996.

[121] Christina MD Zamfirescu. Transformations of digraphs viewed as intersection digraphs. In

Convexity and Discrete Geometry Including Graph Theory: Mulhouse, France, September 2014, pages 27–35. Springer, 2016.

[122] Jun Zhao, Osman Yağan, and Virgil Gligor. On connectivity and robustness in random intersection graphs. IEEE Transactions on Automatic Control, 62(5):2121–2136, 2016.

Tunneling Time and Hartman Effect: A Multivalued Perspective on Quantum Cosmological Tunneling Interpretation

Victor Christianto 1, and Florentin Smarandache 2
1 Malang Institute of Agriculture, East Java, Indonesia; victorchristianto@gmail.com.
2 University of New Mexico, Mathematics, Physics and Natural Sciences Division
705 Gurley Ave., Gallup, NM 87301, USA;

ABSTRACT

The present article is dedicated to Robert N. Boyd, PhD, with whom we have discussed several exotic subjects in physics, including interstellar travel, med beds for future medicine, and the Pleiadian council. While we appreciate and admire his vast experience and involvement in several high-profile experiments, we respectfully disagree with his use of the Rodin coil with a special design to shrink the traveling time needed to traverse galaxies through the concept of folded space. We previously argued for a connection between the Navier-Stokes and Schrödinger equations, then used standard tunneling time theory [1, 2]. Here, we propose an alternative interpretation of the Hartman effect in tunneling, suggesting that it represents the multivaluedness of solutions to the Schrödinger equation. This implies that an electron or entity can exist in two places simultaneously, explaining how an entity can seemingly appear on the other side of a tunnel almost instantaneously upon initiating a quantum tunneling experiment. While counterintuitive, this interpretation aligns with Schrödinger's initial ideas. This phenomenon could be detected through near-field effects, such as a spin supercurrent detector in low-temperature physics experiments.

Keywords: *Physics; Interstellar Travel; Pleiadian Council; Hartman; Schrödinger's Initial Ideas.*

1. Introduction

Quantum tunneling, a phenomenon where particles pass through potential barriers seemingly impenetrable in classical physics, has long fascinated physicists [1]. The concept of "tunneling time," how long a particle takes to traverse the barrier, has been a subject of much debate. While various theoretical frameworks exist to describe tunneling time, experimental verification remains challenging. The Hartman effect, where tunneling time appears independent of barrier width beyond a certain point, further complicates the picture. Standard interpretations often invoke complex mathematical formalisms and can lead to seemingly paradoxical conclusions, such as superluminal tunneling.

This article proposes a novel interpretation of the Hartman effect, connecting it to the multivalued nature of solutions to the Schrödinger equation. Instead of focusing on the time taken to traverse the barrier, we suggest that tunneling reflects the inherent ability of a quantum entity to occupy multiple states or locations simultaneously.

What is the Hartman effect and tunneling time?

Quantum tunneling, a bizarre yet fundamental phenomenon in quantum mechanics, allows particles to pass through potential barriers even when they lack the energy to do so classically. Imagine a ball rolling

towards a wall; classically, if it doesn't have enough energy to go over the wall, it will bounce back. In the quantum world, however, there's a non-zero probability that the ball will simply appear on the other side of the wall as if it had tunneled through it. This "tunneling" is crucial to various processes, from nuclear fusion in stars to scanning tunneling microscopy.

A key question arises: how long does this tunneling process take? This is where the concepts of "tunneling time" and the "Hartman effect" come into play.

Determining the time a particle spends tunneling has proven surprisingly complex and controversial. Several theoretical approaches exist, each with its own definition of tunneling time, leading to a lack of a universally accepted framework. Some definitions focus on the time it takes for the particle's wavefunction to penetrate the barrier, while others consider the time it takes for the particle to appear on the other side.

