Enhancing Question Pairs Identification with Ensemble Learning: Integrating Machine Learning and Deep Learning Models

Salsabil Tarek[1*], Hatem M. Noaman[2], Mohammed Kayed[3]
Computer Science Department, Faculty of Computer Science, Nahda University, Beni-Suef 62511, Egypt[1]
Computer Science Department, Faculty of Computers and Artificial Intelligence,
Beni-Suef University, Beni Suef 62511, Egypt[2, 3]

***Abstract*—The effectiveness of machine learning (ML) and deep learning (DL) models on the Quora question pairs dataset is investigated in this study. ML models, including AdaBoost, reached 73.44% test accuracy, while ensemble learning approaches enhanced outcomes even further, with the Hard-Voting Ensemble achieving 76.13%. DL models, such as FCN, demonstrated test accuracy of 81% with cross validation. These findings contribute to natural language processing by demonstrating the potential of ensemble learning for ML models and the DL models' detailed pattern-capturing capacity.**

***Keywords*—Ensemble learning; natural language processing; deep learning; machine learning**

I. INTRODUCTION

Ensemble learning has grown increasingly popular as an efficient machine learning technique to increase accuracy and predictability in predictions. At its core, ensemble learning entails merging multiple models into a more accurate predictor [1]. There are various techniques for producing multiple models simultaneously. Bagging is one such approach [2], where multiple models are trained on random subsets of training data for multiple models to share a set of results. Another technique, called boosting [3], allows models to be trained sequentially to address errors from prior models. A third technique called stacking [4] uses multiple models trained on identical data and employs a meta-model as a bridge to integrate their outputs. Stacking is a method by which multiple models are trained to find an ideal way of combining their predictions. Predictions from base models serve as input into an intermediate-level model which then learns how to weigh and combine them to produce a final prediction.

Ensemble learning offers numerous advantages over single models. It reduces the risk that an overfitted model becomes too complex and learns noise instead of patterns in data; and captures more patterns. Ensemble learning can also enhance the predictability and stability of predictions by creating a model less sensitive to small fluctuations in data. Furthermore, ensemble learning combines all models' strengths for improved accuracy in forecasts. Ensemble learning in machine learning refers to the practice of combining multiple models into one to increase accuracy and robustness. This involves training multiple models on one dataset using different initializations or hyperparameters for their training sessions. Ensemble learning's central concept is that combined models will outperform individual ones due to being better at capturing more patterns while avoiding overfitting. Ensemble learning in natural language processing has yielded excellent results for a range of tasks such as text classification and sentiment analysis. Ensemble learning can easily manage various data types - textual as well as structured data - making it applicable to many NLP tasks. Selecting the optimal ensemble method and configuration can be a complex process that involves extensive experimentation and evaluation. Furthermore, training multiple models may prove too expensive a prospect; hence ensemble learning has proven an indispensable asset to NLP applications. Recent machine learning studies have demonstrated the power of ensemble learning over individual classifiers when it comes to improving performance. Ensemble learning has had a considerable effect on machine learning applications, leading to its widespread usage across various domains such as text classification. [5- 8]. Deep neural networks (DNN), one of the cornerstones of machine learning, have emerged as a formidable force over recent years. DNNs have contributed to advancing natural language processing and text classification techniques. Speech recognition, object detection, visual object recognition, and object identification all benefit. [9]. Deep learning techniques differ from classical machine learning in that they automatically identify and extract complex features without manually creating them [10]. Deep learning employs multiple network architectures to address problems, including feed-forward neural nets, convolutional networks, and recurrent networks [11]. Recently, many attempts have been made to combine DNNs and ensemble methods to enhance prediction performance. To develop ensemble deep learning, one of the easiest and simplest approaches is integrating deep learning directly into existing ensemble learning methods. Most attempts focus on creating weighted-average models of deep learning models; studies have demonstrated that ensembles incorporating DNNs outperform individual DNNs for classification tasks [12].

