OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 02 - Issue 07 (2025)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

# ENHANCING INSTANCE SELECTION WITH ENSEMBLE METHODS: A COMPARATIVE ANALYSIS OF ACCURACY AND DATA COMPRESSION ACROSS MULTIPLE ALGORITHMS

***Muhammad Ans Khalid[1], Hira Batool[2], Abdulrehman Arif[3], Syed Zohair Quain Haider[4]*, Hassan Ahmad[5]***

*[1, 3, 4]Department of Computer science and Information Technology, University of Southern Punjab, Multan, Pakistan*

*[2]Department of Computer Science, National University of Modern Languages (NUML), Islamabad, Pakistan.*

*[5]Department of Computer Science & IT, Times Institute Multan, Pakistan*

****Corresponding Author:** zohairhaider67@gmail.com*

**Article Info**



**Abstract**

Instance selection is a crucial preprocessing step aimed at reducing the computational complexity of machine learning models, particularly when integrated with feature selection and dimensionality reduction techniques. This process enhances the overall prediction accuracy by selecting the most representative instances and removing redundant or irrelevant data points. While instance selection has been widely studied, the application of ensemble methods to instance selection remains underexplored in existing literature. Ensemble methods, known for their ability to combine multiple models to improve predictive performance, offer a promising avenue for refining instance selection strategies. In this study, we explore the efficacy of four ensemble strategies voting, feature bagging, Additive Noise, and Bagging when applied to a diverse set of nine instance selection algorithms: CNN, ENN, GE, IB2, IB3, All-KNN, RMHC, MC, and RENN. The goal is to determine how well these ensemble methods can enhance the performance of individual instance selection algorithms in terms of accuracy and compression efficiency. The experimental evaluation is conducted in three distinct phases. In the first phase, we focus on assessing the accuracy and data compression capabilities of each instance selection method on a specific dataset. This phase establishes a baseline performance metric for the instance selection algorithms and provides an understanding of how well they generalize across different types of data. The second phase involves evaluating the performance of the four ensemble strategies across six different datasets. This phase aims to assess the robustness of ensemble methods in improving the accuracy of instance selection algorithms, especially when dealing with diverse data characteristics. The ensemble methods are expected to offer a more stable and generalized solution compared to individual instance selection techniques, potentially leading to improved prediction results. In the final phase, we conduct a comparative analysis of the compression effectiveness of the nine instance selection algorithms, both individually and within the ensemble frameworks, across the six datasets. By comparing the compression ratios achieved by each method, we aim to identify the most efficient instance selection techniques that retain key information while reducing the size of the data for subsequent modeling. This analysis is critical for understanding the trade-off between computational efficiency and the preservation of valuable data.

**Keywords:**

*Instance Selection, Repeated Edited Nearest Neighbor, Clonal Selection Algorithm, Object Selection by clustering, Back Propagation.*

## 1. Introduction

The typical data mining process comprises four main stages: data collection, data preprocessing, predictive modeling, and post-processing. Instance Selection (IS) plays a crucial role during the data preparation stage by helping to identify the most relevant training data. Through IS techniques, the training dataset can be effectively refined by filtering out redundant or noisy instances. This filtering not only enhances computational efficiency but also boosts model accuracy. IS algorithms act as intelligent data filters, determining which examples should be retained or discarded from the dataset. While predictive models are constructed in the third phase, their success heavily depends on the quality of the training data. Although ensemble learning has been widely explored, only limited research has addressed the use of ensembles specifically in IS. This study aims to evaluate four ensemble techniques—Bagging, Feature Bagging, Voting, and Additive Noise adapted for instance selection.

The concept of Artificial Intelligence (AI) was first introduced in 1956. Since then, it has evolved into a vital part of modern technology, playing a significant role in our everyday lives. AI refers to the simulation of human intelligence in machines that are designed to think, learn, and act like humans. It encompasses systems and devices that can perform tasks typically requiring human cognition, such as learning from experience, reasoning, and problem-solving. In simple terms, AI enables machines and software to process information and make decisions in a manner similar to human thinking. These systems can improve their performance over time through experience and data input. Common applications of AI include self-driving vehicles and computer programs capable of playing strategic games like chess both of which rely heavily on data analysis and processing.

Supervised learning is a machine learning approach focused on discovering patterns and relationships between input features and a target variable. The target variable can be either numerical or categorical. When it is numerical, the task is referred to as regression, and when it is categorical, it is known as classification. The target variable typically represents individual values or classes. The outcome of supervised learning is a model a structure formed by the algorithm during training using labeled datasets. This process enables the model to learn associations between the input features and the output variable. Lazy learning algorithms do not build a predictive model during the training phase. Instead, they store the training data and delay computation until a prediction request is made. This approach has both advantages and drawbacks. Lazy learners require significant storage to maintain the entire dataset and tend to be slower at prediction time. However, they offer very fast training and can adapt quickly by simply adding new instances to the database, which is especially useful in dynamic or changing environments. They also provide good local approximations due to their reliance on raw data during prediction.

Eager learning algorithms, in contrast, construct a predictive model during the training phase. These models such as decision trees, neural networks, or support vector machines capture the relationship between input features and target values. Once trained, the model can be used to quickly make predictions for new instances. Eager learners require more time during training but benefit from efficient predictions and reduced storage needs, as only the trained model (not the entire dataset) must be retained.

### 1.1. Problem Statement

Instance Selection is a vital data preprocessing technique widely applicable across various machine learning tasks. Given the massive size of modern datasets, IS methods help reduce data volume, making it more manageable and less resource-intensive for training algorithms. By eliminating irrelevant, noisy, or redundant instances before model training, IS techniques can enhance the efficiency and accuracy of classification models. Despite their importance, there remains a gap in understanding how ensemble strategies can further improve the performance of IS methods.

The primary objective of IS techniques is to reduce the size of the training dataset—a process referred to as compression without compromising prediction accuracy. A secondary but equally important goal is to enhance model performance by removing outliers and noisy data points. Ensemble learning combines multiple models or techniques to improve overall prediction quality. When applied to IS, ensemble methods iterate over different subsets of the data, creating diverse selections and thus enhancing robustness and accuracy. This research focuses on comparing various IS ensemble approaches. Initially, an overview of IS strategies and recent advancements in ensemble methods for predictive modeling is presented. The study is divided into three key parts:

1. Analyzing the trade-off between accuracy and compression;
2. Comparing the accuracy of four ensemble methods adapted to IS techniques across multiple datasets;
3. Evaluating the compression ratios achieved with and without the use of ensemble techniques.

## 2. Related work

Wrapper and filter approaches constitute the primary categories of sample selection techniques. To address the challenges associated with instance selection (IS), numerous methods have been introduced. A common foundation for many wrapper strategies is the K-Nearest Neighbors (KNN) classifier. Condensed Nearest Neighbor (CNN) [3] is one of the earliest methods introduced. It is a basic incremental IS strategy that aims to reduce the original dataset T. Initially, one instance p is randomly picked from each class Y to form the subset S. Then, each instance in T is classified using S. If p is misclassified, it is added to S. However, this can result in noisy instances remaining due to incorrect classification by their neighbors.

A generalized form of CNN, known as GCNN, was proposed in [4]. It utilizes a stronger absorption criterion compared to the original version. Absorption is determined by the nearest neighbor and other instances belonging to different classes. Initially, a y-prototype is selected at random for each class y. After verifying all samples, if they are strongly absorbed, the process ends. If any y-samples remain unabsorbed, a new y-prototype is randomly selected for that label. If none are available, the algorithm proceeds without changes. Consequently, S comprises instances that represent T. GCNN tends to outperform other instance-based data reduction techniques in terms of accuracy.

Edited Nearest Neighbor (ENN) [8] is another classical IS method. It operates by eliminating noisy instances based on their class agreement with their KNN neighbors. If an instance in T belongs to a different class than the majority of its KNN, it is excluded. The Repeated ENN variant applies the ENN method iteratively until all instances conform to the majority class of their k-nearest neighbors. All k-NN [5] is an alternative to ENN, where a loop from i = 1 to k identifies and removes misclassified instances by KNN. Multi-edit, another edited KNN approach by [9], begins by randomly partitioning the data into n subsets. Each partition is trained using a 1-NN classifier, and neighbors are found in adjacent partitions. This continues for t iterations until no further instances are eliminated.

Selected Nearest Neighbor (SNN) [10] is a modified version of CNN, which forms a subset S such that each element in S is more similar to another instance from the same class in S than to any instance in the total set TS. Here, TS is classified exactly by 1NN. Instance-based learning methods IB2 and IB3 were introduced in [11]. IB2 incrementally selects correctly classified instances. IB3 extends IB2 by tracking classification history, ensuring the removal of an instance does not impact classification accuracy. DROP through DROP5 are additional KNN variants discussed by [7]. DROP1 eliminates an instance p if its associates in TS can be correctly classified without it. DROP2 improves this by scanning the full training set. DROP3 and DROP4 incorporate ENN to filter noise prior to applying DROP2. DROP5 adds the removal of nearest enemies to smooth decision boundaries. Author [7] also introduced Iterative Case Filtering (ICF), using Reachable(p) and Coverage(p) sets, representing neighbors and associates. ICF removes p if the size of Reachable is greater than Coverage. It incorporates ENN to aid the process. A similar method, C-Pruner, was proposed by [12], using a spin order to filter instances from the same class. If |Coverage| < |Reachable|, the instance is considered noisy; if reachable instances correctly classify p, it is deemed irrelevant. Support Vector Machines (SVMs) were applied for instance selection by [13]. Since only support vectors are used to define decision boundaries, they inherently function as selected instances. [14] combined SVMs with DROP2 in a wrapper strategy, while SV-KNN Clustering uses SVM for selection followed by K-means clustering on the resulting support vectors. Evolutionary algorithms (EA) were adopted in IS by [15][13], drawing on biological evolution principles [14]. Chromosomes representing instance subsets are evaluated by fitness functions, generally involving a classifier. Chromosomes evolve over multiple generations through crossover and mutation operations. The meme algorithm, introduced by [18], merges EA with local search in a single evolutionary cycle. Chromosomes encoded as binary strings are refined for both precision and subset size reduction, improving the evolutionary process.

**Table 1 Summary of IS Techniques**

| Technique | Type | Base |
|---|---|---|
| CNN | W | Misclassification |
| GCNN | W | Misclassification, Absorption |
| ALL-KNN | W | Misclassification |

| | | |
|---|---|---|
| SNN | W | Misclassification |
| SV-KNN | W | SVM, KNN |
| ENN | W | Misclassification |
| ICF | W | Reachable, Coverage |
| EA | W | Natural Evaluation |
| TS | W | Tabu Search |
| CSA | W | Local Search |
| MA | W | Evolutionary |
| IB | W | Misclassification |
| Multiedit | W | Misclassification |
| Drop | W | Associate |
| GA | W | Chromosomes |
| GCM | F | Clustering |
| NSB | F | Clustering |
| OSC | F | Interior instances ,clustering, border |
| CLU | F | Interior instances, clustering. |
| PSR | F | Instance relevance |
| WP | F | Instance weight |
| POP | F | Border Instances, weakness |

## 3. Methods and materials

This section outlines the methodology adopted in our research. The accompanying figure illustrates the specific methods and procedures implemented throughout the study.

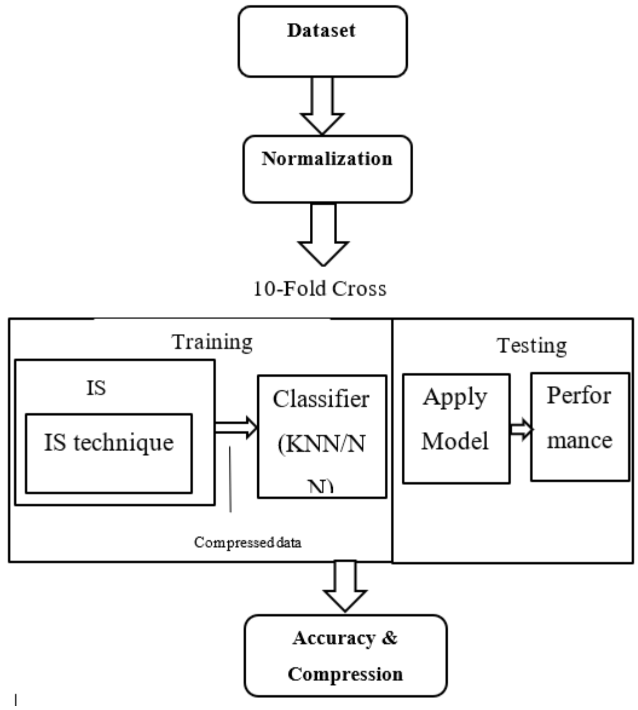



**Figure 1 Flow of Experiment**

The figure depicts how the experimental work is carried out. We retrieve the dataset and normalized it. Then we used cross validation for training and testing the dataset. In the training portion, we used the ensemble classifier with the IS technique and classify the compressed dataset using a predictive model. In the training portion, compression and accuracy of the dataset will measure. The chapter continues with a detailed description of each stage.