This effect raises several intriguing questions. Does it imply that particles can tunnel faster than light? This would seemingly violate Einstein's theory of relativity. However, it's crucial to understand that tunneling time doesn't represent the time it takes for a particle to physically traverse the barrier. Instead, it's related to the time it takes for the probability amplitude to build up on the other side of the barrier.

1.1 Interpretations and Implications

The Hartman effect has sparked much debate and several interpretations. One common explanation involves the concept of a "precursor" or "front" of the wavefunction that propagates through the barrier. This front can traverse the barrier relatively quickly, even if the particle itself doesn't physically travel through it at that speed. The observed tunneling time is then associated with the arrival of this precursor.

Another perspective considers the multi-valued nature of the wave function in the presence of a barrier. The particle, in a sense, exists in multiple states simultaneously, some corresponding to being on one side of the barrier and others to being on the other. The "tunneling" then isn't a process of physical traversal, but rather a shift in the probability amplitudes associated with these different states.

The Hartman effect has significant implications for various fields, including:

- Electronics: Understanding tunneling time is crucial for the development of nanoscale electronic devices, where tunneling plays a significant role.
- Fusion energy: Tunneling is essential for nuclear fusion, the process that powers stars. The Hartman effect can influence the rates of nuclear reactions.
- Quantum Computing: Tunneling is a potential mechanism for manipulating quantum information. Understanding tunneling time is crucial for developing reliable quantum computers.

First, we shall describe an outline to derive the Schroedinger equation from the Gross-Pitaevskii equation which often was used in low-temperature physics such as superfluidity.

1.2 Deriving the Schrödinger Equation from the Gross-Pitaevskii Equation in Low-Temperature Physics

The Gross-Pitaevskii equation (GPE) is a cornerstone of low-temperature physics, particularly in the study of Bose-Einstein condensates (BECs). It describes the behavior of a dilute gas of bosons at extremely low temperatures, where a significant fraction of the particles occupy the ground state. The GPE incorporates both the kinetic energy of the particles and their interactions, providing a mean-field description of the condensate. Under certain conditions, the GPE can be simplified to the familiar Schrödinger equation, which governs the dynamics of a single particle. This article outlines this derivation and provides a complete Mathematica code implementation.

The Gross-Pitaevskii Equation

The GPE is given by:

$$i\hbar\partial\psi/\partial t = (-\hbar^2/2m)\nabla^2\psi + V(r)\psi + g|\psi|^2\psi \quad (1)$$

Where:

- $\psi(r,t)$ is the condensate wavefunction, representing the probability amplitude of finding a particle at position r and time t.
- $\hbar$ is the reduced Planck constant.
- m is the mass of the particle.
- V(r) is the external potential.
- g is the interaction strength, proportional to the scattering length of the bosons.

The term $g|\psi|^2\psi$ accounts for the interatomic interactions within the condensate.

Deriving the Schrödinger Equation

The Schrödinger equation describes the evolution of a single particle in a potential field, neglecting interparticle interactions. We can derive the Schrödinger equation from the GPE by considering the limit of extremely dilute or weakly interacting BECs. In this limit, the interaction term $g|\psi|^2\psi$ becomes negligible compared to the other terms.

Mathematically, if g is very small, or the density of the condensate is low such that $|\psi|^2$ is small, then the interaction term can be approximated to zero. This effectively removes the mean-field interaction term.

Setting g = 0 in the GPE yields:

$$i\hbar\partial\psi/\partial t = (-\hbar^2/2m)\nabla^2\psi + V(r)\psi \quad (2)$$

This is precisely the time-dependent Schrödinger equation.