The Quora Question-Pairs dataset is one of the most frequently utilized resources for identifying question pairs. With over 400,000 questions identified as either duplicates or not duplicates, this dataset offers an excellent way to pinpoint question pairings. Duplicate questions must be identified to reduce redundancy on search engines, forums, and question-answering software. Unfortunately, due to its wide range of languages and question structures, this task is no easy feat. This

study investigates the efficacy of ensemble learning methods in recognizing question pairs from the Quora Question Pairs data set. Various ensemble methods were compared to improve the accuracy of machine learning models as well as deep learning models. Current experiments demonstrate how ensemble learning can be accomplished by combining models such as logistic regression, random forests, and XGBoost with deep learning models like convolutional neural networks or long-short-term memory networks. This study will conduct further investigations of hyperparameters and assess the interpretability of ensemble models, contributing to an expanding body of research in ensemble learning and natural language processing (NLP), providing insight into optimal ensemble methods and configurations to identify question pairs; these results may also have applications in search engines, online forums and question answering software systems.

This research performs comparative experiments on the dataset using the most popular ensemble techniques; weighted and vote ensemble methods, which contribute to and encompass the development of ensemble learning algorithm, extensive experimentation with various machine learning models, performance evaluation, comparative analysis against existing methods, and an exploration of prediction strategies within the context of ensemble learning for semantic similarity. To that purpose, the following are the primary contributions of the paper:

In current study, comprehensive trials were carried out by cross-validating training several machine learning models, followed by precise evaluations. This strategy demonstrates the ensemble methods underlying potential for assessment the performance of models on the dataset of Quora question pairings. Ensemble learning approaches using a varied variety of machine learning models had not been investigated on the Quora question pairs dataset.

This analysis using Current Ensemble Techniques is expanded to evaluate the effectiveness of ensemble approach with a number of well utilized ensemble techniques. The competitiveness and benefits in the context of semantic similarity are shown in this comparative analysis.

This research covers a comparison by applying deep learning to the Quora question pairs dataset which enables us to illustrate deep learning's performance and natural benefits in dealing with the issues given by question pair classification tasks that providing a complete evaluation of model performance and emphasizing the practical value of these gains in tackling real-world situations.

This paper is organized as follows. Section II presents the related works. Data preparation is shown in Section III. Section IV discusses the proposed model. While the results and discussion are given in Section V. Finally, Section VI concludes the current research.

II. Related Work

Several studies as shown in Table I, have investigated the effectiveness of different ensemble methods, such as bagging, boosting, and stacking, in improving the performance. Study by Dhakal et al. [13] focused on used Natural Language Processing to address the issue of question duplication in Q&A forums by using Deep Learning to determine whether question pairings are duplicates. Sharma et al. research [14] investigated the task of Natural Language Understanding (NLU) through the analysis of duplicate questions in the Quora dataset. They explored the dataset extensively and applied a variety of machine learning models, including linear and tree-based models. To overcome the duplicate question problem provided by the Quora dataset, they tried an enormous number and variety of machine learning models. A basic Continuous Bag of Words neural network performed the best, they also performed error analysis and discovered some subjectivity in the dataset's labelling.

Chandra and Stefanus [15] modelled the Quora question pairings dataset to find a related question; The assignment is a binary categorization. They attempted several methodologies and algorithms, as well as a distinct approach from earlier efforts. For XGBoost and CatBoost, they employed Bag of Words with Count Vectorizer and Term Frequency-Inverse Document Frequency with Unigram for feature extraction. Furthermore, they tested the WordPiece tokenizer, which considerably increases model performance, and they were able to get up to 97 percent accuracy. They tested Bag of Words with two boosting algorithms: Catboost and XGBoost. They also used simple LSTM and BERT to evaluate Quora Question Pairs. The results reveal that BERT outperformed the other models.

The goal of research done by Sharma et al. [16] was to determine whether a question pair is similar; they used a dataset provided by Quora on Kaggle to accomplish this. That dataset had four lakh records, which assisted us in training their models and obtaining the necessary outcomes. They employed Natural Language Processing knowledge and different classification and boosting techniques to determine which is more useful, then they examined the accuracy of various models to determine which method is best suited for the task. The same has been done with the aid of multiple graphs and tables to highlight the differences in the accuracy of various algorithms. It was critical to clean and pre-process the data before applying any algorithm. After that, they used techniques such as the Count Vectorizer with XG Gradient Boosting, the TF-IDF Vectorizer with XG Gradient Boosting, Logistic Regression, and Random Forest.

