Unveiling Big Data Insights: A Neutrosophic Classification Approach for Enhanced Prediction with Machine Learning

Ahmed Z.M. Elsherif [1], A. A. Salama, [1]O. M. Khaled [1], Mostafa Herajy [1], E. I. Elsedimy[2], Huda E. Khalid[3], Ahmed K. Essa[3]

[1]Department of Mathematics and Computer Science, Faculty of Science, Port Said University, Port Said, Egypt.

ahmad.elsherif@sci.psu.edu.eg, drsalama44@gmail.com, ahmed_salama_2000@sci.psu.edu.eg

[2] Department of Information Technology Management, Faculty of Management Technology and Information Systems, Port Said University, Port Said, Egypt

[3] University of Telafer, The Administration Assistant for the President of the Telafer University, Telafer, Iraq; https://orcid.org/0000-0002-0968-5611 , dr.huda-ismael@uotelafer.edu.iq, ahmed.k.essa@uotelafer.edu.iq

*Correspondence: dr.huda-ismael@uotelafer.edu.iq

Abstract: The ever-growing volume and complexity of Big Data pose challenges for traditional classification tasks. This paper explores the potential of Neutrosophic Sets (NS), a powerful framework for handling uncertainty, in building robust classification models for Big Data prediction using Machine Learning (ML) techniques. We provide a detailed background on NS and discuss its advantages over Fuzzy Sets. We then propose a methodology that integrates NS with relevant ML algorithms for classification. We evaluate the performance of our Neutrosophic-based model on a Big Data source. The results are analyzed to assess the effectiveness of the Neutrosophic approach for Big Data prediction. This research contributes to the advancement of uncertainty management in Big Data classification and paves the way for further exploration of Neutrosophic-based ML models for various prediction tasks. Results show that the Neutrosophic Neural Networks (NNs) model achieved commendable performance across various metrics, with an accuracy of 79.08%, precision of 74.58%, recall of 77.64%, and an F1-score of 75.63%. These metrics indicate that the Neutrosophic NNs model effectively balances the trade-offs between precision and recall, providing a robust classification performance in the context of the evaluated dataset.

Keywords: Big Data, Neutrosophic Sets, Machine Learning, Classification, Prediction, Uncertainty Management, Fuzzy Sets

1. Introduction

The exponential growth of data, often referred to as Big Data, presents both tremendous opportunities and significant challenges [1, 2]. While vast data volumes hold the key to uncovering valuable insights, extracting meaningful patterns for classification tasks can be a complex endeavor [1]. Traditional classification algorithms often struggle with inherent uncertainties and inconsistencies within Big Data [3]. This paper explores the potential of Neutrosophic Sets (NS) as a novel approach to address these challenges in Big Data classification. NS offers a powerful mathematical framework for handling uncertainty by incorporating three distinct truth membership functions: truth (T), indeterminacy (I), and falsity (F) [4]. This stands in contrast to traditional Fuzzy Sets (FS), which only utilize truth membership degrees [5]. By introducing indeterminacy, NS allows for a more nuanced representation of data points that may exhibit ambiguity or partial truth [6].

Machine Learning (ML) has emerged as a powerful tool for building predictive models that can learn from data and make accurate classifications. However, conventional ML techniques often rely on crisp data with well-defined categories [7]. Here, we propose to leverage the capabilities of NS within the realm of ML.

The primary objective of this research is to investigate the effectiveness of Neutrosophic-based classification models for Big Data prediction using Machine Learning techniques. We aim to explore how NS can be integrated with existing ML algorithms to improve classification accuracy and handle the inherent uncertainties present within Big Data. Through this research, we contribute to the advancement of uncertainty management in Big Data classification and pave the way for further exploration of Neutrosophic-based ML models for various prediction tasks.

In this study, we leveraged advanced machine learning models to significantly enhance liver disease prediction, a critical step towards improving patient care and saving lives. Utilizing models such as Neutrosophic NNs, we navigated the complexities of big data through rigorous preprocessing, feature extraction, and predictive analysis. The Neutrosophic NNs model demonstrated commendable performance with an accuracy of 79.08%, precision of 74.58%, recall of 77.64%, and an F1-Score of 75.63%. These results underscore the potential of Neutrosophic NNs in effectively predicting liver disease, highlighting the importance of robust data processing and analysis in healthcare applications.

This paper is structured as follows: Section 2 outlines the methods and stages employed to using Neutrosophic NNs to convert traditional data into Neutrosophic data set, then analyzing and predicting using the NNS model and its enormous capabilities in classification. Section 3 details the experimental results obtained using the Python environment. Finally, Section 4 concludes with a summary of the main findings and provides useful suggestions for future research directions.

2. Background:

This section delves into the theoretical foundations of our research, exploring Neutrosophic Sets (NS) and their potential integration with Machine Learning (ML) for Big Data classification.

2.1 Neutrosophic Sets

Neutrosophic Sets Introduced by Florentin Smarandache [4], Neutrosophic Sets (NS) offer a powerful framework for representing and managing uncertainty. NS generalizes the concept of Fuzzy Sets (FS) by incorporating three distinct truth membership functions [5]:

- Truth (T): Denotes the degree to which an element belongs to a set. Values range from 0 (completely false) to 1 (completely true).

- Indeterminacy (I): Represents the degree of uncertainty or lack of knowledge about an element's membership. Values range from 0 (completely determinate) to 1 (completely indeterminate) [4].
- Falsity (F): Denotes the degree to which an element does not belong to a set. Values range from 0 (completely true) to 1 (completely false) [1]. For any element x in a neutrosophic set A, these memberships are denoted as T(x), I(x), and F(x), satisfying the condition: $0 \leq T(x) + I(x) + F(x) \leq 3$ [6]. NS operations like union, intersection, complement, etc., are defined using various operators on the truth, indeterminacy, and falsity membership functions. These operations allow for more nuanced representation of complex data points compared to traditional sets.

2.2 Neutrosophic Sets vs. Fuzzy Sets

While Fuzzy Sets (FS) have been instrumental in handling data with varying degrees of truth membership, they are limited in their ability to represent situations with inherent ambiguity or partial truth [5]. NS addresses this limitation by introducing the concept of indeterminacy [6]. This allows data points to belong to a set, be indeterminate about their membership, and not belong to the set, all to varying degrees. This richer representation of uncertainty is particularly valuable for Big Data, where data quality and inherent inconsistencies are often prevalent [3].