### 3.1. Normalization

Normalization scales values to fit into a defined range. It's important to change the value range while engaging with the properties of many scales and units. For a proper comparison, all characteristics be scaled equally by using the Euclidean distance. Normalization is important for comparing properties of different sizes. Normalization can be accomplished in four ways.

- Statistical normalizing is another term for z transformation. The average of the data is subtracted from all values, after that the standard deviation is divide, the data distribution therefore data has a zero mean and a one variance. It maintains the basic range of the data and is fewer affected by extremes.
- All Attribute values are normalized to a set of values called a range. As a result, the highest value is max, and the lowest number is min. All other numbers have been scaled to fit inside the specified

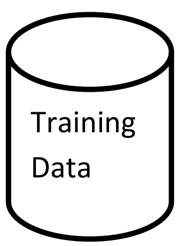
range. Outliers can impact this strategy since the boundaries shift towards them. However, because this approach preserves the original data point distribution, it may also be helpful to anonymize data.

- This normalization is defined as the percentage of each feature value to the whole attribute. This indicates that the entire sum of the feature values is divided by each value. Both positive and negative, missing values too, are not included in the total. Negative numbers can be handled otherwise, when used as absolute values, they will produce a miscalculation.

- The interquartile range is used to normalize data. The difference among the twenty-five to seventy-five percentiles is calculated using the interquartile range, commonly known as the high and low quartiles. They are determined by sorting the data first and then choosing the data that detach the top twenty five percent of instances from the rest. The sorted data is divided in half by the value of fifty percent. IQR is the outcome of the interquartile range normalization computation. The range of IQR is in the interval between the center of fifty percent of the data, hence outliers have less of an impact using this normalization procedure. In this approach, unbounded values as well as missing values can be disregarded. In addition, if no limited values can be identified, the attribute will be disregarded.

### 3.2. Cross Validation

Cross-validation involves two main stages: one dedicated to training and the other to testing. In the training phase, a model is developed using a specific subset of the data, while in the testing phase, the model's performance is assessed using previously unseen data. The dataset is divided into k equally sized subsets. One of these subsets is used as the test set, and the remaining k–1 subset is used for training. This process is repeated k times, with each subset serving as the test set exactly once. To produce a final performance estimate, the results from all k iterations are aggregated commonly by calculating the median or mean of the performance scores. The parameter k, representing the number of folds, is adjustable based on experimental requirements. Evaluating a model on independent test data offers a more reliable indication of its generalization ability. It also helps in identifying overfitting where the model performs well on training data but poorly on unseen data. Consequently, a model that performs effectively on test data is more likely to maintain accuracy when applied to new datasets.




| **Iteration** | **Fold 1** | **Fold 2** | **Fold 3** | **Fold 4** | **Fold 5** |
|---|---|---|---|---|---|
| **1st** | Test | Train | Train | Train | Train |
| **2nd** | xTrain | Test | Train | Train | Train |
| **3rd** | Train | Train | Test | Train | Train |

| | | | | | |
|---|---|---|---|---|---|
| **4th** | Train | Train | Train | Test | Train |
| **5th** | Train | Train | Train | Train | Test |

**Figure 2 Illustration of 5-Fold Cross Validation**

### 3.2.1. Sampling Methods

Different sampling strategies can be used to generate subsets for cross-validation:

- **Linear Sampling:** Divides the dataset while preserving the original order of examples. Subsets are composed of consecutive entries.
- **Shuffled Sampling**: Randomly generates subsets by randomly selecting instances from the dataset.
- **Stratified Sampling:** Ensures that each subset maintains the same class distribution as the full dataset. For example, in a binary classification task, it maintains approximately equal proportions of each class in all subsets.

### 3.3. Ensemble of Instance Selection (IS)

The use of IS within ensembles has been explored through various approaches. In [5], the initial concept involved aggregating outputs from ENN models over several k-values. Another strategy, as outlined in [41], applied boosting to enhance instance selection. An indirect approach introduced by [42] utilized an ensemble of classifiers constructed through diverse IS outputs, maintaining diversity by varying the datasets resulting from IS. A democratic voting approach to IS was also proposed by [42], where frequently misclassified instances are flagged for removal. In addition, [43] implemented Bagging and Feature Bagging as ensemble strategies for example selection. A comprehensive comparison was conducted by [44], examining methods such as MultiBoost, ReweightBoost, FloatBoost, and AdaBoost. This work treated IS as a binary classification problem each instance is labeled as either retained or discarded based on an IS-specific label. It's important to note that IS ensembles differ from classifier ensembles. In IS ensembles, we often lack the ground truth about which instances should be retained. Therefore, IS labels are approximated and can only be inferred post-classification. This constraint limits the use of supervised ensemble combiners, prompting the use of methods like Bagging and Feature Bagging, which operate without requiring ground truth labels. In this study, four ensemble approaches were evaluated for their effectiveness in the IS context. The goal was to enhance instance selection by leveraging ensemble strategies while maintaining or improving classification accuracy.

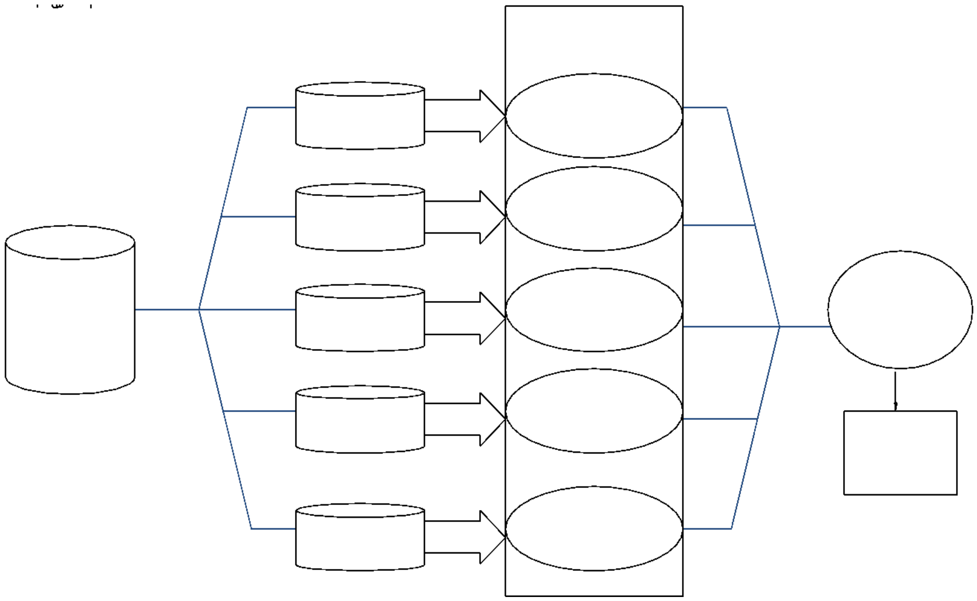


**Figure 3 Ensemble of instance selection**

**Algorithm 1 Algorithm of Ensemble Learning**

**Input**: Training set D= {($m_1$,$n_1$),….,($m_n$,$n_n$)
Number of iterations **Z**
Threshold **t**
**Output**: Subset of D (selected instance) D ⊆ S
Initialize:
  IS = Instance Selection algorithm
  S = sample instances from D

For i = 1 to z do:
    A = Apply Ensemble Learning on D and S
    P = Apply Instance Selection algorithm IS on A
    V = Record Votes for instances in S

X = Select Instances from D based on Voting results V and threshold t

Return X

- Bagging

Each prototype set (p) is generated as a randomly selected subset of the original dataset (D). The training set (T) is then constructed using an instance selection (IS) base method, referred to as the IS model. Since the dataset (D) is sampled from a uniform distribution, all selected instances in (p) are assigned equal weights.

**Algorithm 2 Algorithm Bagging of IS**

---

**Input**: Training set D= {($m_1$,$n_1$),….,($m_n$,$n_n$)}

**Output**: Subset of D (selected instance) D ⊆ S

Initialize:

IS = Instance Selection algorithm

S = Sampled subset from training data D

For i = 1 to z do:

A = Generate bootstrap sample from D and S

P = Apply IS algorithm on A

V = Collect votes for instances in S

X = Select instances from D based on votes V and threshold t

Return X

---

- Feature Bagging

It differs from traditional bagging primarily in its sampling strategy. While both methods employ an instance selection (IS) model to generate prototype sets (P), feature bagging forms the dataset (D) by randomly selecting a subset of features from the training set (T) rather than instances. The selected features are drawn from a uniform distribution, and each vote in the ensemble contributes equally to the final decision, as all are assigned equal weights.

**Algorithm 3 Algorithm feature Bagging of IS**

---

**Input**: Training set D= {($m_1$,$n_1$),….,($m_n$,$n_n$)}

**Output**: Subset of D (selected instance) D ⊆ S

IS= IS algorithm

S= samples ratio

z= number of iteration

t = Threshold

wi=weight

Initialize weights w_i = 0 for all instances in D

---

For i = 1 to z do:

A = Draw a sample from D using ratio S

P = Apply instance selection algorithm IS on A

For each instance selected in P:

Increase its weight: w_i += 1/z

V = Aggregate votes (weights) for all instances in D

X = Select instances from D where vote weight w_i ≥ threshold t

Return X

- Additive Noise

In the additive noise approach, the instance selection (IS) model is used to construct each sample set (P. To form the final dataset, noise is introduced to each input instance within the training set. Despite the perturbation, all resulting samples contribute equally to the ensemble, with each vote carrying the same weight.

**Algorithm 4 Algorithm Noise of IS**

**Input**: Training set D= {$(m_1,n_1)$,….,$(m_n,n_n)$}

**Output**: Subset of D (selected instance) D ⊆ S

D: Training dataset
IS: Instance selection algorithm

S= samples ratio

z= number of iteration

t = Threshold

wi=weight

N= Noise

$\sigma$ = noise level //add Gaussian noise

Initialize weights w_i = 0 for all instances in D

For i = 1 to z do:

For each instance m in D:

```
        Add Gaussian noise: m' = m + N(0, σ)
     A = D with added noise
     P = Apply IS algorithm on A
     For each instance in P:
        Increase its vote weight: w_i += 1/z
  V = Aggregate votes for all instances
  X = Select instances from D where w_i ≥ threshold t
  Return X
```

- Voting

Voting is a classification technique in which multiple independent models often of different types are trained on the same dataset. These models then "vote" on the predicted output for each instance being classified. The final prediction is based on the majority or weighted majority of votes. This approach enhances the overall accuracy by combining the strengths of different models.

### 3.4. Classification

Classification is a supervised learning approach used to predict the category or class to which a given data instance belongs. For example, classification can be used to determine whether a train will arrive on time, be delayed, or be significantly late. Another common use case includes categorizing individuals based on age groups such as underage, average, or above-average. For a classification model to be effectively evaluated, the dataset must be labeled. This means each data instance should have a label (the true class) and predictor attributes (features used to predict the class). The label holds the actual values, while the prediction attribute represents the values predicted by the classification model.

#### 3.4.1. k-Nearest Neighbor (KNN)

The KNN algorithm classifies an unlabeled instance by comparing it to the k most similar instances in the training set referred to as its nearest neighbors. The similarity or "closeness" is typically measured using distance metrics in an n-dimensional feature space, with Euclidean distance being one of the most commonly used metrics. Since distance-based calculations are sensitive to scale, it is recommended to normalize the data before applying KNN.

The classification process using KNN involves two main steps:

1. **Distance Calculation:** Compute the distances between the new (unlabeled) instance and all instances in the training set.

2. **Voting Mechanism:** Determine the class of the new instance based on a majority vote among its k

nearest neighbors. Optionally, weighted voting can be applied, where closer neighbors have a greater influence on the final prediction than those further away.

An Artificial Neural Network (ANN), or simply a Neural Network (NN), is a computational model inspired by the structure and functioning of biological neural networks found in the human brain. ANNs consist of interconnected artificial neurons that process data collectively using connectionist principles. These networks are capable of learning and adapting their internal structure during training by adjusting their parameters in response to input data and desired outputs. Modern neural networks are widely employed for modeling complex input-output relationships and identifying intricate patterns within datasets.

### 3.4.2. Feedforward Neural Network (FFNN)

A FFNN is a type of ANN in which the flow of information is unidirectional from input nodes through any hidden layers to output nodes without forming any loops or cycles. These networks are particularly suited for problems involving classification, regression, and pattern recognition.

### 3.4.3. Backpropagation

The Backpropagation algorithm is a supervised learning technique used to train ANNs. It consists of two primary phases:

1. **Forward Propagation:** Inputs are passed through the network to produce an output.
2. **Backward Propagation:** The output is compared with the target value, and the resulting error is propagated backward through the network to update the weights.