Anishaa et al. [17] proposed a novel approach by filtration of the Quora datasets using SQLite which takes one-quarter the time it takes to pre-process the same dataset using existing methodologies such as python functions. It concluded that XGBoost outperformed the other machine learning approaches discussed, it has also been discovered that pre-processing with SQLite has improved response time. To analyses and find the best model, they employed machine learning techniques such as Random Forest, Logistic Regression, Linear SVM (Support Vector Machine), and XGBoost. The error log loss functions (0.887, 0.521, 0.654, and 0.357) of the machine learning algorithms were analyzed and compared. XGBoost has the best performance among the other models.

Chandra and his colleagues [18] provided a technique for detecting duplicate question pairs in their study by dividing the

selected dataset in a 70:30 ratio. A technique known as random splitting was employed. It was discovered that if a feature timestamp for each question was given, then a time-based splitting might be utilized to partition the dataset, because the questions asked earlier differed greatly from the questions answered recently. Enabling this feature increases accuracy. The model uses the Glove pre-embedding to classify the questions. Features such as fuzzywuzzy help to achieve very minimal log loss. The log loss for the XGBoost model was 0.35, while the log loss for the Siamese LSTM model was 0.21.

Furthermore, Gontumukkala et al. [19] proposed a method to overcome two drawbacks of Quora as the occurrence of duplicate questions that cause ambiguity and insincere questions that lessen the value of the site by suggesting a strategy to address these two issues using Deep Learning (DL) and Natural Language Processing (NLP) approaches. Bi-directional Long Short-Term Memory (BiLSTM) and Bi-Gated Recurrent Unit (BiGRU) architectures with attention mechanisms were used for both problems, and Siamese Manhattan Long Short-Term Memory (MaLSTM) architectures were used for question pair identification. Five different word embeddings were used for each problem. When it comes to accuracy, precision, recall, and F1 Score, the models that have been used are performing well. For the classification of sincere questions, their model achieved the highest accuracy of 95% and the highest F1 score of 0.82 using FastText + BiLSTM + BiGRU. For the identification of Quora question pairs, their research work achieved the highest accuracy of 90% and the highest F1 score of 0.89 using Paraphrase-MiniLM-L6-v2 + Siamese MaLSTM.

Sendi et al. [20] introduced a transparent, deep ensemble classification method based on multiagent arguments. This approach leverages deep learning algorithms combined with argumentation to outperform traditional ensemble methods, providing explain-ability while meeting Explainable AI needs. Furthermore, Mohammed and Kora [21] proposed a novel ensemble meta-learning strategy which combines multiple classifiers was proposed.

Karlos and his colleagues [22] presented the proposed ensemble method outshone other ensemble methods on benchmark datasets in terms of performance. Furthermore, its meta-learner's performance was further improved by taking advantage of probability distributions for class labels. This paper describes an ensemble-based training scheme for binary classifying using random feature splitting.

Gonçalves et al. [23] assessed the effectiveness of a multi-view ensemble for full-text classification using different document sections as views. Results demonstrate its accuracy in classification accuracy and F1-score calculations. C4.5 serves as their meta-learner to implement support vector machine algorithms for stacking. For views creation, they utilized the OHSUMED full-text biomedical dataset; results from experiments demonstrate that multi-view techniques significantly improve text classification within biomedical text mining. Findings indicate that adding text from certain sections to datasets outperforms simply using titles and abstracts alone.

Haghighi and Omranpour [24] offered an ensemble classifier stacking model to recognize handwritten digits. Addressing different writing styles and structural similarities among digits, this model uses a convolutional network (CNN) paired with bidirectional long-short-term memory (BLSTM) to unify both methods. It utilizes the innovative use of image class probability vectors as input to the meta-classifier, further increasing accuracy with its deep-learning model through BLSTM's ability to learn vectors and arrays. Stacking ensemble classification helps reduce recognition errors by considering similarities between Persian/Arabic numbers and writing style variations. The model was tested on a large dataset consisting of 102.352 points from 102.352 classes of Persian/Arabic data. It achieved high accuracy rates of 99.98% for the training set and 99.39% for the test set. These results demonstrate enhanced performance compared with convolutional neural network experiments and previous research.