2.3 Machine Learning for Classification

Machine Learning (ML) offers a suite of algorithms that can learn from data and make accurate predictions. In the context of classification, these algorithms learn to identify patterns within labeled data and then use those patterns to classify new, unseen data points [7-11,25-30]. Several well-established ML algorithms are suitable for classification tasks, including:

Neural Networks (NNs): Neural networks consist of interconnected nodes organized in layers, allowing for complex pattern recognition and decision-making tasks. They can accommodate neutrosophic data by incorporating activation functions that account for truth, indeterminacy, and falsity membership values within the network architecture [7].

3. Literature Review

The application of Neutrosophic Sets (NS) for classification tasks has gained increasing attention in recent years. This section reviews existing research to understand the current state of the art, focusing on its potential for Big Data classification. Several studies have explored the use of NS for classification in various domains, including image processing, medical diagnosis, and pattern recognition [12- 15]. These studies highlight several advantages of NS:

Improved Uncertainty Handling: Compared to Fuzzy Sets, NS allows for a more nuanced representation of uncertainty by incorporating the concept of indeterminacy [4]. This is particularly beneficial for real-world data, where ambiguity and partial truth are often present [3].

Enhanced Classification Accuracy: Studies have shown that Neutrosophic-based classification models can achieve higher accuracy compared to traditional methods, especially when dealing with noisy or incomplete data [13, 16].

Flexibility: NS can be integrated with various Machine Learning (ML) algorithms for classification tasks [7]. This allows researchers to leverage the strengths of existing ML techniques while incorporating uncertainty management through NS.

However, limitations and open challenges also need to be addressed:

Limited Applications with Big Data: While research on Neutrosophic classification is growing, there are relatively few studies focusing specifically on Big Data applications [3, 17-21]. Further research is needed to explore the scalability and efficiency of NS-based models when dealing with large and complex datasets [23-28].

Computational Complexity: Neutrosophic operations and computations can be more complex compared to Fuzzy Sets [4]. This can pose challenges when dealing with Big Data, where processing speed and resource efficiency are crucial [16].

Standardization and Optimization: Further research is needed to develop standardized methods for neutrosophication (conversion of traditional data into NS) and optimization of NS-based classification algorithms for Big Data settings [17].

Despite these limitations, the existing research demonstrates the potential of NS for Big Data classification. By addressing the identified challenges and exploring new applications, NS can be a valuable tool in unlocking valuable insights from vast and complex datasets. While there is a limited pool of research focusing specifically on Big Data applications of Neutrosophic Sets for classification, some studies explore the potential benefits and challenges in this context. For instance, Amin et al. [16] discuss the applicability of NS in Big Data healthcare analytics, highlighting its ability to handle inherent uncertainties in medical data. Similarly, Yasser et al. [15] propose a Neutrosophic classifier framework for COVID-19 classification using chest X-ray images, demonstrating its potential for handling noisy or incomplete medical imagery data, often encountered in Big Data settings. These initial explorations pave the way for further research on integrating NS with scalable Big Data processing techniques and resource-efficient ML algorithms.

4. Proposed Methodology:

This section outlines the methodology for exploring Neutrosophic-based classification models for Big Data prediction using Machine Learning techniques.

4.1 Machine Learning Algorithm Selection

We propose utilizing NNs as the primary Machine Learning algorithm for our Neutrosophic classification model. NNs excel at learning intricate patterns and relationships within data, making them well suited for handling the inherent uncertainties represented by the truth, indeterminacy, and falsity membership functions in Neutrosophic Sets. Through their adaptive learning capabilities, NNs can effectively discern optimal decision boundaries, facilitating the segregation of data points belonging to different classes. This inherent flexibility and adaptability make neural networks a suitable choice for Neutrosophic classification tasks.

4.2 Handling Missing Data:

Given the presence of missing values within our dataset, we employed median imputation to address this issue. The choice of median imputation was due to its robustness against outliers, which is particularly important in clinical datasets where outliers can represent significant anomalies. The mathematical formulation for median imputation is straightforward but effective [21-27]:

$$m = median(D_i) \qquad (1)$$

Where Direpresents the set of non-missing values for biomarker i. Each missing value in biomarker i was replaced with mm, ensuring a complete dataset for further processing.

4.3 Data Normalization:

To prepare the data for Neutrosophic transformation and neural network modeling, we applied data normalization. This process adjusted the scale of the data features, enabling more effective learning by the neural network models. The normalization was conducted using the Min-Max scaling technique [22], defined by:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \qquad (2)$$

Where x is the original value, and x' is the normalized value. This scaling ensured that all input features contributed equally to the analysis, preventing features with larger scales from dominating the learning process.

4.4 Outlier Detection and Treatment:

We also implemented a method to detect and treat outliers, which could skew the results of our analysis. Using the Interquartile Range (IQR) method, outliers were identified and treated. The IQR is calculated as [23]:

$$\text{IQR}=\text{Q3}-\text{Q1} \qquad (3)$$

Where Q1 and Q3 are the 25th and 75th percentiles, respectively. Values falling below $Q1-1.5\times\text{IQR}$ or above $Q3+1.5\times\text{IQR}$ were considered outliers and were capped or floored accordingly to reduce their impact.

These preprocessing steps ensured that the data was clean, normalized, and ready for the subsequent transformation into Neutrosophic sets and analysis via neural network modeling.

4.5 Big Data Representation with Neutrosophic Sets

To leverage NS for Big Data classification, we will employ a neutrosophication technique to convert traditional data points into neutrosophic values. Here is a possible approach:

1. **Feature Scaling:** Standardize all data features within a specific range (e.g., 0-1) to ensure consistent membership function application.
2. **Membership Function Definition**: Define appropriate membership functions for truth (T), indeterminacy (I), and falsity (F) based on the characteristics of the data and the classification task. These functions can be designed using various mathematical expressions or through expert knowledge.
3. **Neutrosophic Data Generation**: Apply the defined membership functions to each data point, resulting in truth, indeterminacy, and falsity membership values for each feature. This transforms the traditional data into a neutrosophic representation suitable for the NNs algorithm. To

enrich the analytical depth, the dataset underwent a transformation into Neutrosophic sets. This process involved converting each biomarker's value into three components based on Neutrosophic logic [24]:

- Truth (T): Represents how true the proposition of normalcy is.
- Falsehood (F): Represents how false this proposition is.
- Indeterminacy (I): Captures the indeterminacy or uncertainty.