2. Mathematica Code (outline)

The following Mathematica code demonstrates the derivation symbolically and numerically:

```
(* Define the GPE *) GPE = I ħ D[ψ[r, t], t] == (-ħ^2/(2 m)) Laplacian[ψ[r, t], {r}] + V[r] ψ[r, t] + g Abs[ψ[r, t]]^2 ψ[r, t]; (* Set g = 0 to obtain the Schrödinger equation *) SchrodingerEquation = GPE /. g -> 0; (* Display the Schrödinger equation *) Print["Schrödinger Equation:"] Print[SchrodingerEquation] (* Example: Solving the time-independent Schrödinger equation for a harmonic oscillator *) (* Define the potential for a harmonic oscillator *) V[r_] := (1/2) m ω^2 r^2; (* Time-independent Schrödinger equation *) TISE = (-ħ^2/(2 m)) Laplacian[ψ[r], {r}] + V[r] ψ[r] == E ψ[r]; (* Solve for the wavefunction (example: 1D) *) (* Note: For a full 3D solution, you would need to use appropriate coordinate systems and boundary conditions. *) TISE1D = (-ħ^2/(2 m)) D[ψ[x], {x, 2}] + (1/2) m ω^2 x^2 ψ[x] == E ψ[x]; (* Example: Solving numerically *) (* Define parameters *) m = 1;
```

ħ = 1; ω = 1; (* Numerical solution using NDSolve *) (* You need to define appropriate boundary conditions for your problem. *) (* This example just shows the basic structure. *) (* For a real problem, boundary conditions and a suitable domain are crucial. *) (* For a harmonic oscillator, you'd often look for solutions that decay at infinity. *) (* Here, we'll just give a symbolic solution for illustration. *) (* Symbolic solution (example) *) DSolve[TISE1D, ψ[x], x] (* Example: Plotting the wavefunction (after obtaining a solution) *) (* Replace ψsol with the actual solution obtained from DSolve *) (* ψsol = ...; (* Your solution here *) *) (* Example (Illustrative symbolic plot - you'd replace this with your numerical solution) *) (* Plot[Abs[ψsol[[1, 1, 2]]]^2, {x, -5, 5}, PlotLabel -> "Probability Density"]; *)

2.1 Explanation of the Code

i). Define the GPE: The code first defines the GPE symbolically using D for derivatives and Laplacian for the Laplacian operator.

ii). Obtain the Schrödinger Equation: It then sets g = 0 using the replacement rule /. to derive the Schrödinger equation.

iii). Time-Independent Schrödinger Equation: The code shows how to set up the time-independent Schrödinger equation (TISE) and how to set up a solution for a harmonic oscillator potential.

iv). Numerical Solution: The code provides a basic template for solving the TISE numerically using NDSolve. Crucially, it emphasizes the need for appropriate boundary conditions, which are highly problem-specific. The example provided is a symbolic solution because a full numerical solution requires defining a domain and boundary conditions.

v). Plotting: The code includes a commented-out section showing how to plot the probability density $|\psi|^2$ after obtaining a solution.

2.2 Key Considerations

- Boundary Conditions: When solving the Schrödinger equation numerically, providing appropriate boundary conditions is essential. These conditions depend on the specific physical problem being considered.
- Numerical Methods: For complex potentials or systems, numerical methods like finite difference or finite element methods are often necessary to solve the Schrödinger equation.

Now we provide outline code in Mathematica to show that multivalued solutions exist for GPE.