The algorithm adjusts the weights of the connections iteratively to minimize the value of a predefined error function. This process is repeated for many training cycles until the network reaches an acceptable performance level. When the error becomes sufficiently small, the network is considered to have learned the underlying pattern or function associated with the target data.

### 3.4.4. Multilayer Perceptron (MLP)

A MLP is a type of FFNN composed of multiple layers of interconnected neurons. Each neuron (except in the input layer) applies a non-linear activation function commonly a sigmoid function to its inputs. MLPs are organized as directed acyclic graphs, with each layer fully connected to the next. Training an MLP typically involves the backpropagation algorithm, where multiple layers of processing units (neurons) are adjusted iteratively. The goal is to minimize the error between the predicted and actual outputs across training examples. In many real-world applications, MLPs demonstrate strong performance due to their ability to model non-linear relationships and generalize from data when appropriately trained. The activation function used in many neural networks is the sigmoid function, which maps input values into the range between 0 and 1. To align with this output range, input values are typically normalized, often to a range between -1 and +1. Normalization ensures that the data is compatible with the activation function and improves the learning performance of the model.

In neural networks, the type of output node depends on the nature of the task:

- For classification problems, a sigmoid output node is commonly used.
- For regression tasks, the output node is typically linear to handle continuous values.

## 4. Results and implementation

This section presents the results of the experimental work and performance evaluation of the proposed methodology. The effectiveness of combining IS techniques with different ensemble learning classifiers is assessed and analyzed.

### 4.1. Dataset detail

To ensure fair and consistent evaluation, multiple experiments were conducted using six well-known datasets sourced from the UCI Machine Learning Repository. The characteristics of these datasets are summarized in Table 1.

**Table 2 Datasets Detail**

| Datasets | Number of Classes | Number of Attributes | Number of Instances |
|---|---|---|---|
| Iris | 3 | 4 | 150 |
| Sonar | 2 | 60 | 208 |
| Heat | 3 | 13 | 303 |
| Glass | 6 | 9 | 214 |
| Liver | 2 | 7 | 345 |
| Cleveland | 3 | 60 | 297 |

### 4.2. Parameter Configuration

To maintain consistency across experimental conditions, predefined parameter settings were used for all methods. The parameter values for each technique are provided in Table 3.

**Table 3 Parameters Configuration**

| Method | Parameters |
|---|---|
| **CNN** | - |
| **ENN** | K=3,5 |
| **All-KNN** | K Start =3, K Stop= 5,7 |
| **GE** | - |

| | |
|---|---|
| **IB2** | K= 3,5 Parameter upper interval =0.9 Parameter lower interval =0.7 |
| **IB3** | K= 3,5 |
| **RENN** | K=3,5 |
| **MC** | Prototype= 5-50, Iteration= [50,200] |
| **RMHC** | Prototype= 5-50 iteration [50,200] |
| **Bagging** | Threshold=0.1 iteration=10 |
| **Feature Bagging** | Threshold=0.1 iteration=10 sample ratio=0.8 |
| **Additive Noise** | Threshold=0.1 iteration=10 Noise=0.1 |
| **KNN** | K=3,5,7 |
| **NN** | Training cycle=100-500 Learning rate = [0.01,0.1] |

### 4.3. Experimental setup

The experimental process starts with data preprocessing, where all numerical attributes are normalized to the [0, 1] range. A 10-fold cross-validation technique is applied to ensure robust performance assessment. Within each fold, instance selection techniques are employed to reduce the training set, after which the classification model either KNN or NN is trained and tested.

### 4.4. Classification Performance Analysis

The performance of the ensemble methods is significantly influenced by the threshold value applied. In this context, a threshold defines the criteria for selecting samples or models during the ensemble process. Different threshold values result in varying subsets of instances, directly affecting the model's performance. Identifying an optimal threshold is therefore crucial for maximizing classification accuracy and generalization.

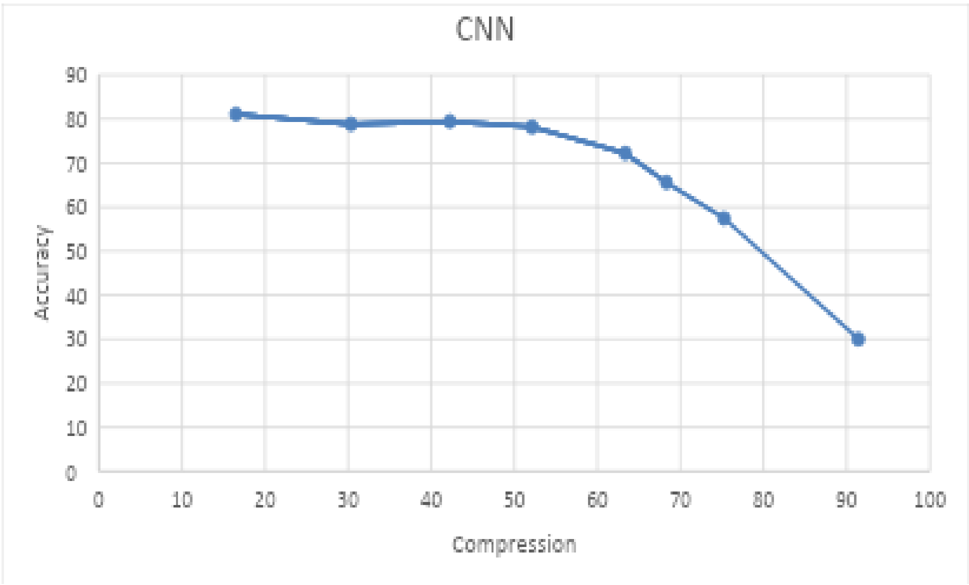



**Figure 4 Threshold Performance on CNN**

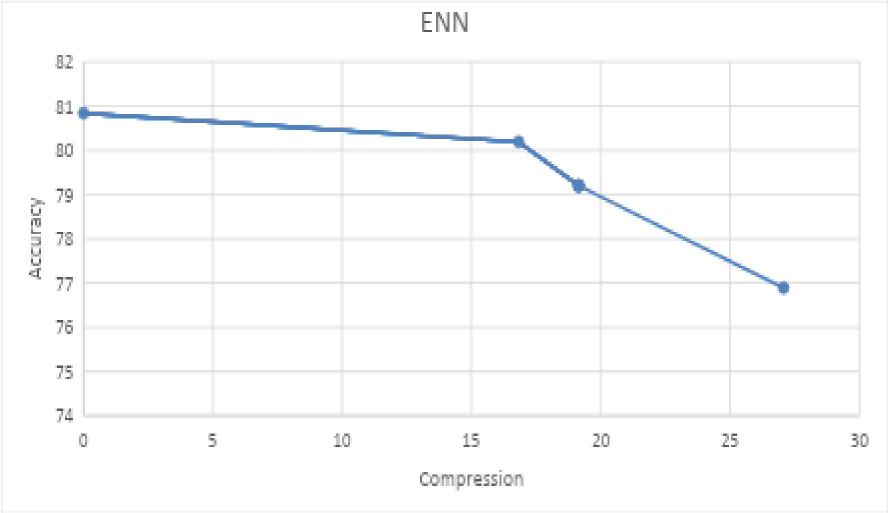



**Figure 5 Threshold Performance on ENN**

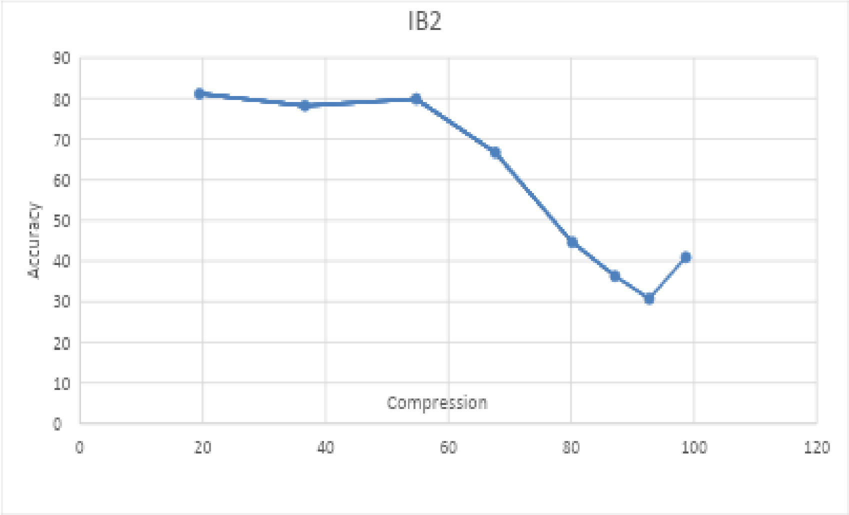



**Figure 6 Threshold Performance on IB2**

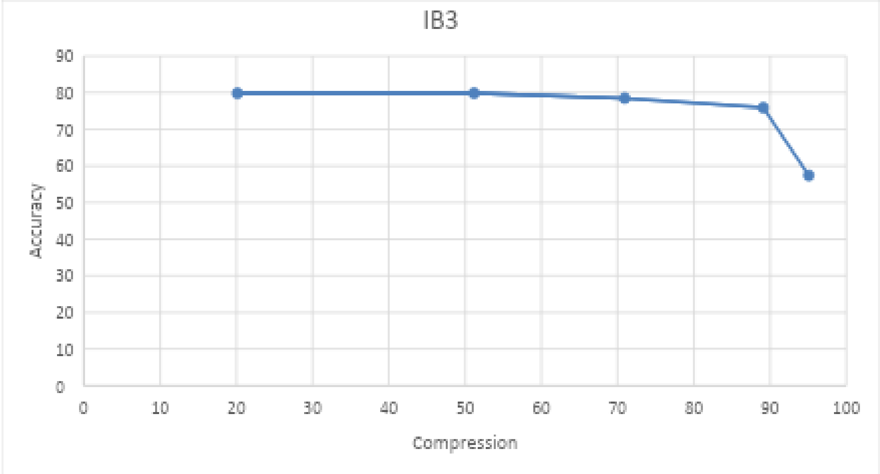



**Figure 7 Threshold Performance on IB3**

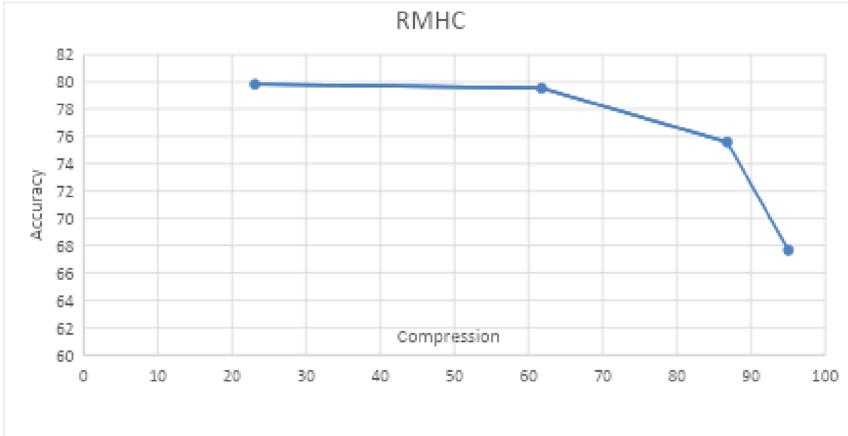



**Figure 8 Threshold Performance on RMHC**

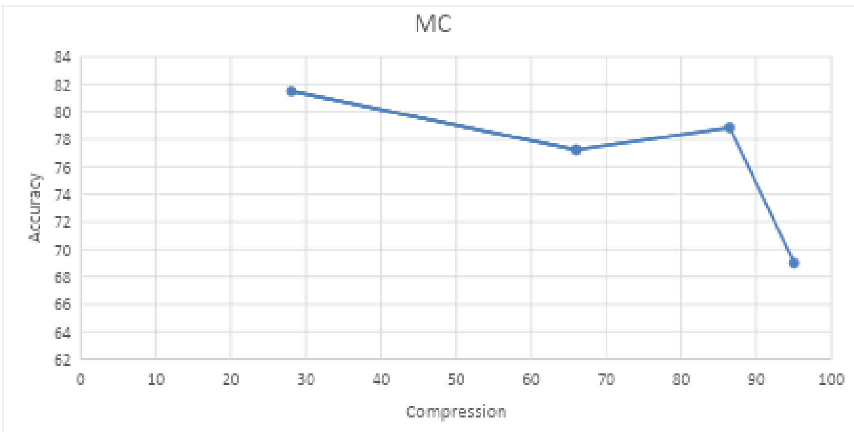



**Figure 9 Threshold Performance on MC**

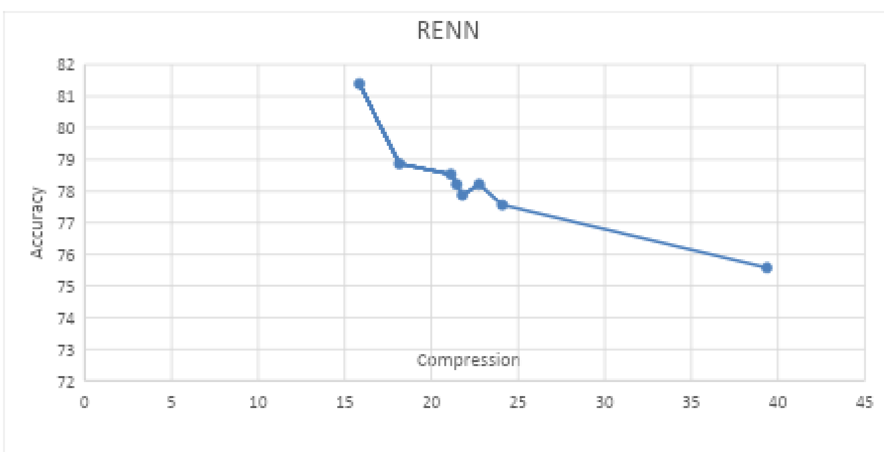



**Figure 10 Threshold Performance on RENN**

In the figures above, we examined the relationship between compression and accuracy based on varying threshold values, and the results were compared using different instance selection techniques. The Heart dataset was selected for this evaluation, and the accuracy of the 5-NN classifier was measured for each acceptance threshold: {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, and 0.8}. The experimental results indicate that increasing the threshold leads to greater dataset compression, which significantly affects classification accuracy in methods such as IB2, IB3, CNN, MC, and RMHC. In contrast, methods like ENN, All-KNN, and RENN demonstrate more stability, where minor compression has a negligible impact on classification performance. This suggests that these techniques are more resilient to instance reduction while preserving classification accuracy.