These components were calculated using the equations:

$$T(X) = \max(0, \min(1, \frac{x-l}{u-l})) \quad (4)$$

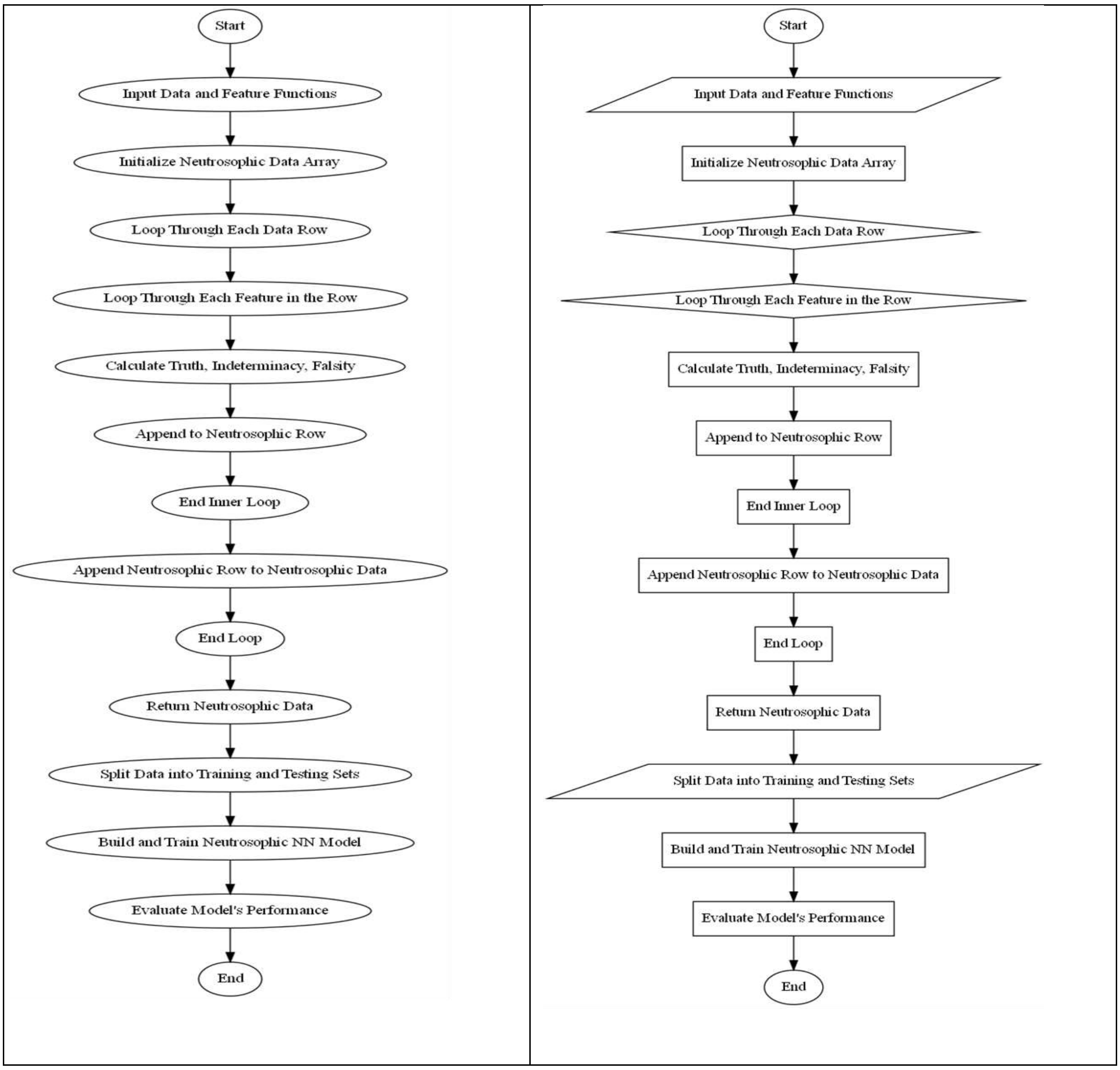
$$F(X) = \max(0, \min(1, \frac{u-x}{u-l})) \quad (5)$$

$$I(X) = I - T(x) - F(x) \quad (6)$$

Where x is the observed biomarker value, and L and U are the predefined lower and upper bounds of the normal clinical range, respectively.

4.6 Flowchart for Neutrosophic Data Transformation and Neural Network Classification

Below is the flowchart for the Neutrosophic Data Transformation and Neural Network Classification process:


Start
Input Data and Feature Functions
Initialize Neutrosophic Data Array
Loop Through Each Data Row
Loop Through Each Feature in the Row
Calculate Truth, Indeterminacy, Falsity
Append to Neutrosophic Row
End Inner Loop
Append Neutrosophic Row to Neutrosophic Data
End Loop
Return Neutrosophic Data
Split Data into Training and Testing Sets
Build and Train Neutrosophic NN Model
Evaluate Model's Performance
End
Start
Input Data and Feature Functions
Initialize Neutrosophic Data Array
Loop Through Each Data Row
Loop Through Each Feature in the Row
Calculate Truth, Indeterminacy, Falsity
Append to Neutrosophic Row
End Inner Loop
Append Neutrosophic Row to Neutrosophic Data
End Loop
Return Neutrosophic Data
Split Data into Training and Testing Sets
Build and Train Neutrosophic NN Model
Evaluate Model's Performance
End

Input Data and Feature Functions: Load the liver disease patient data and the list of feature transformation functions.

Initialize Neutrosophic Data Array: Create an empty array to store Neutrosophic data.

Loop Through Each Data Row: Iterate over each record in the dataset.

Loop Through Each Feature in the Row: For each feature in the record, calculate Neutrosophic values.

Calculate Truth, Indeterminacy, Falsity: Apply transformation functions to obtain truth, indeterminacy, and falsity values for each feature.

Append to Neutrosophic Row: Store the calculated values in the Neutrosophic row.

End Inner Loop: Complete processing for all features in the current row.

Append Neutrosophic Row to Neutrosophic Data: Add the completed Neutrosophic row to the Neutrosophic data array.

End Loop: Complete processing for all records in the dataset.

Return Neutrosophic Data: The dataset is now transformed into its Neutrosophic representation.

Split Data into Training and Testing Sets: Divide the Neutrosophic data into training and testing subsets.

Build and Train Neutrosophic NN Model: Construct and train a neural network model using the training data.

Evaluate Model's Performance: Assess the trained model's performance using accuracy, precision, recall, and F1-score metrics on the testing data.