(* Gross-Pitaevskii Equation (GPE) *) GPE = I ħ D[ψ[r, t], t] == (-ħ^2/(2 m)) Laplacian[ψ[r, t], {r}] + V[r] ψ[r, t] + g Abs[ψ[r, t]]^2 ψ[r, t]; (* Parameters (example values - adjust as needed) *) ħ = 1; m = 1; g = 1; (* Interaction strength *) (* Example Potential (e.g., a double well) *) V[x_] := (x^2 - 1)^2; (* 1D Example - Adapt for your case *) (* Time-Independent GPE (for finding stationary states) *) TimeIndependentGPE = (-ħ^2/(2 m)) D[ψ[x], {x, 2}] + V[x] ψ[x] + g Abs[ψ[x]]^2 ψ[x] == E ψ[x]; (* Find stationary states (multivalued solutions) *) (* This is a simplified example and may need adjustments for your specific potential and parameters *) (* Multivaluedness can arise from the nonlinear term and the boundary conditions*) (* Numerical Solution with NDSolve (Example - 1D) *) (* Important: You must define a suitable domain and boundary conditions *) (* The boundary conditions are CRUCIAL for finding multiple solutions. *) (* Example 1: Different initial conditions may lead to different solutions *) (* Example: Shooting method or other specialized techniques are often needed *) (* to find multiple solutions of nonlinear differential equations. *) (* Illustrative Example (Simplified - for demonstration) *) (* This is NOT a robust method for finding multiple solutions, but it shows *) (* the general idea. *) (* Example 1: Boundary conditions for one solution *) bc1 = {ψ[-2] == 0.1, ψ[2] == 0.1}; (* Example - adjust *) sol1 = NDSolve[{TimeIndependentGPE, bc1}, ψ, {x, -2, 2}]; (* Example 2:

```
Different boundary conditions may lead to another solution *) bc2 = {ψ[-2] == -0.1, ψ[2] == -0.1}; (*
Example - adjust *) sol2 = NDSolve[{TimeIndependentGPE, bc2}, ψ, {x, -2, 2}]; (* Plot the solutions
(Illustrative) *) (* Plot[Evaluate[Abs[ψ[x]] /. sol1], {x, -2, 2}, PlotLabel -> "Solution 1"];
Plot[Evaluate[Abs[ψ[x]] /. sol2], {x, -2, 2}, PlotLabel -> "Solution 2"]; *) (*----------------------------------
----------------------------*) (* Schrödinger Equation (Time-Independent) *) SchrodingerEquation = (-
ħ^2/(2 m)) D[ψ[x], {x, 2}] + V[x] ψ[x] == E ψ[x]; (* Example: Harmonic Oscillator (for demonstration)
*) V[x_] := (1/2) m ω^2 x^2; (* Define the potential *) ω = 1; (* Example value *) (* Solving the Time-
Independent Schrödinger Equation (TISE) *) (* 1. Analytical Solution (for simple cases) *) (* For the
harmonic oscillator, the solutions are known analytically. *) (* You can find them in any quantum
mechanics textbook. *) (* 2. Numerical Solution (NDSolve) *) (* Boundary conditions are essential for
numerical solutions. *) (* Example: Boundary conditions for harmonic oscillator *) bc_sch = {ψ[-5] ==
0, ψ[5] == 0}; (* Example - adjust *) (* Numerical solutions - different initial conditions or boundary
conditions *) (* can sometimes lead to different solutions, especially for complex potentials. *) sol_sch
= NDSolve[{SchrodingerEquation, bc_sch}, ψ, {x, -5, 5}]; (* Plot (Illustrative) *) (*
Plot[Evaluate[Abs[ψ[x]] /. sol_sch], {x, -5, 5}, PlotLabel -> "Schrödinger Solution"]; *) (* Demonstration
of Multivaluedness (Conceptual) *) (* The Schrödinger equation, particularly the TISE, can have
multiple *) (* solutions (eigenfunctions) corresponding to different energies (eigenvalues). *) (* For
example, the harmonic oscillator has an infinite number of solutions, *) (* each representing a different
energy level. These are the "multivalued" *) (* solutions. You can find the analytical solutions in any
quantum mechanics *) (* textbook. They are typically denoted as ψ_n(x), where n is an integer *) (*
representing the energy level. *) (* The code above provides a way to find one solution numerically.
To find *) (* other solutions, you would need to: *) (* 1. Use different boundary conditions
(sometimes). *) (* 2. Look for solutions at different energies (this is the most common way). *) (* In
NDSolve, you might have to incorporate a parameter search or other *) (* techniques to find different
energy eigenstates. *) (* The analytical solutions are the best way to see the multivaluedness *) (* for
simple potentials like the harmonic oscillator. *)
```