**Table 4 Iris Datasets Accuracy**

| IRIS-Accuracy | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **KNN** | | | | | | | | | |
| | CNN | ENN | All-KNN | GE | IB2 | IB3 | MC | RENN | RMHC |
| Basic | 61.33 | 96.67 | 96.67 | 70 | 50.67 | 87.33 | 94.67 | 96.67 | 54.67 |
| Bagging | 98 | 96.67 | 96.67 | 97.33 | 97.33 | 97.33 | 96.67 | 97.33 | 97.33 |
| Feature-Bagging | 96.67 | 96.67 | 97.33 | 96.67 | 96.67 | 97.33 | 97.33 | 97.33 | 97.33 |
| Noise | 97.33 | 97.33 | 97.33 | 97.33 | 97.33 | 97.33 | 97.33 | 97.33 | 96.67 |
| **NN** | | | | | | | | | |
| | CNN | ENN | All-KNN | GE | IB2 | IB3 | MC | RENN | RMHC |
| Basic | 67.33 | 96.67 | 86 | 56.67 | 89.33 | 96 | 74.67 | 96.67 | 47.33 |
| Bagging | 97.33 | 96.67 | 96.67 | 97.33 | 97.33 | 97.33 | 97.33 | 96.67 | 97.33 |
| Feature-Bagging | 97.33 | 96.67 | 96.67 | 96.67 | 97.33 | 97.33 | 96.67 | 96.67 | 97.33 |
| Noise | 97.33 | 96.67 | 97.33 | 97.33 | 97.33 | 97.33 | 97.33 | 97.33 | 97.33 |

**Table 5 Sonar Datasets accuracy**

| Sonar-Accuracy | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **KNN** | | | | | | | | | |
| | CNN | ENN | All-KNN | GE | IB2 | IB3 | MC | RENN | RMHC |
| Basic | 72.5 | 77.81 | 79.74 | 80.62 | 66.29 | 63.5 | 65.5 | 78.31 | 63.5 |
| Bagging | 83.6 | 79.74 | 81.19 | 82.1 | 83.1 | 82.62 | 82.19 | 79.74 | 80.17 |
| Feature-Bagging | 83.6 | 78.88 | 80.26 | 78.88 | 83.6 | 81.21 | 82.14 | 80.34 | 80.19 |
| Noise | 82.62 | 81.67 | 82.14 | 82.62 | 82.62 | 82.17 | 82.14 | 82.62 | 82.14 |
| **NN** | | | | | | | | | |

| | CNN | ENN | All-KNN | GE | IB2 | IB3 | MC | RENN | RMHC |
|---|---|---|---|---|---|---|---|---|---|
| Basic | 76 | 80.31 | 80.79 | 80.24 | 68.31 | 71.6 | 79.83 | 71.6 | 75.45 |
| Bagging | 83.74 | 80.79 | 83.74 | 81.31 | 81.26 | 82.76 | 82.21 | 79.83 | 82.69 |
| Feature-Bagging | 83.24 | 80.24 | 82.19 | 81.76 | 82.62 | 81.69 | 82.67 | 80.83 | 81.26 |
| Noise | 83.14 | 82.17 | 83.24 | 83.74 | 83.29 | 83.67 | 83.24 | 82.26 | 84.12 |

**Table 6 Heart datasets accuracy**

| Heart-Accuracy | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **KNN** | | | | | | | | | |
| | CNN | ENN | All-KNN | GE | IB2 | IB3 | MC | RENN | RMHC |
| Basic | 80.48 | 79.2 | 80.2 | 81.81 | 78.49 | 77.55 | 76.56 | 77.88 | 77.2 |
| Bagging | 82.13 | 80.84 | 80.2 | 81.81 | 81.15 | 81.47 | 81.49 | 78.57 | 80.17 |
| Feature-Bagging | 81.82 | 80.17 | 80.2 | 81.48 | 81.82 | 80.48 | 81.49 | 81.48 | 80.83 |
| Noise | 81.81 | 80.52 | 81.47 | 81.81 | 82.47 | 80.81 | 81.86 | 80.19 | 81.81 |
| **NN** | | | | | | | | | |
| | CNN | ENN | All-KNN | GE | IB2 | IB3 | MC | RENN | RMHC |
| Basic | 78.24 | 80.31 | 80.86 | 81.81 | 82.09 | 78.85 | 75.57 | 81.56 | 73.2 |
| Bagging | 82.54 | 81.29 | 82.18 | 83.18 | 83.16 | 83.14 | 83.49 | 80.19 | 82.15 |
| Feature-Bagging | 84.16 | 83.14 | 80.47 | 82.84 | 82.16 | 82.18 | 83.49 | 83.81 | 82.51 |
| Noise | 83.82 | 82.86 | 82.22 | 83.15 | 85.45 | 85.45 | 83.16 | 81.82 | 82.2 |

**Table 7 Glass datasets accuracy**

| Glass-Accuracy | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **KNN** | | | | | | | | | |
| | CNN | ENN | All-KNN | GE | IB2 | IB3 | MC | RENN | RMHC |
| Basic | 70.119 | 63.07 | 69.7 | 69.7 | 61.65 | 66.43 | 59.83 | 61.36 | 50.61 |
| Bagging | 70.19 | 63.25 | 69.29 | 69.22 | 70.19 | 70.17 | 69.26 | 61.41 | 69.72 |
| Feature-Bagging | 70.17 | 67.42 | 69.29 | 67.42 | 70.17 | 69.7 | 69.72 | 65.11 | 67.84 |
| Noise | 70.17 | 65.09 | 69.29 | 69.7 | 70.17 | 69.7 | 68.35 | 64.61 | 67.36 |
| **NN** | | | | | | | | | |
| | CNN | ENN | All-KNN | GE | IB2 | IB3 | MC | RENN | RMHC |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Basic | 57.47 | 64.68 | 64.13 | 65.95 | 61.69 | 60.41 | 61.36 | 63.27 | 55.13 |
| Bagging | 70.17 | 66.04 | 69.24 | 69.22 | 70.19 | 69.22 | 69.24 | 61.84 | 67.47 |
| Feature-Bagging | 70.65 | 65.43 | 68.83 | 68.77 | 71.62 | 70.67 | 68.31 | 66.06 | 67.36 |
| Noise | 70.79 | 66.47 | 69.14 | 66.47 | 71.15 | 69.29 | 67.88 | 64.16 | 66.49 |

**Table 8 Liver dataset accuracy**

| Liver Accuracy | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **KNN** | | | | | | | | | |
| | CNN | ENN | All-KNN | GE | IB2 | IB3 | MC | RENN | RMHC |
| Basic | 64.36 | 66.36 | 66.39 | 65.22 | 58.87 | 54.75 | 58.31 | 65.51 | 58.29 |
| Bagging | 65.22 | 67.27 | 66.39 | 65.22 | 65.51 | 67.85 | 66.98 | 66.7 | 68.13 |
| Feature-Bagging | 65.5 | 67.24 | 65.22 | 65.22 | 64.42 | 67.24 | 66.16 | 68.14 | 66.99 |
| Noise | 65.22 | 65.22 | 65.22 | 65.22 | 65.5 | 65.24 | 68.4 | 65.51 | 66.94 |
| **NN** | | | | | | | | | |
| | CNN | ENN | All-KNN | GE | IB2 | IB3 | MC | RENN | RMHC |
| Basic | 66.67 | 68.13 | 70.13 | 70.14 | 66.69 | 55.97 | 64.38 | 65.52 | 66.93 |
| Bagging | 71.35 | 68.11 | 71.92 | 71 | 72.45 | 72.79 | 70.45 | 68.12 | 69.6 |
| Feature-Bagging | 70.73 | 73.31 | 67.5 | 71.59 | 70.5 | 68.43 | 72.19 | 68.72 | 68.43 |
| Noise | 71.62 | 69.87 | 71.33 | 69.58 | 71.03 | 71.02 | 70.47 | 70.73 | 68.97 |

**Table 9 Cleveland dataset accuracy**

| Cleveland-Accuracy | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **KNN** | | | | | | | | | |
| | CNN | ENN | All-KNN | GE | IB2 | IB3 | MC | RENN | RMHC |
| Basic | 79.47 | 80.8 | 82.49 | 82.49 | 72.37 | 81.52 | 79.2 | 80.8 | 78.47 |
| Bagging | 82.51 | 81.15 | 82.49 | 82.49 | 81.48 | 81.17 | 81.48 | 82.85 | 82.85 |
| Feature-Bagging | 82.84 | 82.84 | 82.49 | 82.84 | 82.15 | 81.84 | 81.47 | 82.48 | 81.48 |
| Noise | 81.16 | 82.15 | 82.84 | 82.49 | 82.16 | 83.16 | 82.48 | 81.48 | 82.48 |
| **NN** | | | | | | | | | |
| | CNN | ENN | All-KNN | GE | IB2 | IB3 | MC | RENN | RMHC |
| Basic | 80.86 | 81.17 | 81.17 | 80.8 | 78.11 | 81.52 | 78.76 | 81.48 | 81.86 |
| Bagging | 83.2 | 83.17 | 82.83 | 83.16 | 82.84 | 82.16 | 82.16 | 84.16 | 84.16 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Feature-Bagging | 82.8 | 82.84 | 82.15 | 82.48 | 83.17 | 82.17 | 82.51 | 82.84 | 83.53 |
| Noise | 82.13 | 82.85 | 82.84 | 82.52 | 82.15 | 82.82 | 83.52 | 83.56 | 82.84 |