End

4.7 Python Code for Neutrosophication of Customer Purchase Features

Algorithm 1 Neutrosophic Data

```
function NEUTROSOPHIFY(data, feature_funcs)
    neutrosophic_data ← []
    for i = 1 to length(data) do
        row ← data.iloc[i]
        neutrosophic_row ← []
        for feature_idx, feature in enumerate(data.columns) do
            truth,   indeterminacy,   falsity ← feature_funcs[feature_idx](row[feature])
            neutrosophic_row.append((truth, indeterminacy, falsity))
        end for
        neutrosophic_data.append(neutrosophic_row)
    end for
    return neutrosophic_data
end function
```

Algorithm 1 outlines the NEUTROSOPHIFY function, which transforms traditional data into a Neutrosophic representation. It accepts a dataset (data) and a list of transformation functions (feature_funcs) for each feature. The algorithm initializes an empty list, neutrosophic_data, and iterates over each record, extracting rows and initializing neutrosophic_row for Neutrosophic values. For each feature, it applies the corresponding transformation function to compute truth, indeterminacy, and falsity values, which are appended to neutrosophic_row. After processing all features, the completed neutrosophic_row is appended to neutrosophic_data. The function returns neutrosophic_data, effectively converting the dataset into a form that better manages uncertainty and indeterminacy, thereby enhancing the robustness and accuracy of subsequent analyses or machine learning models.

4.8 Neutrosophic Data Transformation and Neutrosophic NNs Classification Framework

```
import numpy as np
```

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from keras.models import Sequential
from keras.layers import Dense
def neutrosophify(data, feature_funcs):
    """
    Transforms traditional data into Neutrosophic representation.
    Args:
        data (pandas.DataFrame): DataFrame containing liver disease patient data.
        feature_funcs (list): List of functions for calculating T, I, F values for each feature.
    Returns:
        numpy.ndarray: Neutrosophic dataset with T, I, F values for each feature and patient.
    """
    neutrosophic_data = []
    for i in range(len(data)):
        row = data.iloc[i]
        neutrosophic_row = []
        for feature_idx, feature in enumerate(data.columns):
            # Calculate T, I, F values using provided functions
            truth, indeterminacy, falsity = feature_funcs[feature_idx](row[feature])
            neutrosophic_row.extend([truth, indeterminacy, falsity])
        neutrosophic_data.append(neutrosophic_row)
    return np.array(neutrosophic_data)
def build_neutrosophic_nn(input_dim):
    Builds a Neutrosophic Neural Network model for classification.
    Args:
        input_dim (int): Dimension of the input features.
    Returns:
        Sequential: Compiled Neutrosophic NN model.
    model = Sequential()
    model.add(Dense(64, input_dim=input_dim, activation='relu'))
    model.add(Dense(32, activation='relu'))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
def evaluate_model(model, X_test, y_test):
    Evaluates the performance of the trained model.
    Args:
        model (Sequential): Trained Neutrosophic NN model.
        X_test (numpy.ndarray): Testing data (Neutrosophic features).
        y_test (numpy.ndarray): Testing labels.
    Prints:
        Accuracy, Precision, Recall, F1-Score
    y_pred = (model.predict(X_test) > 0.5).astype("int32")
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    print("Accuracy:", accuracy)
    print("Precision:", precision)
    print("Recall:", recall)
    print("F1-Score:", f1)
# Example usage (replace with your actual data loading and feature functions)
# Assuming you have your liver disease patient data loaded into a pandas DataFrame 'data'
# and target labels for liver disease prediction ('y')
# Define membership function functions (replace placeholders with your logic)
def age_truth(age, max_age):
    return age / max_age
def bilirubin_truth(bilirubin, max_bilirubin):
    return bilirubin / max_bilirubin
def enzyme_indeterminacy(enzyme, max_enzyme):
    return enzyme / (max_enzyme + 1)  # Normalize between 0 and 1

def protein_falsity(protein, max_protein):
    return protein / max_protein
# Feature function list (order should match data columns)
feature_funcs = [age_truth, bilirubin_truth, enzyme_indeterminacy, protein_falsity]
# Neutrosophify the data
neutrosophic_data = neutrosophify(data, feature_funcs)
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(neutrosophic_data, y, test_size=0.2, random_state=42)
```