Araque et al. [25] investigated ways of improving performance using both deep learning techniques and traditional surface approaches for Sentiment Analysis. Deep learning offers advantages over surface approaches in terms of automatic feature extraction and richer representation abilities. This paper features six contributions; as an initial task, a deep-learning-based sentiment classifier using word embedding is constructed as a baseline solution. Second, two ensemble techniques combine the baseline with other surface classifications commonly employed for Sentiment Analysis. Thirdly, they introduce two models that leverage data from multiple sources by combining surface and deep features. Fifthly, a taxonomy that classifies all proposed models is presented. Seven datasets from microblogging and movie reviews domains are utilized to conduct various experiments that compare performance between proposed models and baseline deep learning systems. An F1-Score analysis verifies the performance of their proposed models.

A study done by Onan et al. [26] implemented a multi objective voting scheme for sentiment analysis that uses optimization. The ensemble method incorporates a static classifier, majority voting errors, forward search, and multi objective differentiation evolution algorithm. Base learners include Bayesian log regression, naive Bayes (linear discriminant analysis), logistic regression, and support vector machine while the current method outshone ensemble learning techniques in various classification tasks. Ankit and Saleena [27] offered a Twitter Sentiment Analysis which detects sentiments and opinions within tweets. To achieve accurate classification of tweets they selected an accurate classifier. They consider common base classifiers such as Naive Bayes and Random Forest, SVMs, and Logistic Regression as base classifiers. An ensemble classifier combining all these classifiers is then proposed to improve performance and accuracy.

Convolutional Neural Networks (CNN) [28] used to identify the semantic similarity of questions using the Quora question pairs dataset. Glove pre-trained word embedding applied to identify the semantic similarity between queries. This word embedding vector is fed into CNN, and the results are compared to Siamese Neural Networks. The model achieved an accuracy of 79%. Wang et al. [29] used the Stack Overflow dataset to investigate three deep learning algorithms

to identify duplicate questions: DQ-CNN, DQ-RNN, and DQ-LSTM, which are based on CNN, RNN, and LSTM, respectively. Six distinct question groups are used to evaluate the effectiveness of DQ-CNN, DQ-RNN, and DQ-LSTM. Except for the Ruby question group, their experimental results reveal that DQ-LSTM outperforms DupPredictor, Dupe, DupePredictorRepT, and DupeRep in terms of recall-rate@5, recall-rate@10, and recall-rate@20.

TABLE I. SUMMARY OF THE PREVIOUS WORKS

MODEL	Accuracy	Precision	Recall	F1-score
Supervised Machine Learning Algorithm [13]				
Random Forest [13]	0.741	-	-	-
Logistic Regression [13]	0.677	-	-	-
Decision Tree [13]	0.683	-	-	-
Support Vector Machine [13]	0.542	-	-	-
K Nearest Neighbors[13]	0.719	-	-	-
Multinomial Naive Bayes [13]	0.673	-	-	-
Most frequent class [14]	63.1	-	-	-
LR with Unigrams[14]	75.4	-	-	63.8
LR with Bigrams[14]	79.5	-	-	70.6
Linear LR with Trigrams[14]	80.8	-	-	71.8
LR with Trigrams, tuned[14]	80.1	-	-	71.5
SVM with Unigrams[14]	75.9	-	-	63.7
SVM with Bigrams[14]	79.9	-	-	70.5
SVM with Trigrams[14]	80.9	-	-	72.1
Tree-Based Decision Tree[14]	73.2	-	-	65.5
Random Forest[14]	75.7	-	-	66.9
Gradient Boosting[14]	75.0	-	-	66.5
CBOW[14]	**83.4**	-	-	**77.8**
LSTM[14]	81.4	-	-	75.4
LSTM + Attention [14]	81.8	-	-	75.5
BiLSTM[14]	82.1	-	-	76.2
BiLSTM + Attention [14]	82.3	-	-	76.4
CV-XGBoost [15]	68.09	-	-	-
CV-CatBoost [15]	74.66	-	-	-
TF-IDF-XGBoost [15]	69.14	-	-	-
TF-IDF CatBoost [15]	**75.39**	-	-	-
XGB [16]	0.79	0.80	0.80	-
Logistic regression [16]	0.74	0.75	0.73	-
SGDC [16]	0.74	0.73	0.75	-
Random forest [16]	0.83	0.81	0.82	-
XGBoost [18]	69%	0.79	0.69	0.73
Paraphrase- MiniLM-L6-v2 + Siamese MaLSTM [19]	90%	0.85	0.94	0.89
LSTM [30]	83.8%	-	-	0.79
BiLSTM + Frame-GBDT [31]	87.92%	-	-	-
Neural Networks + Multi-head Attention [32]	86.83%	0.84	0.81	0.82
Siamese LSTM [33]	82.77%	0.79	0.70	0.75
XG Boost [34]	81%	-	-	-
BERT Model [35]	80%	-	-	-