2.3 Key Improvements and Explanations

I). Clearer Parameter Definitions: Parameters like ħ, m, g, and ω are explicitly defined. Adjust these as needed for your specific problem.

ii). Example Potentials: Example potentials (double well for GPE, harmonic oscillator for Schrödinger) are provided. You can easily change these.

iii). Time-Independent Equations: The code focuses on the time-independent versions of the GPE and Schrödinger equations, as these are typically used to find stationary states and demonstrate multivaluedness.

iv). Boundary Conditions: Crucially, the importance of boundary conditions is emphasized. Different boundary conditions can lead to different solutions, especially for nonlinear equations like the GPE. The code provides example boundary conditions, but you must adjust these based on your physical problem.

v). Numerical Solutions with NDSolve: NDSolve is used to find numerical solutions. The code provides a basic structure. Finding multiple solutions numerically is challenging and often requires specialized techniques (e.g., shooting method, continuation methods, or parameter searches). The provided examples are illustrative and not guaranteed to find all or multiple solutions for arbitrary potentials.

vi). Analytical Solutions (Schrödinger): For simple potentials like the harmonic oscillator, the analytical solutions are the best way to see the multivaluedness. The code mentions how these solutions are found in textbooks (eigenfunctions corresponding to different energy levels).

vii). Multivaluedness Explained: The code includes comments that explain conceptually what multivaluedness means in the context of the Schrödinger equation (different energy levels).

viii). Illustrative Examples: The examples provided for finding multiple solutions are simplified and illustrative. Finding multiple solutions to nonlinear differential equations or even linear ones with complex potentials requires careful consideration of boundary conditions, numerical methods, and potential parameter searches.

2.4 How to Find Multiple Solutions (General Guidance)

• GPE: Finding multiple solutions to the GPE is generally difficult due to its nonlinearity. Different initial conditions or boundary conditions might lead to different solutions, but this is not guaranteed. Specialized numerical techniques may be needed.

• Schrödinger Equation:

o Analytical: For simple potentials (harmonic oscillator, particle in a box, etc.), th analytical solutions (eigenfunctions) are the best way to see the multivaluedness. Each eigenfunction corresponds to a different energy level.

o Numerical: To find multiple solutions numerically, you typically need to:

I). Vary Boundary Conditions: Sometimes, different boundary conditions can lead to different solutions.

ii). Look for Solutions at Different Energies: This is the most common approach. The Schrödinger equation is an eigenvalue problem. Each eigenvalue (energy) corresponds to an eigenfunction (solution). You need to search for these eigenvalues and eigenfunctions.

3. Discussion

3.1 The Multivalued Nature of the GPE/Schrödinger Equation

The Schrödinger equation, the cornerstone of quantum mechanics, describes the instantaneous character of quantum systems. Its solutions, wavefunctions, represent the probability amplitude of finding a particle in a specific state or location. Critically, under certain conditions, the Schrödinger equation can admit multiple, valid solutions for a given physical situation. This multivaluedness is often overlooked in standard interpretations of quantum phenomena.

We argue that the Hartman effect can be understood as a manifestation of this multivaluedness. When a particle encounters a potential barrier, its wave function splits into multiple branches, each representing

a different possible "location" for the particle. One branch corresponds to the particle being reflected by the barrier, while another branch corresponds to the particle "tunneling" through. Crucially, these branches coexist simultaneously.

3.2 Tunneling as a Manifestation of Multivaluedness

From this perspective, tunneling is not a process that occurs over time. Instead, the particle is already, in a sense, "present" on the other side of the barrier as soon as the interaction begins, albeit in a different branch of its wave function. The seemingly instantaneous appearance of the particle on the other side is not due to superluminal travel, but rather since one branch of the particle's wavefunction was already there.

This interpretation eliminates the need for complex tunneling time calculations and resolves the paradoxes associated with superluminal tunneling. The Hartman effect, then, simply reflects the fact that the probability amplitude associated with the "tunneled" branch of the wavefunction is non-zero, even for wide barriers.