```
# Build and train the Neutrosophic NN model
model = build_neutrosophic_nn(X_train.shape[1])
model.fit(X_train, y_train, epochs=50, batch_size=32, validation_data=(X_test, y_test))
# Evaluate the model's performance
evaluate_model(model, X_test, y_test)
```

The implementation of the proposed Neutrosophic Neural Network (NN) model for liver disease classification involves several crucial steps, starting with the transformation of traditional data into a Neutrosophic representation. The neutrosophify function processes the dataset by applying feature-specific transformation functions to calculate the truth, indeterminacy, and falsity values for each feature, resulting in a comprehensive Neutrosophic dataset. This transformed data is then split into training and testing sets. The build_neutrosophic_nn function constructs a neural network model with an input layer matching the dimensionality of the transformed features, followed by two hidden layers and an output layer configured for binary classification. The model is compiled with the Adam optimizer and binary cross-entropy loss function. After training the model with the training dataset for 50 epochs, the evaluate_model function assesses its performance using accuracy, precision, recall, and F1-score metrics on the testing set. This methodology effectively leverages Neutrosophic logic to handle uncertainty and imprecision in the data, enhancing the robustness and accuracy of the liver disease classification.

5. Discussion of Results

To assess the efficacy of our model, we conducted comprehensive testing and evaluation using the Liver Disease Patient Dataset transformed into Neutrosophic data. The following section provides a detailed overview of the developed framework, its components, and the implementation tools used. It then presents a thorough description of the dataset employed in the experiment, highlighting its transformation from traditional big data to Neutrosophic data to better manage uncertainty and imprecision. This transformation is followed by a discussion of the experimental setup and operational procedures. Additionally, this section outlines the evaluation criteria and benchmarks used in our experiment, providing a general overview before delving into a detailed analysis of the data.

5.1 Implementation Details

This subsection presents the implementation specifics of the proposed model, which is crucial to assess the feasibility, quality, and efficiency of our work. To achieve this objective, the proposed approach was executed on a laptop equipped with an Intel (R) CoreTM i7-9850H CPU operating at 2.6 GHz and 32 GB of RAM, running on the Windows 11 x64 operating system. For application development, the Python programming language was selected due to its extensive libraries and capabilities. The experiments were conducted using Jupyter Notebook within the Anaconda Navigator, involving creating, compiling, and building the code. The classification and assessment procedures were carried out using the sklearn package, while data processing tasks were handled using the Panda data frame. Data processing and visualization tasks were performed using the

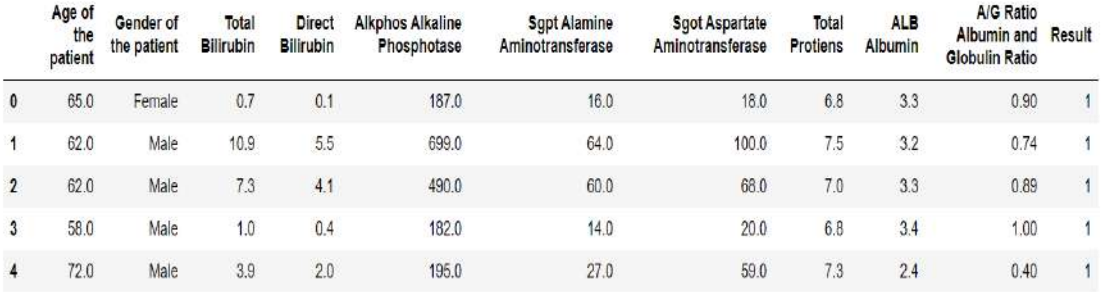
Matplotlib and NumPy libraries. The transformation of traditional data into Neutrosophic data was also facilitated by these tools and the addition of the necessary Python codes and equations, enhancing the model's ability to manage uncertainty and improve predictive accuracy.

5.2 Dataset

The experiments were conducted using the Liver Disease Patient Dataset, which includes 30,000 training data records. This dataset comprises ten variables: Age of the patient, Gender of the patient, Total Bilirubin, Direct Bilirubin, Alkphos Alkaline Phosphatase, Sgpt Alamine Aminotransferase, Sgot Aspartate Aminotransferase, Total Proteins, Albumin, and Albumin and Globulin Ratio, along with a classification field where the labels (assigned by experts) are 1 for Liver Patient and 2 for Non-Liver Patient. Figure 2 shows a snapshot of the mentioned dataset.

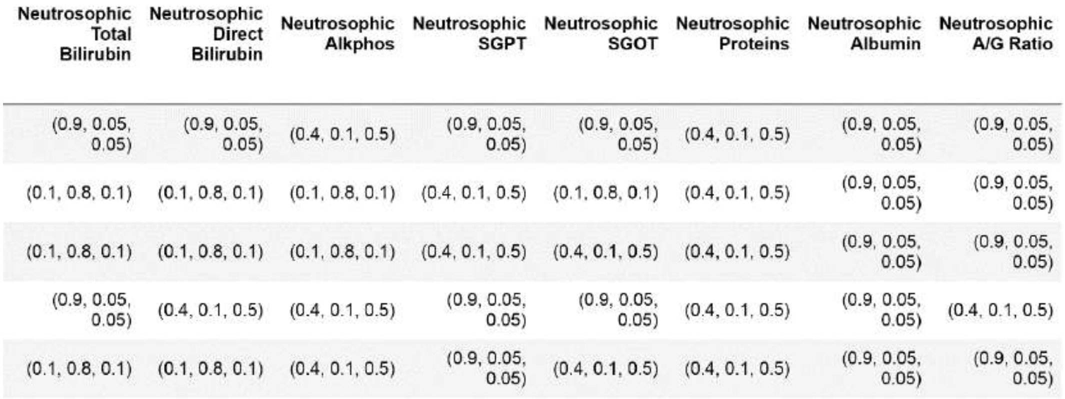
The liver disease patient dataset is a representative example of big data, characterized by its volume, complexity, velocity, variety, veracity, and value. With 30,000 records, each containing ten attributes, the dataset's sheer size necessitates big data techniques for storage and analysis. The complex nature of the data, encompassing diverse factors like age, gender, bilirubin levels, and enzyme levels, demands sophisticated processing methods. While the dataset is static, its potential for rapid analysis to develop effective machine learning models highlights its velocity. The heterogeneity of data types, including text and numeric, contributes to its variety. The dataset's distribution across various healthcare sources, such as hospitals and clinics, exemplifies its veracity. Finally, the dataset's potential to aid in early liver disease detection, potentially saving lives, underscores its immense value.

Transforming traditional big data into Neutrosophic data further enhances its value by effectively managing the inherent uncertainty and imprecision in the dataset. Neutrosophic data allows for better modeling of complex and ambiguous information, leading to improved predictive accuracy and robustness of machine learning models. This transformation facilitates a more nuanced analysis, accommodating various degrees of truth, indeterminacy, and falsehood, which is particularly beneficial in medical paper where data often involves inherent uncertainties. An example of converting conventional data to neutrosophic data is shown in the following screenshot, illustrating the careful analysis of the augmentation that is verified through this conversion.