III. DATASET

In this study, experiments were conducted on the Quora Question Pairs dataset, which is widely used for question pairs identification tasks. The dataset consists of 404,290 question pairs, each identified by a unique ID. For each question pair, the dataset provides the question IDs (qid1 and qid2), the actual text of the first question (question1), the actual text of the second question (question2), and a binary label indicating whether the questions are duplicates (1) or not duplicates (0).

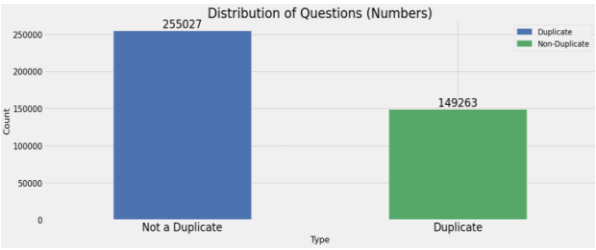

Fig. 1. The distribution of questions in the QQP dataset.

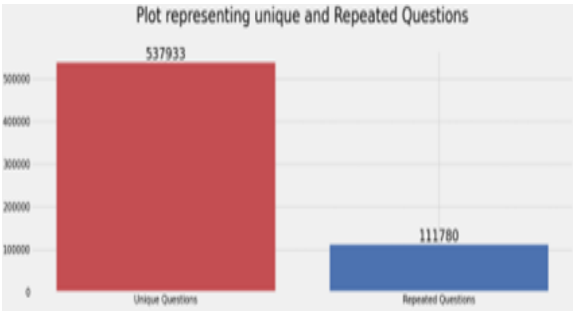

Fig. 2. Unique and repeated questions in the QQP dataset.

Fig. 1 shows a notable aspect of the dataset is the distribution of question pairs. Approximately 63.08% of the pairs are labeled as not similar or non-duplicates, while 36.92% are labeled as similar or duplicates. This class imbalance should be taken into consideration during the data preprocessing and model training stages.

One interesting idea were explored during the data preparation stage is the identification of unique and repeated questions as shown in Fig. 2. By analyzing the dataset, discovered that 98% of the questions occur only once, implying that most questions do not repeat themselves. The dataset also revealed that the maximum number of times a question is repeated is fifty.

Understanding the distribution and uniqueness of questions provides valuable insights into a dataset, aiding in designing appropriate preprocessing techniques and sampling strategies.

Cross-validation was used to assess the performance of models on the Quora Question Pairs dataset. The code snippet illustrates how to select and initialize various machine learning algorithms, such as Gaussian Naive Bayes (GNB), Logistic Regression (LR), Stochastic Gradient Descendant, Decision Tree (DT), Random Forest AdaBoost Extra Trees. Deep Learning classifiers included Fully Connected Networks (FCN), LSTM Bidirectional LSTM.

As part of models' assessment, stratified 10-fold cross-validation were used to compare models. This technique ensures each fold maintains an equal distribution of classes to that found in the original dataset and minimizes potential bias. Accuracy scores were used to judge each classifier's performance; files containing this information allow for thorough comparisons among them.

Cross-validation provides us with a powerful way to evaluate the performance of simple models for question pair identification using the Quora dataset. Reliable estimates of each model's accuracy allow us to make informed choices about suitability for ensemble learning; the results of which will inform future hyperparameter tuning and model selection steps leading to an ideal ensemble model to identify question pairs.