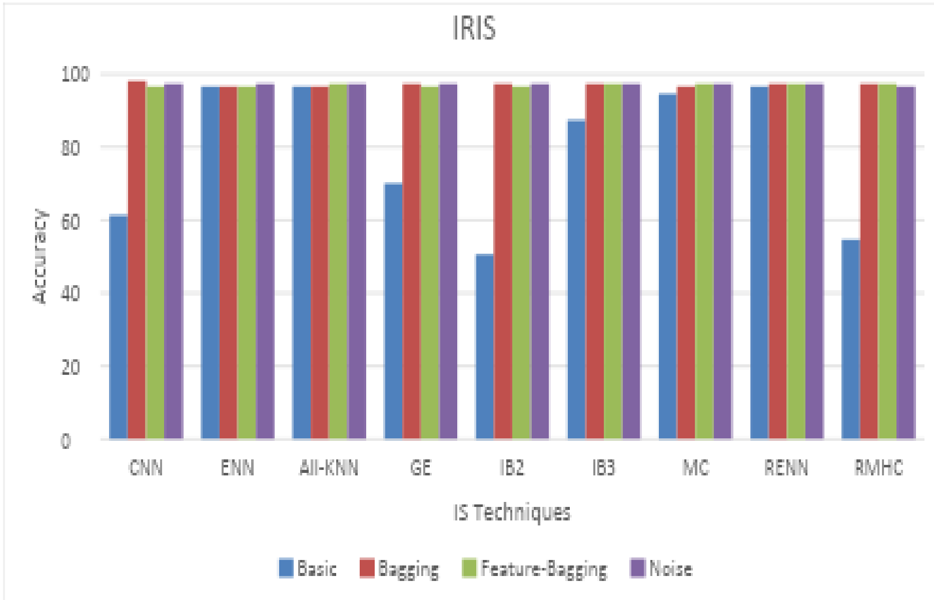



**Figure 11 Accuracy of Iris dataset on KNN classifier**

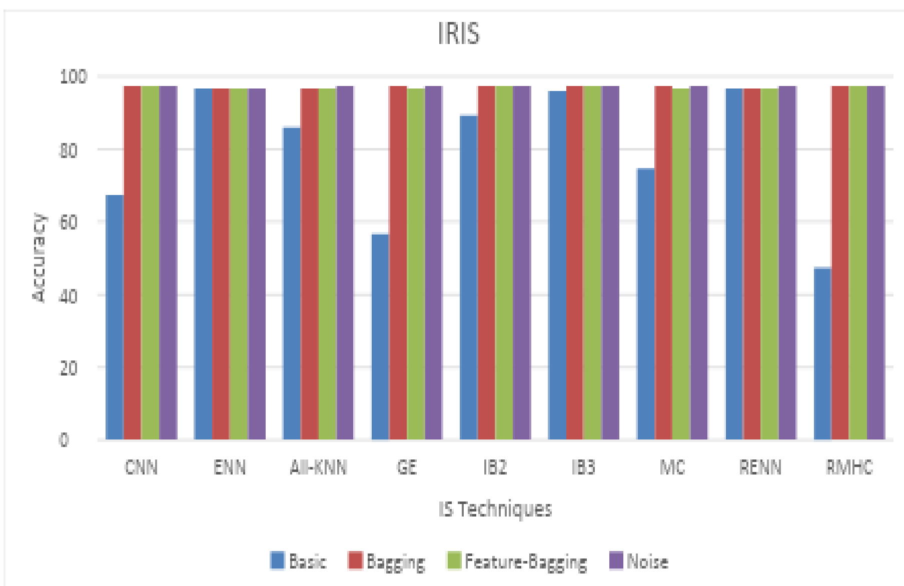



**Figure 12 Accuracy of Iris dataset on NN classifier**

Figures 11 and 12 illustrate that ensemble learning techniques contribute to improved classification accuracy. As shown in Table 4, the Bagging method achieves an accuracy of 98% with the KNN classifier and 97.33% with the NN classifier. While Bagging demonstrates higher accuracy with KNN, the overall average performance across methods is slightly better with the NN classifier. Among all instance selection techniques, ENN and All-KNN show the lowest performance on both KNN and NN classifiers.

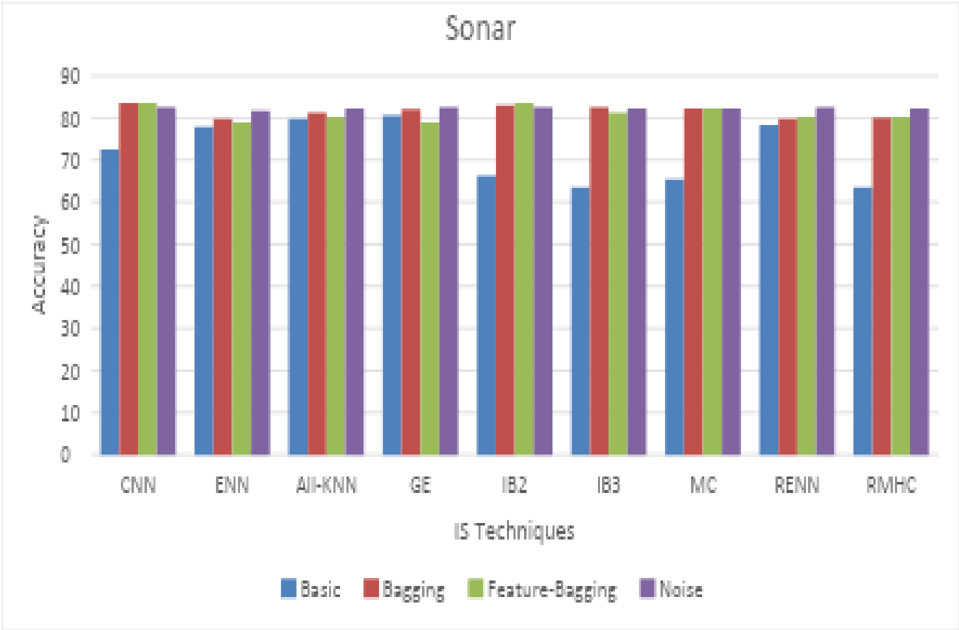



**Figure 13 Accuracy of Sonar dataset on KNN classifier**

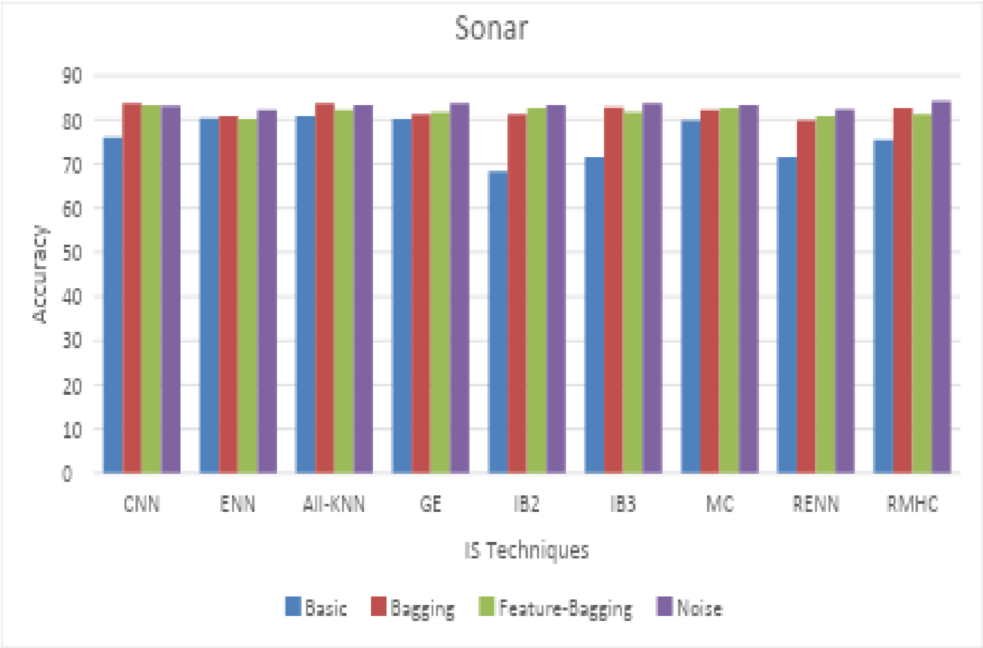



**Figure 14 Accuracy of Sonar dataset on NN classifier**

Figures 13 and 14 demonstrate that ensemble learning techniques yield higher accuracy compared to individual instance selection methods. According to Table 5, Bagging achieves an accuracy of 83.6% with the KNN classifier and 83.74% with the NN classifier. Among the instance selection techniques, CNN, IB2, and IB3 show strong performance with the KNN classifier, while CNN, IB2, and All-KNN perform better with the NN classifier.

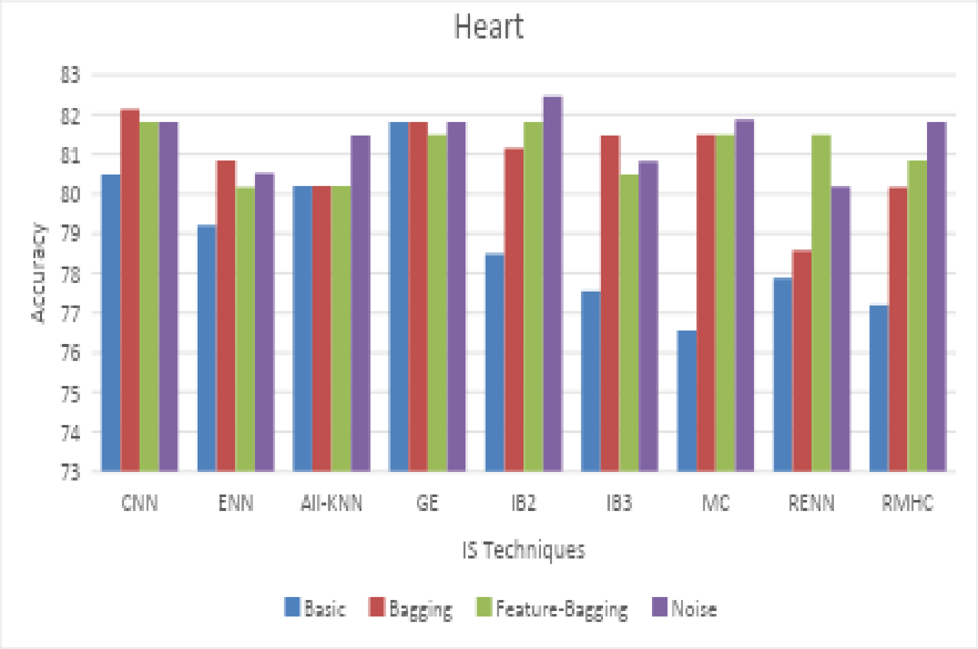



**Figure 15 Accuracy of Heart dataset on KNN classifier**

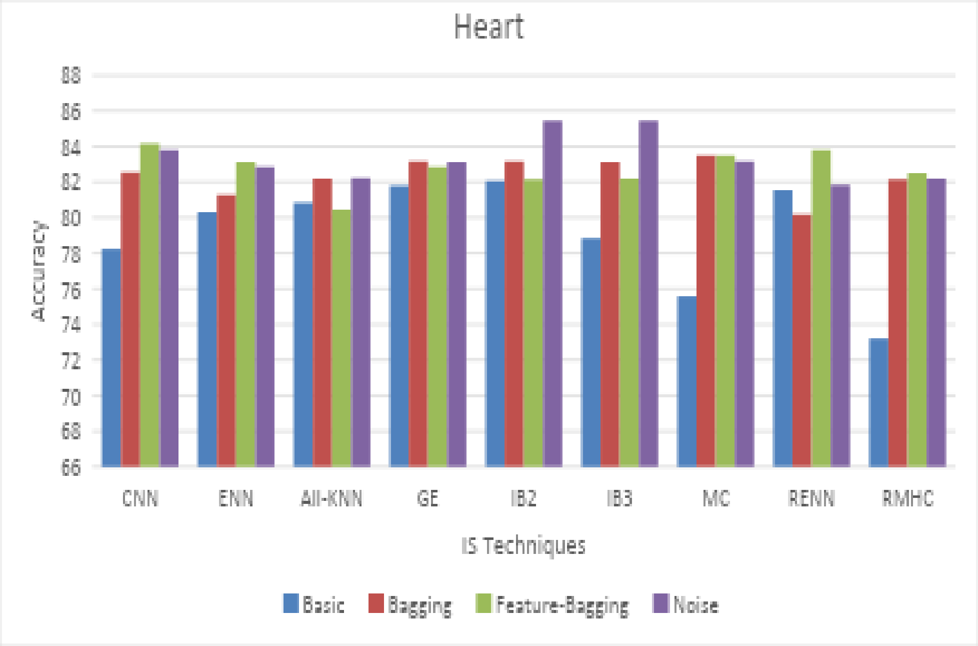



Figure 16 Accuracy of Heart dataset on NN classifier

The results for the Heart dataset using Bagging, Feature Bagging, and Additive Noise are illustrated in Figures 15 and 16. As shown in the graphs, the performance of CNN, GE, IB2, IB3, and MC with the KNN classifier is superior to other methods. Similarly, in the case of the NN classifier, CNN, GE, IB2, IB3, and MC outperform the rest, as well as their KNN counterparts. Notably, the NN classifier increases accuracy by approximately 2% to 3% in GE, IB2, IB3, All-KNN, and MC across all ensemble methods, as detailed in Table 6.