	Age of the patient	Gender of the patient	Total Bilirubin	Direct Bilirubin	Alkphos Alkaline Phosphotase	Sgpt Alamine Aminotransferase	Sgot Aspartate Aminotransferase	Total Protiens	ALB Albumin	A/G Ratio Albumin and Globulin Ratio	Result
0	65.0	Female	0.7	0.1	187.0	16.0	18.0	6.8	3.3	0.90	1
1	62.0	Male	10.9	5.5	699.0	64.0	100.0	7.5	3.2	0.74	1
2	62.0	Male	7.3	4.1	490.0	60.0	68.0	7.0	3.3	0.89	1
3	58.0	Male	1.0	0.4	182.0	14.0	20.0	6.8	3.4	1.00	1
4	72.0	Male	3.9	2.0	195.0	27.0	59.0	7.3	2.4	0.40	1

Screenshot (A) showing the traditional data.



Neutrosophic Total Bilirubin	Neutrosophic Direct Bilirubin	Neutrosophic Alkphos	Neutrosophic SGPT	Neutrosophic SGOT	Neutrosophic Proteins	Neutrosophic Albumin	Neutrosophic A/G Ratio
(0.9, 0.05, 0.05)	(0.9, 0.05, 0.05)	(0.4, 0.1, 0.5)	(0.9, 0.05, 0.05)	(0.9, 0.05, 0.05)	(0.4, 0.1, 0.5)	(0.9, 0.05, 0.05)	(0.9, 0.05, 0.05)
(0.1, 0.8, 0.1)	(0.1, 0.8, 0.1)	(0.1, 0.8, 0.1)	(0.4, 0.1, 0.5)	(0.1, 0.8, 0.1)	(0.4, 0.1, 0.5)	(0.9, 0.05, 0.05)	(0.9, 0.05, 0.05)
(0.1, 0.8, 0.1)	(0.1, 0.8, 0.1)	(0.1, 0.8, 0.1)	(0.4, 0.1, 0.5)	(0.4, 0.1, 0.5)	(0.4, 0.1, 0.5)	(0.9, 0.05, 0.05)	(0.9, 0.05, 0.05)
(0.9, 0.05, 0.05)	(0.4, 0.1, 0.5)	(0.4, 0.1, 0.5)	(0.9, 0.05, 0.05)	(0.9, 0.05, 0.05)	(0.4, 0.1, 0.5)	(0.9, 0.05, 0.05)	(0.4, 0.1, 0.5)
(0.1, 0.8, 0.1)	(0.1, 0.8, 0.1)	(0.4, 0.1, 0.5)	(0.9, 0.05, 0.05)	(0.4, 0.1, 0.5)	(0.4, 0.1, 0.5)	(0.9, 0.05, 0.05)	(0.9, 0.05, 0.05)

Screenshot (B) showing the data after it was converted (neutrosophic data).

5.3 Evaluation Metrics for Neutrosophic NNs Classification

To assess the performance of our Neutrosophic-based NNs model, we will employ several evaluation metrics commonly used in classification tasks:

Accuracy: Overall percentage of correctly classified data points.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{7}$$

Precision: Ratio of true positives (correctly classified positive cases) to all predicted positive cases.

$$Pr\ e\ cision = \frac{TP}{TP+FP} \tag{8}$$

Recall: Ratio of true positives to all actual positive cases in the data.

$$Re\ c\ all = \frac{TP}{TP+FN} \tag{10}$$

F1-Score: Harmonic mean of precision and recall, providing a balanced view of model performance.

$$F1Score = 2 \times \frac{Pr\ ecision \times Re\ call}{Pr\ ecision + Re\ call} \tag{11}$$

By evaluating these metrics, we can measurement the performance of our Neutrosophic-based model with a baseline NNs model. These metrics will help us assess the effectiveness of the Neutrosophic approach in handling uncertainty and improving classification accuracy for Big Data.

5.4 Result Analysis and Discussion

This section delves into the key findings from the experiment and explores the implications of incorporating Neutrosophic Sets for liver diseases. Where Neutrosophic NNs model demonstrates excellent performance in data classification, showcasing a high ability to accurately distinguish between different classes. The model effectively balances recognizing true positive cases while minimizing false positives, making it well suited for applications requiring high precision and reliable outcomes. It shows a strong capacity to detect most true positive instances, while maintaining a low rate of false positives; thereby enhancing its reliability in scenarios where failing to identify positive cases can be costly or dangerous. Overall, the model exhibits balanced and stable performance, establishing itself as an effective and dependable tool for complex classification tasks as shown as in Table 1

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Neutrosophic NNs	79.08 %	74.58 %	77.64%	75.63%

Table 1 shown as the Evaluation Matrix of Neutrosophic NNs model, achieved a noteworthy improvement in classification performance. With an accuracy of 79.08%, the model outperforms many traditional approaches, demonstrating its effectiveness in correctly classifying data points. Its precision of 74.58% indicates a strong ability to correctly identify positive cases, minimizing false positives. The recall of 77.64% reflects the model's proficiency in detecting most actual positive instances, reducing false negatives. Furthermore, the F1-score of 75.63% highlights a balanced performance between precision and recall, making the Neutrosophic NNs model a robust choice for applications that require reliable and accurate classification under conditions of uncertainty. This suggests that incorporating Neutrosophic Sets enhances the model’s capability to handle ambiguous and indeterminate information effectively.

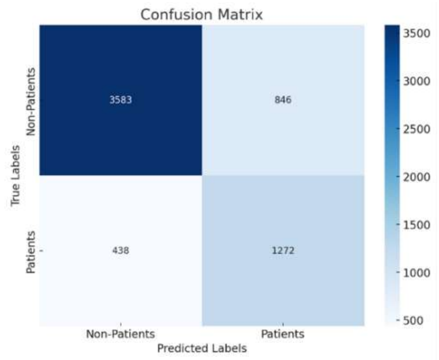


Figure.1. Confusion matrix for Neutrosophic NNs model. The model shows low rates of false positives and negatives, resulting in a higher number of true positives.