IV. PROPOSED MODEL

This paper proposes a model of a schematic representation architecture that uses ensemble approaches on the QQP dataset to improve the accuracy and reliability of question pair similarity prediction as provided in Fig. 3. This architecture integrates their predictions to provide strong and comprehensive similarity evaluations, with the goal of capitalizing on the strengths of varied base models. The ensemble technique tries to increase predictive accuracy while also establishing a more robust foundation for question pair analysis by synthesizing multiple perspectives on question relatedness. This architecture advances the subject of question similarity assessment within the context of the Quora dataset by integrating several modelling strategies and evaluating their combined efficacy.

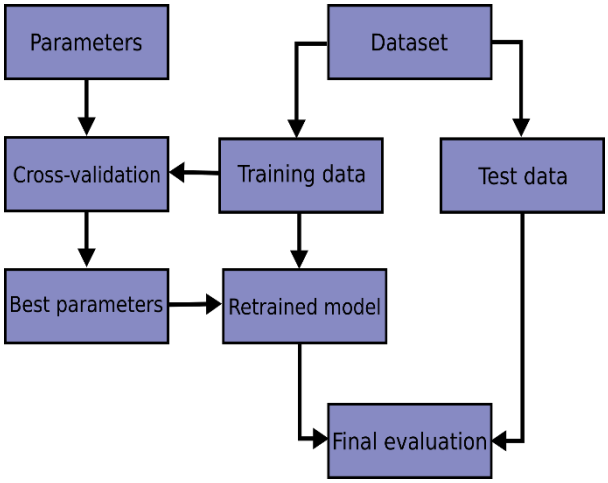

Fig. 3. Proposed general model architecture.

At this stage, text data was processed in various ways to extract features that would aid in the modeling process. The text was initially tokenized using the Tokenizer module of TensorFlow Keras preprocessing modules. This process converted sentences to integer sequences according to word indexes; tokenized sequences were then either extended or reduced until reaching 25 words long. Pre-trained GloVe word embeddings were employed to capture semantic information present in text documents, using word-to-vector maps provided. By iterating over the word index, a matrix of word

embeddings was produced, initially initialized with zeroes before gradually being filled up by GloVe embeddings' word vectors. This word embedding matrix was employed to transform text data, using tokenized and padded question data to fit neural network input requirements for model training. These features were denoted q1_data (original data) and q2_data (transformed version of data), then stored for training use.

Ensemble Learning with Machine Learning Classifiers Approach

Ensemble learning techniques are powerful tools that can increase model performance by combining predictions from different base models into one cohesive prediction. In this research study [36], voting ensemble and weighted average were employed as ensemble techniques.

The Voting Ensemble Technique involves aggregating individual model predictions to produce a final result. A hard voting technique was employed, in which the model with the highest number of votes was selected to predict class. This collective decision-making enabled more effective performances compared to any one model within an ensemble.

Soft voting was utilized, taking into account the probabilities associated with each class prediction and adding up each label's predicted probabilities to determine which class had the highest predicted probability. This method also factored in confidence levels associated with each model prediction to enhance ensemble predictive abilities.

Simple averaging the ensemble techniques was used as a simple averaging approach or weighted averaging. This approach averages the predicted values across different base models using element-wise averages and was utilized in both classification and regression tasks to provide more accurate prediction models from diverse models.

This section presents an in-depth overview of the machine learning (ML) classifiers utilized in this ensemble framework to identify question pairs. These algorithms include Gaussian Naive Bayes (GNB), Logistic Regression (LR), Stochastic Gradient descent, Decision Tree Random Forest, and Gradient Boosting Machine.

Gaussian Naive Bayes, a probabilistic classification method that utilizes Bayes' theorem and assumes independence of features, is derived from Bayesian Naive Bayes. Logistic Regression, on the other hand, is a linear model widely used to estimate probabilities associated with binary outcomes using logarithmic functions. Stochastic Gradient Descent optimizer model parameters iteratively using random subsets from the training data, while Decision Tree creates an interwoven tree-like structure by recursively partitioning features into feature splits. Random Forest employs ensemble averaging to combine multiple decision trees into an ensemble for improved prediction performance, while AdaBoost trains weak learners iteratively by assigning greater weights to instances that have been misclassified, and Extra Trees creates a group of random decision trees to improve generalization. Gradient Boosting Machine creates an ensemble by continuously adding models that correct previous errors.