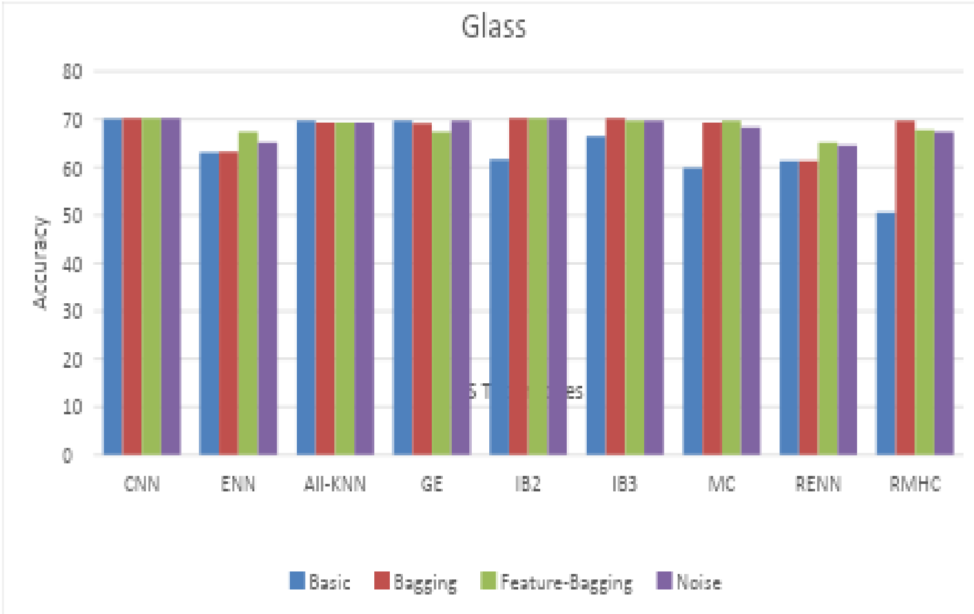



**Figure 17 Accuracy of Glass dataset on KNN classifier**

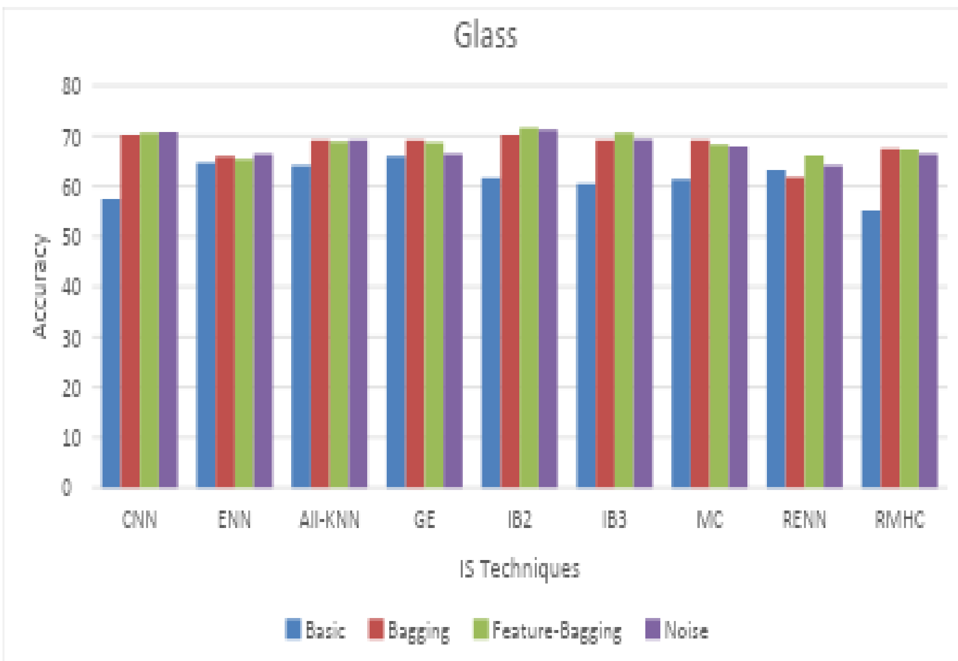



**Figure 18 Accuracy of Glass Dataset Using NN Classifier**

Figures 17 and 18 illustrate the performance results for the Glass dataset. In the case of the KNN classifier, CNN, IB2, and IB3 demonstrate superior performance compared to other instance selection methods. When using the NN classifier, CNN, IB2, and RMHC yield better results. Notably, Additive Noise enhances the NN classifier's performance, increasing accuracy by approximately 2% for IB2, MC, and RMHC. Conversely, ENN, RENN, and All-KNN exhibit the lowest accuracy levels on the Glass dataset.

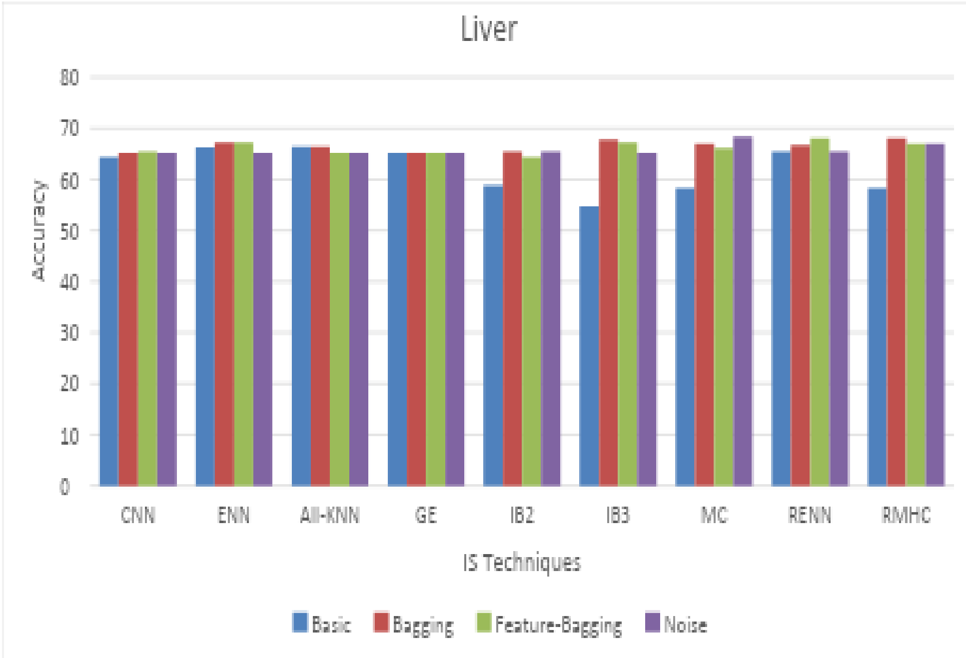



**Figure 19 Accuracy of Liver dataset on KNN classifier**

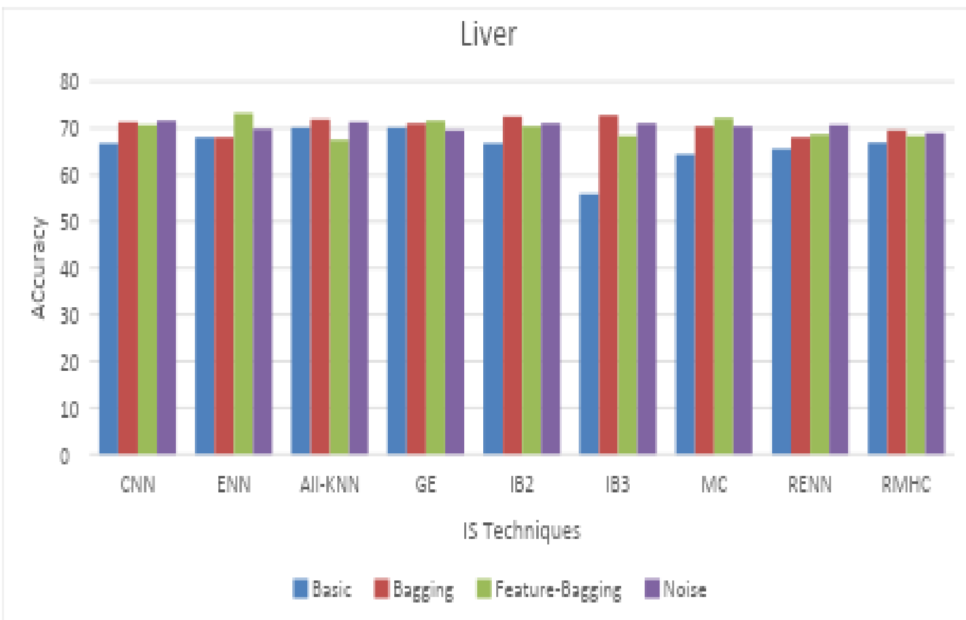



**Figure 20 Accuracy of Liver dataset on NN classifier**

The accuracy results for the Liver dataset are presented in Figures 19 and 20. Using the KNN classifier, ENN, RENN, IB3, and RMHC show relatively strong performance. In contrast, with the NN classifier, CNN, IB2, IB3, and MC outperform other methods. Overall, the NN classifier enhances accuracy by approximately 3% to 6% compared to KNN. Specifically, in the feature bagging approach, NN with additive noise shows an improvement of about 6% in accuracy over KNN. A detailed breakdown of the Liver dataset's accuracy is provided in Table 7.

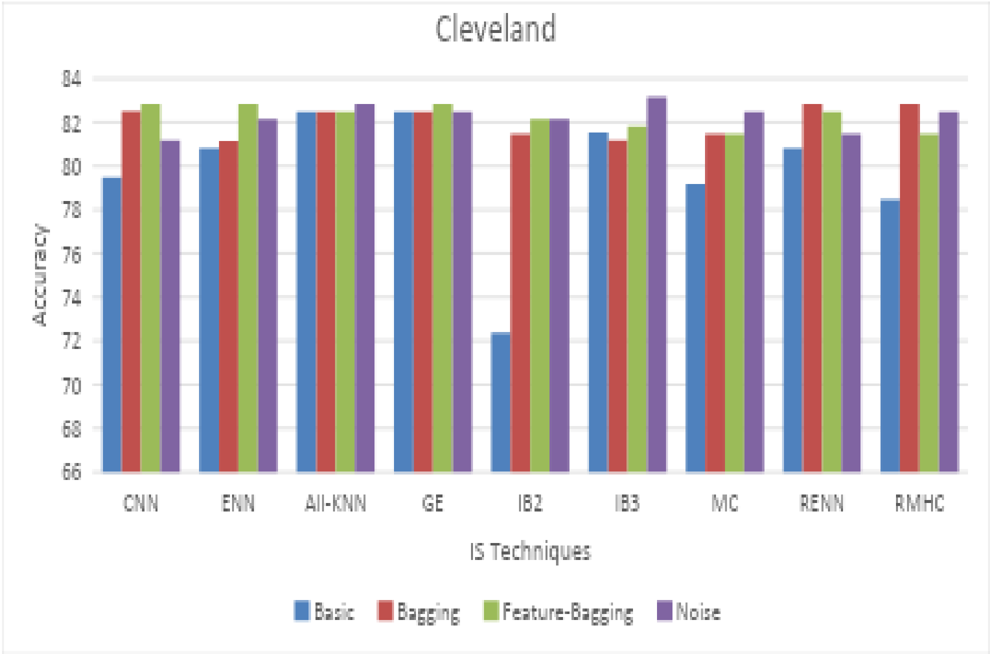



Figure 21 Accuracy of Cleveland dataset on KNN classifier

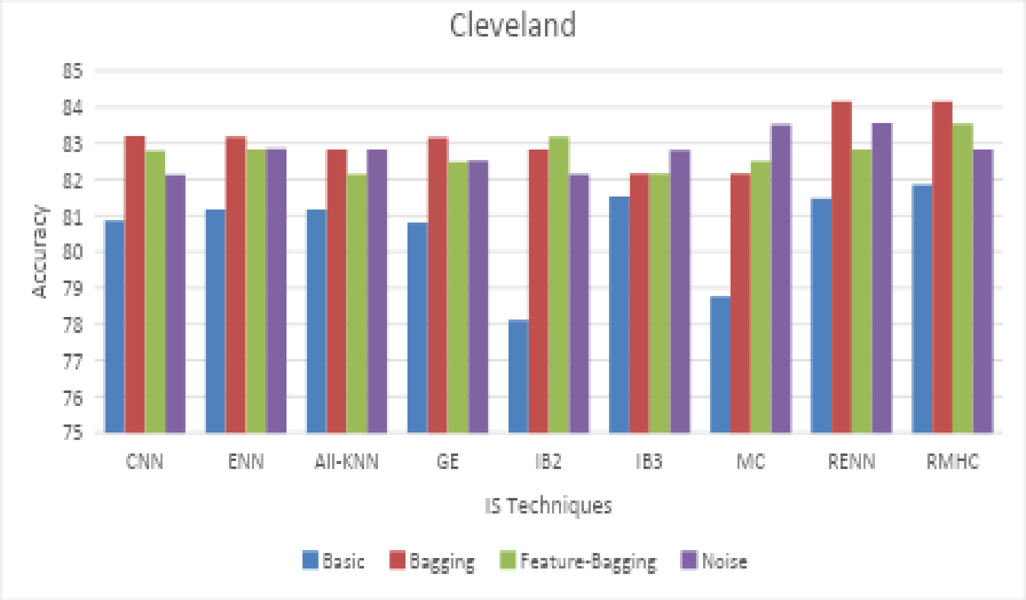



**Figure 22 Accuracy of the Cleveland Dataset Using the NN Classifier**

Figures 21 and 22 illustrate the performance of the ensemble instance selection techniques on the Cleveland dataset. With the KNN classifier, methods like CNN, ENN, RENN, and MC deliver strong results. In contrast, for the NN classifier, CNN, ENN, and RMHC show superior performance. Overall, the NN classifier demonstrates better accuracy than KNN, offering an improvement of approximately 2%. Detailed accuracy results for the Cleveland dataset are presented in Table 8.