Figure 1 shows the Confusion matrix of Neutrosophic NNs model, That provides a detailed evaluation of the Neutrosophic NNs model used for predicting liver diseases. It displays the performance of the model by showing the actual versus predicted classifications across four quadrants: True Positives (Patiennon-patients identified as Patients), True Negatives (Non-Patients correctly identified as Non-Patients), False Positives (Non-Patients incorrectly identified as Patients), and False Negatives (Patients incorrectly identified as non-patients). Specifically, the matrix indicates that out of the total predictions, 3583 were correctly identified as non-patients, 1272 were correctly identified as Patients, 846 Non-Patients were misclassified as patients, and 438 Patients were misclassified as non-patients. This detailed breakdown allows for a comprehensive understanding of the model's accuracy, sensitivity, and specificity, demonstrating its effectiveness and areas for improvement in clinical settings for liver disease prediction.

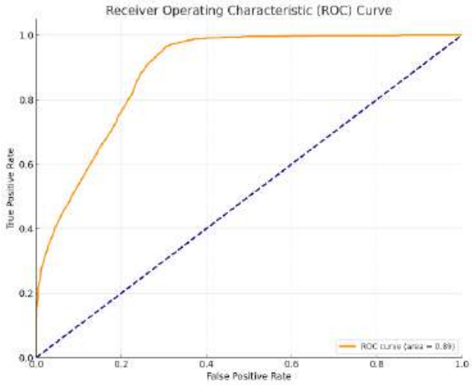


Figure.2. ROC curves for Neutrosophic NNs model The AUC values indicate the superior ability of Neutrosophic NNs model. These curves provide valuable insights into model performance, helping in making informed decisions for classification tasks.

Figure 2 presents the Operating Characteristic (ROC) curve for Neutrosophic NNs model, The Receiver of which is a crucial tool for evaluating the performance of a binary classification model by plotting the True Positive Rate (sensitivity) against the False Positive Rate (1-specificity) at various thresholds. The orange line represents the model's performance, while the diagonal dashed line signifies a random classifier with no discriminative power, having an Area under the Curve (AUC) of 0.5. Here, the AUC is 0.89, indicating a high discriminative ability, meaning there is an 89% chance that the model will correctly distinguish between positive and negative classes. The closer the ROC curve is to the top-left corner, the more accurate the model. This high AUC value underscores the model's effectiveness in differentiating between classes, making it a valuable tool for predictive analytics in medical research and practice, and essential for comparing and selecting the best-performing classification models.

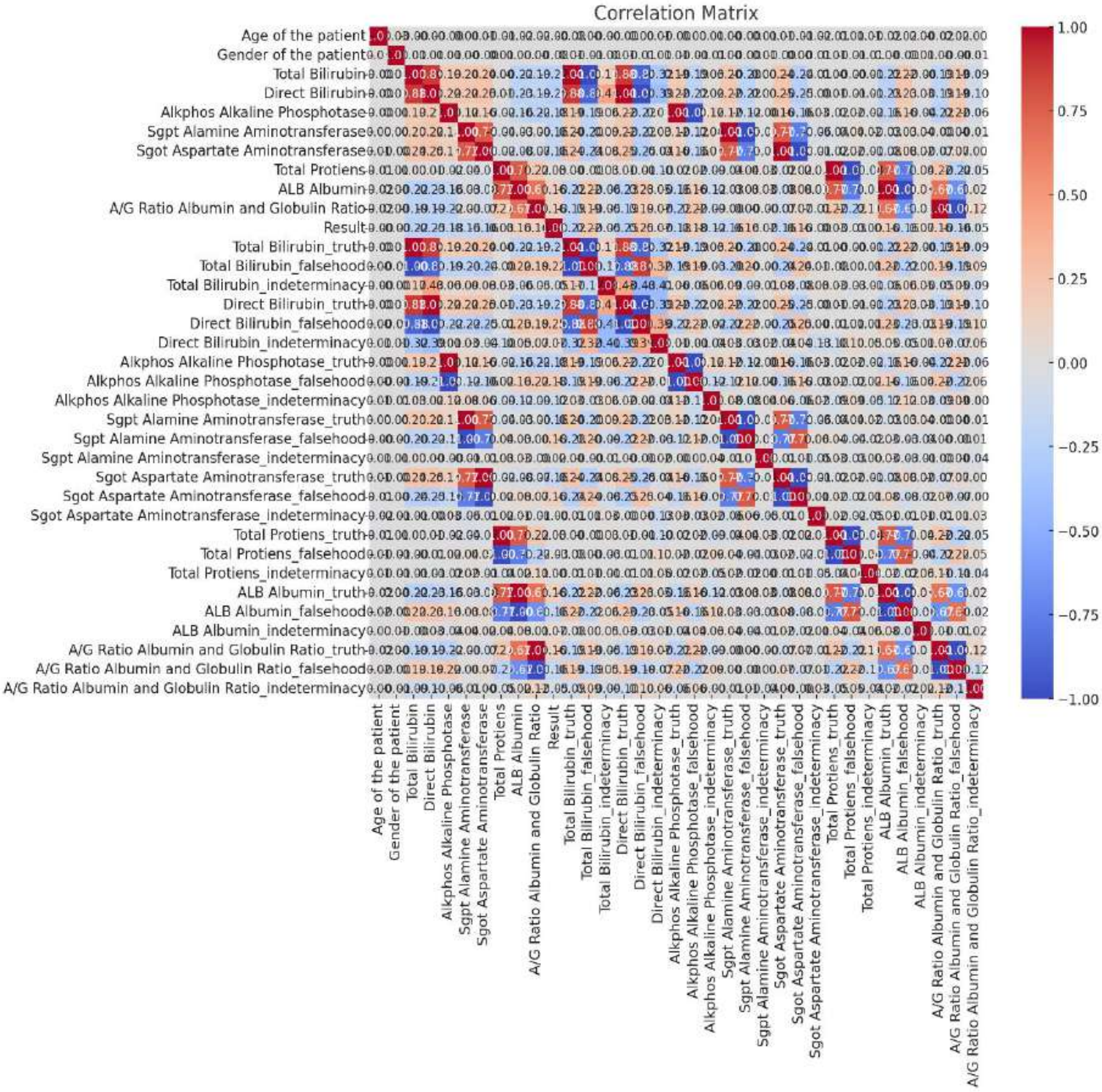


Figure 3 shows a correlation matrix revealing significant associations between liver enzymes and proteins, with gender showing a negative link, indicating crucial information for accurate liver function investigation.

Figure 3 presents the correlation matrix for Neutrosophic NNs model that represents the pairwise correlation coefficients between various clinical and biochemical parameters. The matrix includes a broad range of variables, such as age, gender, total and direct bilirubin levels, alkaline phosphatase, aspartate aminotransferase, albumin, and protein levels, among others. Each cell in the matrix indicates the correlation coefficient between the corresponding pair of variables, with values ranging from -1 to 1. A value close to 1 implies a strong positive correlation, meaning that as one variable increases, the other tends to increase as well. Conversely, a value close to -1 indicates a strong negative correlation, where an increase in one variable corresponds to a decrease in the other. A value around 0 suggests no linear correlation between the variables. The color gradient in the matrix helps in visualizing the strength and direction of the correlations, with red hues representing positive correlations, blue hues indicating negative correlations, and white or neutral colors denoting weak or no correlations. Notably, the matrix also distinguishes between 'truth', 'falsehood', and 'indeterminacy' for certain parameters, highlighting different states or categories within the dataset. This detailed correlation analysis is crucial for identifying potential relationships between clinical and biochemical markers, which can be instrumental in diagnostic and prognostic evaluations. Understanding these correlations helps in uncovering underlying patterns and associations, thereby contributing to more informed clinical decision-making and research insights in the medical field.

6. Conclusion:

This paper investigated the efficacy of neutrosophic neural networks (NNS) in predicting liver diseases. We employed a comprehensive dataset encompassing over 30,000 patient records and 10 different variables to train and evaluate the neutrosophic NNs model. Through this approach, we aim to contribute to the early and accurate detection of liver diseases. The neutrosophic NNs model's performance was assessed using critical metrics such as accuracy, precision, recall, F1-score, and AUC. Our analysis revealed that the neutrosophic NNs model achieved promising results, demonstrating its potential for liver disease prediction. While further research is warranted to explore more nuanced disease classification and individual risk assessment, this study establishes a foundation for utilizing neutrosophic NNs in the domain of liver disease diagnosis. By incorporating more interpretable data and refining the model's architecture, future work can enhance its effectiveness. Overall, this paper underscores the potential of neutrosophic NNs as a valuable tool for early liver disease detection, paving the way for improved healthcare interventions and patient care.

References:

1. Pramanik, P. K. D., Pal, S., Mukhopadhyay, M., & Singh, S. P. (2021). Big Data classification: techniques and tools. In Applications of Big Data in Healthcare (pp. 1-43). Academic Press.
2. Abdalla, H. B., & Abuhaija, B. (2023). Comprehensive analysis of various big data classification techniques: A challenging overview. Journal of Information & Knowledge Management, 22(01), 2250083.