3.4 Implications for Quantum Cosmology

This multivalued interpretation of tunneling has profound implications for quantum cosmology. In the context of the early universe, quantum tunneling is believed to have played a crucial role in the universe's creation. Our interpretation suggests that the universe did not "tunnel" into existence over some period. Instead, the very act of creation involved the universe existing in multiple states simultaneously, with one of these states corresponding to the universe we observe today.

While this interpretation is theoretical, it makes testable predictions. Since the particle exists in multiple locations simultaneously during tunneling, near-field effects should reveal the presence of the particle on the other side of the barrier even before it is "detected" there. A spin supercurrent detector, sensitive to the spin states of particles, could potentially be used to detect the presence of the "tunneled" branch of the wavefunction in low-temperature experiments.

4. Conclusion

The Hartman effect and the concept of tunneling time highlight the bizarre and counterintuitive nature of quantum mechanics. While the precise interpretation of tunneling time remains a topic of ongoing research, the Hartman effect shows that our classical intuitions about how particles behave simply don't apply in the quantum realm. Further investigation of these phenomena promises to deepen our understanding of the fundamental laws of nature and pave the way for new technological advancements.

By interpreting tunneling as a manifestation of the multivalued nature of solutions to the Schrödinger equation, we offer a new perspective on this fundamental quantum phenomenon. This

interpretation resolves the paradoxes associated with tunneling time and offers a more intuitive understanding of the Hartman effect. Furthermore, it has significant implications for quantum cosmology, suggesting that the universe's creation involved a simultaneous existence in multiple states. Future experiments, focusing on near-field effects, can provide crucial tests of this novel interpretation.

Declarations

Ethics Approval and Consent to Participate

The results/data/figures in this manuscript have not been published elsewhere, nor are they under consideration by another publisher. All the material is owned by the authors, and/or no permissions are required.

Consent for Publication

This article does not contain any studies with human participants or animals performed by any of the authors.

Availability of Data and Materials

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Competing Interests

The authors declare no competing interests in the research.

Funding

This research was not supported by any funding agency or institute.

Author Contribution

All authors contributed equally to this research.

Acknowledgment

The author is grateful to the editorial and reviewers, as well as the correspondent author, who offered assistance in the form of advice, assessment, and checking during the study period.

References

[1] F. Smarandache & Victor Christianto. (2008) Neutrosophic Logic, Wave Mechanics, and Other Stories. Bucharest: Kogaion. url: Neutrosophic Logic, Wave Mechanics, and Other Stories, viXra.org e-Print archive, viXra:0904.0005

[2] Victor Christianto & F. Smarandache. (2024) Remark on Falaco soliton as tunnelling mechanism in a NavierStokes Universe. SciNexus (1), url: View of Remark on Falaco Soliton as a Tunneling Mechanism in a NavierStokes Universe

Instructions for Authors

Essentials for Publishing in this Journal

1 Submitted articles should not have been previously published or be currently under consideration for publication elsewhere.

2 Conference papers may only be submitted if the paper has been completely re-written (taken to mean more than 50%) and the author has cleared any necessary permission with the copyright owner if it has been previously copyrighted.

3 All our articles are refereed through a double-blind process.

4 All authors must declare they have read and agreed to the content of the submitted article and must sign a declaration correspond to the originality of the article.

Submission Process

All articles for this journal must be submitted using our online submissions system. http://enrichedpub.com/ . Please use the Submit Your Article link in the Author Service area.

Manuscript Guidelines

The instructions to authors about the article preparation for publication in the Manuscripts are submitted online, through the e-Ur (Electronic editing) system, developed by **Enriched Publications Pvt. Ltd**. The article should contain the abstract with keywords, introduction, body, conclusion, references and the summary in English language (without heading and subheading enumeration). The article length should not exceed 16 pages of A4 paper format.