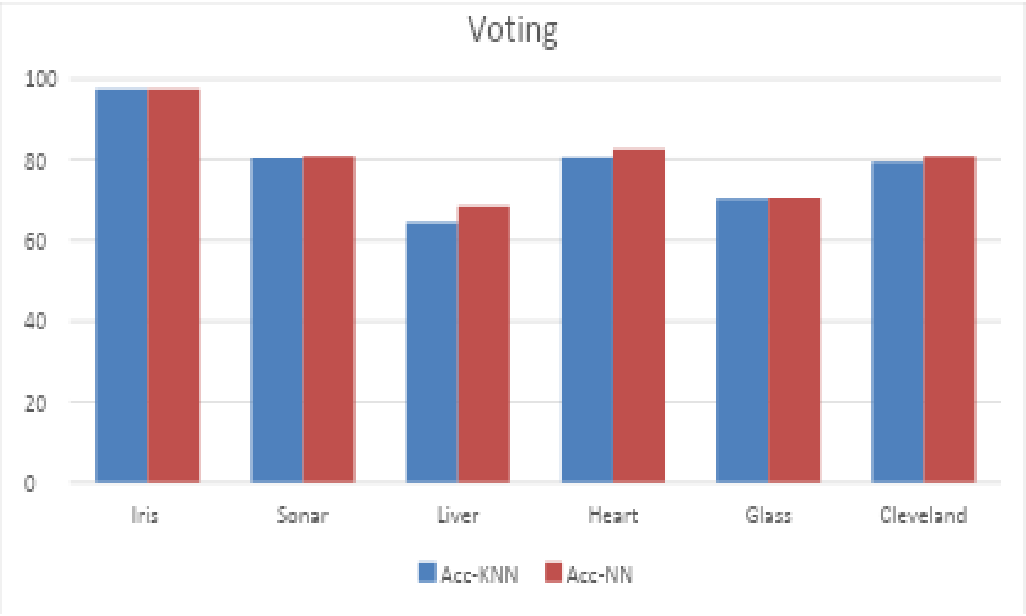



**Figure 23 Accuracy of Voting Technique with KNN and NN Classifiers**

In the voting strategy, the instance selection methods CNN, IB2, IB3, MC, and RMHC were chosen due to their higher compression ratios compared to other techniques. Including all IS methods in the voting process would result in nearly the entire dataset being retained for training. The voting-based accuracy results are presented in Figure 5.20. With the NN classifier, slight improvements in accuracy were observed for the Sonar, Heart, and Cleveland datasets, while the Liver dataset showed a more notable increase of approximately 4%.

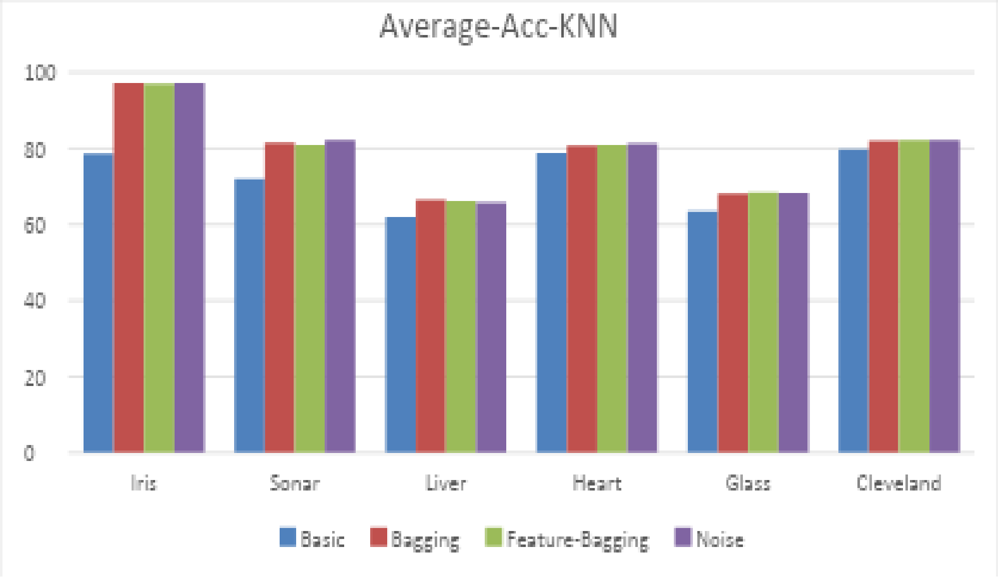



**Figure 24 Average accuracy KNN classifier**

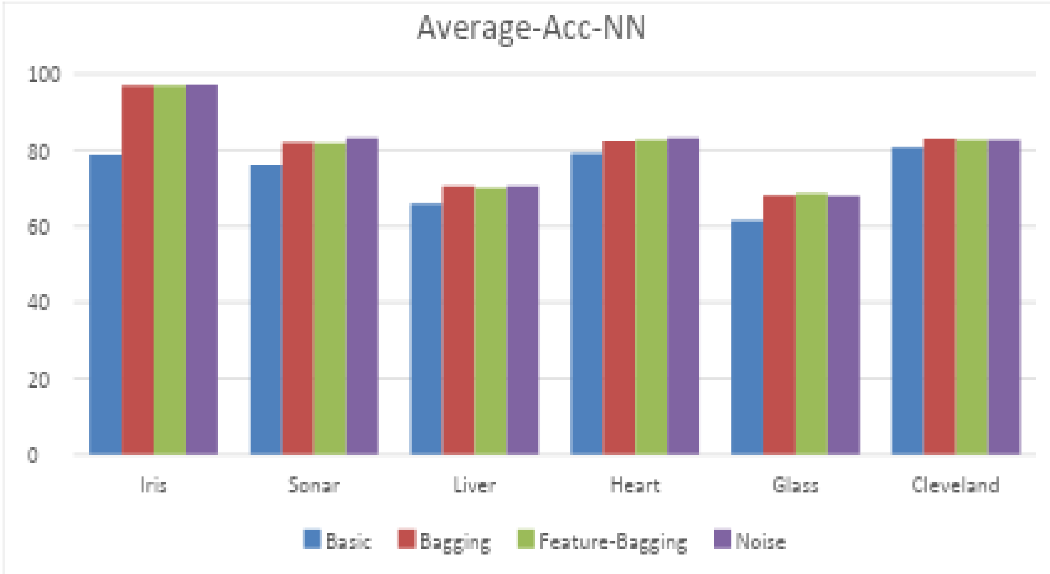



**Figure 25 Average accuracy with NN classifier**

Figures 24 and 25 illustrate the average accuracy achieved using basic methods, bagging, feature bagging, and additive noise. The ensemble instance selection (IS) methods significantly enhance dataset accuracy compared to the basic approach, as shown in the figures. Although the basic method (without ensemble) yields a higher compression ratio, its accuracy remains lower than all ensemble techniques, as indicated in Figure 5.23. Among the ensemble methods, bagging demonstrates the most effective performance. Additionally, the NN classifier consistently outperforms the KNN classifier across all setups.

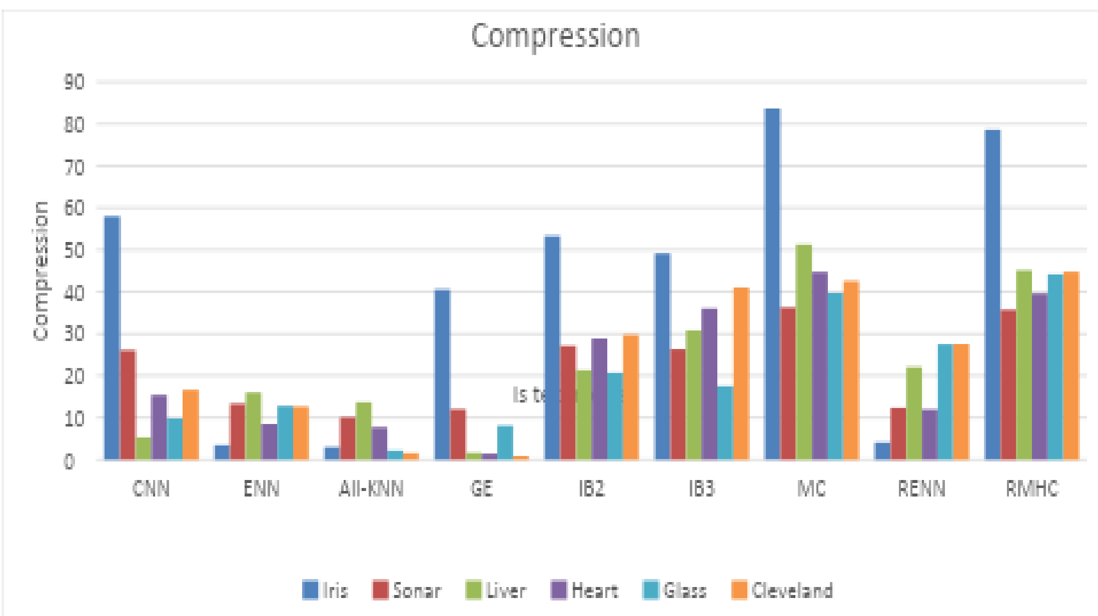



**Figure 26 Compression ratio of all datasets across IS techniques**

Figure 26 presents the compression ratios achieved for each dataset across all instance selection (IS) methods. From the analysis, it is evident that CNN, IB2, and IB3 offer satisfactory compression, while

MC and RMHC demonstrate superior compression performance compared to the others. In contrast, ENN, All-KNN, GE, and RENN show relatively poor compression efficiency.

## 5. Conclusion

In this research, we evaluated the performance of four fundamental ensemble learning strategies—Bagging, Additive Noise, Feature-Bagging, and Voting—combined with nine instance selection (IS) techniques: CNN, ENN, GE, All-KNN, IB2, IB3, RMHC, RENN, and MC, using two classifiers: KNN and NN.

The study was structured into three main phases:

- **Phase 1:** We examined the relationship between compression ratio and accuracy using the Iris dataset. Our findings showed that increasing the compression ratio typically leads to a decrease in classification accuracy.
- **Phase 2:** We assessed the impact of the four ensemble learning techniques across six datasets. Results indicated that the NN classifier consistently outperformed the KNN classifier, particularly when combined with ensemble methods. Among all ensemble techniques, Bagging achieved the highest accuracy in most cases. Additionally, the ensemble-based models outperformed the base IS methods.
- **Phase 3:** We compared the compression performance of the nine IS techniques both with and without ensemble methods. Techniques like CNN, IB2, IB3, MC, and RMHC provided significantly better data reduction compared to ENN, RENN, GE, and All-KNN. However, the results showed that no single ensemble-IS combination universally outperformed all others. Each ensemble method had strengths depending on the IS technique and dataset used.

While this study focused on four ensemble strategies, modern ensemble learning includes advanced methods such as Stacking, Boosting, and AdaBoost, which we plan to explore in future research. Additionally, our study employed nine IS techniques, all based on the wrapping method. Future work will incorporate a wider variety of IS methods, including filter-based techniques and bio-inspired approaches such as Evolutionary Algorithms (EA) and Genetic Algorithms (GA). Our experiments were limited to two classifiers—KNN and NN. Future evaluations will include more diverse classifiers like Random Forest, Naïve Bayes, and Support Vector Machines (SVM) for a broader comparison. Lastly, this study utilized relatively small datasets. For a more comprehensive evaluation, future experiments will be conducted on larger and more complex datasets to test the scalability and robustness of the proposed methods.

## References:


[1] Khan, S.U.R., Asif, S., Bilal, O. et al. Lead-cnn: lightweight enhanced dimension reduction convolutional neural network for brain tumor classification. Int. J. Mach. Learn. & Cyber. (2025). https://doi.org/10.1007/s13042-025-02637-6.

[2] Khan, S. U. R., Asim, M. N., Vollmer, S., & Dengel, A. (2025). Robust & Precise Knowledge Distillation-based Novel Context-Aware Predictor for Disease Detection in Brain and Gastrointestinal. arXiv preprint arXiv:2505.06381..

[3] Hekmat, A., et al., Brain tumor diagnosis redefined: Leveraging image fusion for MRI enhancement classification. Biomedical Signal Processing and Control, 2025. 109: p. 108040.

[4] Khan, Z., Hossain, M. Z., Mayumu, N., Yasmin, F., & Aziz, Y. (2024, November). Boosting the Prediction of Brain Tumor Using Two Stage BiGait Architecture. In 2024 International Conference on Digital Image Computing: Techniques and Applications (DICTA) (pp. 411-418). IEEE.