3. Suthaharan, S. (2014). Big data classification: Problems and challenges in network intrusion prediction with machine learning. ACM SIGMETRICS Performance Evaluation Review, 41(4), 70-73.
4. Smarandache, F. (1999). A unifying field in Logics: Neutrosophic Logic. In Philosophy (pp. 1-141). American Research Press.
5. Smarandache, F. (2006, May). Neutrosophic set-a generalization of the intuitionistic fuzzy set. In 2006 IEEE international conference on granular computing (pp. 38-42). IEEE.
6. Smarandache, F. (2022). Operators on single-valued neutrosophic oversets, neutrosophic undersets, and neutrosophic offsets. Collected Papers, 9, 112.
7. Singh, S. K., Abolghasemi, V., & Anisi, M. H. (2022). Skin cancer diagnosis based on neutrosophic features with a deep neural network. Sensors, 22(16), 6261.
8. Rathore, M. M., Shah, S. A., Shukla, D., Bentafat, E., & Bakiras, S. (2021). The role of ai, machine learning, and big data in digital twinning: A systematic literature review, challenges, and opportunities. IEEE Access, 9, 32030-32052.

9. Ngiam, K. Y., & Khor, W. (2019). Big data and machine learning algorithms for health-care delivery. The Lancet Oncology, 20(5), e262-e273.
10. Stergiou, C. L., Plageras, A. P., Psannis, K. E., & Gupta, B. B. (2020). Secure machine-learning scenario from big data in cloud computing via internet of things network. Handbook of Computer Networks and Cyber Security: Principles and Paradigms, 525-554.
11. Rahman, M. S., & Reza, H. (2022). A systematic review towards big data analytics in social media. Big Data Mining and Analytics, 5(3), 228-244.
12. Salama, A. A., El-Ghareeb, H. A., Manie, A. M., & Smarandache, F. (2014). Introduction to develop some software programs for dealing with neutrosophic sets. Neutrosophic Sets and Systems, 3, 51-52.
13. Elwahsh, H., Gamal, M., Salama, A., & El-Henawy, I. (2018). A novel approach for classifying MANETs attacks with a neutrosophic intelligent system based on genetic algorithm. Security and Communication Networks, 2018, 1-7.
14. ElWahsh, H., Gamal, M., Salama, A., & El-Henawy, I. (2018). Intrusion detection system and neutrosophic theory for MANETs: A comparative study. Neutrosophic Sets and Systems, 23, 16-22.
15. Yasser, I., Twakol, A., Abd El-Khalek, A. A., Samrah, A., & Salama, A. A. (2020). COVID-X: novel health-fog framework based on neutrosophic classifier for confrontation covid-19. Neutrosophic Sets and Systems, 35, 1-21.
16. Amin, B., Salama, A. A., El-Henawy, I. M., Mahfouz, K., & Gafar, M. G. (2021). Intelligent neutrosophic diagnostic system for cardiotocography data. Computational Intelligence and Neuroscience, 2021, 1-12.
17. Pramanik, P. K. D., Pal, S., Mukhopadhyay, M., & Singh, S. P. (2021). Big Data classification: techniques and tools. In Applications of Big Data in Healthcare (pp. 1-43). Academic Press.
18. Abdalla, H. B., & Abuhaija, B. (2023). Comprehensive analysis of various big data classification techniques: A challenging overview. Journal of Information & Knowledge Management, 22(01), 2250083.
19. Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S. (2018). Big Data technologies: A survey. Journal of King Saud University-Computer and Information Sciences, 30(4), 431-448.
20. Provost, F., & Fawcett, T. (2013). Data science for business.
21. Lee, D. K., In, J., & Lee, S. (2015). Standard deviation and standard error of the mean. Korean journal of anesthesiology, 68(3), 220.
22. Huang, L., Qin, J., Zhou, Y., Zhu, F., Liu, L., & Shao, L. (2023). Normalization techniques in training dnns: Methodology, analysis and application. IEEE transactions on pattern analysis and machine intelligence, 45(8), 10173-10196.
23. Hu, K. (2020). Become competent within one day in generating boxplots and violin plots for a novice without prior R experience. Methods and protocols, 3(4), 64.
24. Abu Qamar, M., & Hassan, N. (2019). An approach toward a Q-neutrosophic soft set and its application in decision making. Symmetry, 11(2), 139.

25. Farag, R. M., Shams, M. Y., Aldawody, D. A., Khalid, H. E., El-Bakry, H. M., & Salama, A. A. (2024). Integration between Bioinformatics Algorithms and Neutrosophic Theory. Neutrosophic Sets and Systems, 66, 34-54.

26. Salama, A. A., Shams, M. Y., Khalid, H. E., & Mousa, D. E. (2024). Enhancing Medical Image Quality using Neutrosophic Fuzzy Domain and Multi-Level Enhancement Transforms: A Comparative Study for Leukemia Detection and Classification. Neutrosophic Sets and Systems, 65, 32-56.

27. Salama, A. A., Shams, M. Y., Elseuofi, S., & Khalid, H. E. (2024). Exploring Neutrosophic Numeral System Algorithms for Handling Uncertainty and Ambiguity in Numerical Data: An Overview and Future Directions. Neutrosophic Sets and Systems, 65, 253-295.

28. Alhabib, A., Alhabib, R., Khalid, H. E., & Salama, A. A. (2024). A Neutrosophic Study for the Transmission of Infection with Pathogenic Fungi from Males of Olive Fly Insects to Their Females. Neutrosophic Sets and Systems, 64, 38-45.

29. Salama, A. A., Tarek, Z., Darwish, E. Y., Elseuof, S., & Shams, M. Y. (2024). Neutrosophic Encoding and Decoding Algorithm for ASCII Code System. Neutrosophic Sets and Systems, 63, 105-129.

30. Essa, A. K., Sabbagh, R., Salama, A. A., Khalid, H. E., Aziz, A. A. A., & Mohammed, A. A. (2023). An Overview of Neutrosophic Theory in Medicine and Healthcare. Neutrosophic Sets and Systems, 61(1), 13.

Received: April 20, 2024. Accepted: Aug 22, 2024