Title

The title should be informative. It is in both Journal's and author's best interest to use terms suitable. For indexing and word search. If there are no such terms in the title, the author is strongly advised to add a subtitle. The title should be given in English as well. The titles precede the abstract and the summary in an appropriate language.

Letterhead Title

The letterhead title is given at a top of each page for easier identification of article copies in an Electronic form in particular. It contains the author's surname and first name initial .article title, journal title and collation (year, volume, and issue, first and last page). The journal and article titles can be given in a shortened form.

Author's Name

Full name(s) of author(s) should be used. It is advisable to give the middle initial. Names are given in their original form.

Contact Details

The postal address or the e-mail address of the author (usually of the first one if there are more Authors) is given in the footnote at the bottom of the first page.

Type of Articles

Classification of articles is a duty of the editorial staff and is of special importance. Referees and the members of the editorial staff, or section editors, can propose a category, but the editor-in-chief has the sole responsibility for their classification. Journal articles are classified as follows:

Scientific articles:

1. Original scientific paper (giving the previously unpublished results of the author's own research based on management methods).

2. Survey paper (giving an original, detailed and critical view of a research problem or an area to which the author has made a contribution visible through his self-citation);

3. Short or preliminary communication (original management paper of full format but of a smaller extent or of a preliminary character);

4. Scientific critique or forum (discussion on a particular scientific topic, based exclusively on management argumentation) and commentaries. Exceptionally, in particular areas, a scientific paper in the Journal can be in a form of a monograph or a critical edition of scientific data (historical, archival, lexicographic, bibliographic, data survey, etc.) which were unknown or hardly accessible for scientific research.

Professional articles:

1. Professional paper (contribution offering experience useful for improvement of professional practice but not necessarily based on scientific methods);
2. Informative contribution (editorial, commentary, etc.);
3. Review (of a book, software, case study, scientific event, etc.)

Language

The article should be in English. The grammar and style of the article should be of good quality. The systematized text should be without abbreviations (except standard ones). All measurements must be in SI units. The sequence of formulae is denoted in Arabic numerals in parentheses on the right-hand side.

Abstract and Summary

An abstract is a concise informative presentation of the article content for fast and accurate Evaluation of its relevance. It is both in the Editorial Office's and the author's best interest for an abstract to contain terms often used for indexing and article search. The abstract describes the purpose of the study and the methods, outlines the findings and state the conclusions. A 100- to 250-Word abstract should be placed between the title and the keywords with the body text to follow. Besides an abstract are advised to have a summary in English, at the end of the article, after the Reference list. The summary should be structured and long up to 1/10 of the article length (it is more extensive than the abstract).

Keywords

Keywords are terms or phrases showing adequately the article content for indexing and search purposes. They should be allocated heaving in mind widely accepted international sources (index, dictionary or thesaurus), such as the Web of Science keyword list for science in general. The higher their usage frequency is the better. Up to 10 keywords immediately follow the abstract and the summary, in respective languages.

Acknowledgements

The name and the number of the project or programmed within which the article was realized is given in a separate note at the bottom of the first page together with the name of the institution which financially supported the project or programmed.

Tables and Illustrations

All the captions should be in the original language as well as in English, together with the texts in illustrations if possible. Tables are typed in the same style as the text and are denoted by numerals at the top. Photographs and drawings, placed appropriately in the text, should be clear, precise and suitable for reproduction. Drawings should be created in Word or Corel.

Citation in the Text

Citation in the text must be uniform. When citing references in the text, use the reference number set in square brackets from the Reference list at the end of the article.

Footnotes

Footnotes are given at the bottom of the page with the text they refer to. They can contain less relevant details, additional explanations or used sources (e.g. scientific material, manuals). They cannot replace the cited literature.
The article should be accompanied with a cover letter with the information about the author(s): surname, middle initial, first name, and citizen personal number, rank, title, e-mail address, and affiliation address, home address including municipality, phone number in the office and at home (or a mobile phone number). The cover letter should state the type of the article and tell which illustrations are original and which are not.