[5] Khan, S. U. R., Raza, A., Shahzad, I., & Ali, G. (2024). Enhancing concrete and pavement crack prediction through hierarchical feature integration with VGG16 and triple classifier ensemble. In 2024 Horizons of Information Technology and Engineering (HITE)(pp. 1-6). IEEE https://doi. org/10.1109/HITE63532.

[6] Khan, S.U.R., Zhao, M. & Li, Y. Detection of MRI brain tumor using residual skip block based modified MobileNet model. Cluster Comput 28, 248 (2025). https://doi.org/10.1007/s10586-024-04940-3

[7] Khan, U. S., & Khan, S. U. R. (2024). Boost diagnostic performance in retinal disease classification utilizing deep ensemble classifiers based on OCT. Multimedia Tools and Applications, 1-21.

[8] Raza, A., & Meeran, M. T. (2019). Routine of encryption in cognitive radio network. Mehran University Research Journal of Engineering & Technology, 38(3), 609-618.

[9] Al-Khasawneh, M. A., Raza, A., Khan, S. U. R., & Khan, Z. (2024). Stock Market Trend Prediction Using Deep Learning Approach. Computational Economics, 1-32.

[10] Khan, U. S., Ishfaque, M., Khan, S. U. R., Xu, F., Chen, L., & Lei, Y. (2024). Comparative analysis of twelve transfer learning models for the prediction and crack detection in concrete dams, based on borehole images. Frontiers of Structural and Civil Engineering, 1-17.

[11] Khan, S. U. R., & Asif, S. (2024). Oral cancer detection using feature-level fusion and novel self-attention mechanisms. Biomedical Signal Processing and Control, 95, 106437.

[12] Farooq, M. U., Khan, S. U. R., & Beg, M. O. (2019, November). Melta: A method level energy estimation technique for android development. In 2019 International Conference on Innovative Computing (ICIC) (pp. 1-10). IEEE.

**[13]** Asim, M. N., Ibrahim, M. A., Malik, M. I., Dengel, A., & Ahmed, S. (2020). Enhancer-dsnet: a supervisedly prepared enriched sequence representation for the identification of enhancers and their strength. In Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23–27, 2020, Proceedings, Part III 27 (pp. 38-48). Springer International Publishing.

**[14]** Raza, A.; Meeran, M.T.; Bilhaj, U. Enhancing Breast Cancer Detection through Thermal Imaging and Customized 2D CNN Classifiers. VFAST Trans. Softw. Eng. 2023, 11, 80–92.

**[15]** Dai, Q., Ishfaque, M., Khan, S. U. R., Luo, Y. L., Lei, Y., Zhang, B., & Zhou, W. (2024). Image classification for sub-surface crack identification in concrete dam based on borehole CCTV images using deep dense hybrid model. Stochastic Environmental Research and Risk Assessment, 1-18.

**[16]** Muhammad, N. A., Rehman, A., & Shoaib, U. (2017). Accuracy based feature ranking metric for multi-label text classification. International Journal of Advanced Computer Science and Applications, 8(10).

**[17]** Mehmood, F., Ghafoor, H., Asim, M. N., Ghani, M. U., Mahmood, W., & Dengel, A. (2024). Passion-net: a robust precise and explainable predictor for hate speech detection in roman urdu text. Neural Computing and Applications, 36(6), 3077-3100.

**[18]** Khan, S.U.R.; Asif, S.; Bilal, O.; Ali, S. Deep hybrid model for Mpox disease diagnosis from skin lesion images. Int. J. Imaging Syst. Technol. 2024, 34, e23044.

**[19]** Khan, S.U.R.; Zhao, M.; Asif, S.; Chen, X.; Zhu, Y. GLNET: Global–local CNN's-based informed model for detection of breast cancer categories from histopathological slides. J. Supercomput. 2023, 80, 7316–7348.

**[20]** Saleem, S., Asim, M. N., Van Elst, L., & Dengel, A. (2023). FNReq-Net: A hybrid computational framework for functional and non-functional requirements classification. Journal of King Saud University-Computer and Information Sciences, 35(8), 101665.

**[21]** Hekmat, Arash, Zuping Zhang, Saif Ur Rehman Khan, Ifza Shad, and Omair Bilal. "An attention-fused architecture for brain tumor diagnosis." Biomedical Signal Processing and Control 101 (2025): 107221.

**[22]** Khan, S.U.R.; Zhao, M.; Asif, S.; Chen, X. Hybrid-NET: A fusion of DenseNet169 and advanced machine learning classifiers for enhanced brain tumor diagnosis. Int. J. Imaging Syst. Technol. 2024, 34, e22975.

**[23]** Khan, S.U.R.; Raza, A.;Waqas, M.; Zia, M.A.R. Efficient and Accurate Image Classification Via Spatial Pyramid Matching and SURF Sparse Coding. Lahore Garrison Univ. Res. J. Comput. Sci. Inf. Technol. 2023, 7, 10–23.

**[24]** Farooq, M.U.; Beg, M.O. Bigdata analysis of stack overflow for energy consumption of android framework. In Proceedings of the 2019 International Conference on Innovative Computing (ICIC), Lahore, Pakistan, 1–2 November 2019; pp. 1–9.

**[25]** Shahzad, I., Khan, S. U. R., Waseem, A., Abideen, Z. U., & Liu, J. (2024). Enhancing ASD classification through hybrid attention-based learning of facial features. Signal, Image and Video Processing, 1-14.

**[26]** Khan, S. R., Raza, A., Shahzad, I., & Ijaz, H. M. (2024). Deep transfer CNNs models performance evaluation using unbalanced histopathological breast cancer dataset. Lahore Garrison University Research Journal of Computer Science and Information Technology, 8(1).

**[27]** Bilal, Omair, Asif Raza, and Ghazanfar Ali. "A Contemporary Secure Microservices Discovery Architecture with Service Tags for Smart City Infrastructures." VFAST Transactions on Software Engineering 12, no. 1 (2024): 79-92.

**[28]** Khan, S. U. R., Asif, S., Zhao, M., Zou, W., Li, Y., & Li, X. (2025). Optimized deep learning model for comprehensive medical image analysis across multiple modalities. Neurocomputing, 619, 129182.

**[29]** Khan, S. U. R., Asif, S., Zhao, M., Zou, W., & Li, Y. (2025). Optimize brain tumor multiclass classification with manta ray foraging and improved residual block techniques. Multimedia Systems, 31(1), 1-27.

**[30]** Khan, S. U. R., Asim, M. N., Vollmer, S., & Dengel, A. (2025). AI-Driven Diabetic Retinopathy Diagnosis Enhancement through Image Processing and Salp Swarm Algorithm-Optimized Ensemble Network. arXiv preprint arXiv:2503.14209.

**[31]** Khan, Z., Khan, S. U. R., Bilal, O., Raza, A., & Ali, G. (2025, February). Optimizing Cervical Lesion Detection Using Deep Learning with Particle Swarm Optimization. In 2025 6th International Conference on Advancements in Computational Sciences (ICACS) (pp. 1-7). IEEE.

**[32]** Khan, S.U.R., Raza, A., Shahzad, I., Khan, S. (2025). Subcellular Structures Classification in Fluorescence Microscopic Images. In: Arif, M., Jaffar, A., Geman, O. (eds) Computing and Emerging Technologies. ICCET 2023. Communications in Computer and Information Science, vol 2056. Springer, Cham. https://doi.org/10.1007/978-3-031-77620-5_20

**[33]** Hekmat, A., Zuping, Z., Bilal, O., & Khan, S. U. R. (2025). Differential evolution-driven optimized ensemble network for brain tumor detection. International Journal of Machine Learning and Cybernetics, 1-26.

**[34]** Khan, S. U. R. (2025). Multi-level feature fusion network for kidney disease detection. Computers in Biology and Medicine, 191, 110214.

**[35]** Khan, S. U. R., Asif, S., & Bilal, O. (2025). Ensemble Architecture of Vision Transformer and CNNs for Breast Cancer Tumor Detection From Mammograms. International Journal of Imaging Systems and Technology, 35(3), e70090.

**[36]** Khan, S. U. R., & Khan, Z. (2025). Detection of Abnormal Cardiac Rhythms Using Feature Fusion Technique with Heart Sound Spectrograms. Journal of Bionic Engineering, 1-20.

**[37]** Khan, M.A., Khan, S.U.R. & Lin, D. Shortening surgical time in high myopia treatment: a randomized controlled trial comparing non-OVD and OVD techniques in ICL implantation. BMC Ophthalmol 25, 303 (2025). https://doi.org/10.1186/s12886-025-04135-3

**[38]** Shahzad, I., Raza, A., & Waqas, M. (2025). Medical Image Retrieval using Hybrid Features and Advanced Computational Intelligence Techniques. Spectrum of engineering sciences, 3(1), 22-65.

**[39]** Raza, A., Shahzad, I., Ali, G., & Soomro, M. H. (2025). Use Transfer Learning VGG16, Inception, and Reset50 to Classify IoT Challenge in Security Domain via Dataset Bench Mark. Journal of Innovative Computing and Emerging Technologies, 5(1).

**[40]** Raza, A., & Shahzad, I. (2024). Residual Learning Model-Based Classification of COVID-19 Using Chest Radiographs. Spectrum of engineering sciences, 2(3), 367-396.

**[41]** Raza, A., Soomro, M. H., Shahzad, I., & Batool, S. (2024). Abstractive Text Summarization for Urdu Language. Journal of Computing & Biomedical Informatics, 7(02).

**[42]** HUSSAIN, S., RAZA, A., MEERAN, M. T., IJAZ, H. M., & JAMALI, S. (2020). Domain Ontology Based Similarity and Analysis in Higher Education. IEEEP New Horizons Journal, 102(1), 11-16.

**[43]** Raza, A., & Meeran, M. T. (2019). Routine of encryption in cognitive radio network. Mehran University Research Journal of Engineering & Technology, 38(3), 609-618.

**[44]** Asim, M. N., Ibrahim, M. A., Zehe, C., Cloarec, O., Sjogren, R., Trygg, J., ... & Ahmed, S. (2021, July). L2S-MirLoc: a lightweight two stage MiRNA sub-cellular localization prediction framework. In 2021 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

**[45]** Saleem, S., Asim, M. N., Van Elst, L., Schichtel, P., & Dengel, A. (2024). Rprp-sap: A robust and precise resnet predictor for steering angle prediction of autonomous vehicles. IEEE Access, 12, 21472-21490.

**[46]** Khan, U.S., Khan, S.U.R.: Ethics by Design: A Lifecycle Framework for Trustworthy AI in Medical Imaging From Transparent Data Governance to Clinically Validated Deployment (2025). https://arxiv.org/abs/2507.04249

**[47]** Maqsood, H., & Khan, S. U. R. (2025). MeD-3D: A Multimodal Deep Learning Framework for Precise Recurrence Prediction in Clear Cell Renal Cell Carcinoma (ccRCC). arXiv preprint arXiv:2507.07839.

**[48]** Khan, S. U. R., Asim, M. N., Vollmer, S., & Dengel, A. (2025). FOLC-Net: A Federated-Optimized Lightweight Architecture for Enhanced MRI Disease Diagnosis across Axial, Coronal, and Sagittal Views. arXiv preprint arXiv:2507.06763.

**[49]** Bilal, O., Hekmat, A. & Khan, S.U.R. Automated cervical cancer cell diagnosis via grid search-optimized multi-CNN ensemble networks. Netw Model Anal Health Inform Bioinforma 14, 67 (2025). https://doi.org/10.1007/s13721-025-00563-9

**[50]** M. Wajid, M. K. Abid, A. Asif Raza, M. Haroon, and A. Q. Mudasar, “Flood Prediction System Using IOT & Artificial Neural Network”, VFAST trans. softw. eng., vol. 12, no. 1, pp. 210–224, Mar. 2024.

**[51]** M. Waqas, Z. Khan, S. U. Ahmed and A. Raza, "MIL-Mixer: A Robust Bag Encoding Strategy for Multiple Instance Learning (MIL) using MLP-Mixer," 2023 18th International Conference on Emerging Technologies (ICET), Peshawar, Pakistan, 2023, pp. 22-26. DOI: 10.1109/ICET59753.2023.10374927.

**[52]** Mahmood, F., Abbas, K., Raza, A., Khan,M.A., & Khan, P.W. (2019 ). Three Dimensional Agricultural Land Modeling using Unmanned Aerial System (UAS). International Journal of Advanced Computer Science and Applications (IJACSA) [p-ISSN : 2158-107X, e-ISSN : 2156-5570], 10(1).

**[53]** Meeran, M. T., Raza, A., & Din, M. (2018). Advancement in GSM Network to Access Cloud Services. Pakistan Journal of Engineering, Technology & Science [ISSN: 2224-2333], 7(1).