The dataset was examined using an ensemble learning approach, employing various machine learning models. The dataset was divided into two sets for analysis - X_train (training) and X_test (testing), with target variables created as copies of these two groups Y_train and Y_test, respectively.

For evaluation purposes, widely popular classifiers were trained such as Gaussian Naive Bayes, Logistic Regression, Stochastic Gradient descent, Decision Tree, and Random Forest classifiers as well as AdaBoost Extra Trees Gradient Boosting Machine and Random Forest to compare models' performances against each other. Each of the classifiers employed its respective hyperparameters and training algorithms to train on training data before being deployed on testing data to predict target variables.

An ensemble prediction was created by combining predictions from each model using a simple average technique and then comparing these predictions against labels to calculate accuracy scores. Furthermore, performance metrics such as precision, recall, and F1 scores; specificity loss logs; ROC scores; Cohen's Kappa coefficient of correlation values were calculated and recorded.

Voting Classifier was used to implement a voting-based ensemble. Two variations, hard and soft voting ensembles were explored; hard voting uses majority voting to combine individual classifier predictions; while soft voting used probabilities weighted according to each classifier's confidence in its prediction. Both ensembles were evaluated on the accuracy, classification reports, and confusion matrices for evaluation. A data frame (score) was produced to summarize the results, detailing each model and ensemble's performance metrics.

A. Deep Learning Approach

The current approach also integrates Deep learning (DL) classifiers such as Fully Connected Networks, Long Short-Term Memory, and Bidirectional LSTM into its repertoire. FCN is an architecture of neural networks with fully connected layers; typically used for classification tasks. LSTM is a recurrent neural network type capable of modeling long-term dependencies within sequential data, while Bidirectional LSTM adds context information from past and future inputs by processing sequences both forwards and backward simultaneously. This ensemble framework harnessed their complementary properties and strengths in combination with each other to increase accuracy and robustness for question pair identification tasks. Finally, experimental results were presented as well as evaluating their performance.

As part of the current experimental setup, this study analyzed this dataset using deep learning methods, demonstrating several neural network models including Fully Connected Network, Long Short-Term Memory network (LSTM), and Bidirectional LSTM models. a Time Distributed Layer was used, which employs deep learning architecture to classify questions as duplicates or not. Below is an outline of its implementation and evaluation process.

Initializing all variables and data structures. Next, the dataset was split into two sets - training and testing. Within the training set there can also be further subdivided into five folds

to facilitate cross-validation. The current model was then built using Keras' functional API for flexible architecture consisting of two input layers for every pair of questions posed to it.

The embedding layer transforms words into dense vectors of fixed size for every question, using pre-trained embeddings of words to capture semantic data. Output from this embedding is fed into a Time Distributed Layer followed by a Max Pooling operation to produce fixed-length representations of each question, before being concatenated together and passed through several dense layers such as batch normalization and dropout regularization to reach completion.

The final layer is a dense layer with a sigmoid activation function. This layer produces a score that indicates the likelihood of two questions being identical.

A 5-fold cross-validation approach is used to evaluate the model. Once trained using a set number of epochs and callbacks are implemented to save weights based on validation accuracy, then tested against a test set to evaluate accuracy and loss of prediction.

The algorithm incorporates visual elements like confusion matrices and learning curves to get a general sense of how well its model is performing. Other metrics, such as precision, recall, F1 scores and specificity metrics were used to evaluate the classification performance of the models.

TABLE II. UTILIZING CROSS VALIDATION FOR COMPARATIVE ANALYSIS OF SIMPLER MODELS ACCURACIES

MODEL	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold 10
Naive Bayes	0.67119	0.67324	0.67324	0.67409	0.67698	0.67641	0.67446	0.67347	0.67273	0.66998
Logistic Regression	0.70775	0.70867	0.70860	0.70701	0.71157	0.71078	0.71273	0.70778	0.71071	0.70859
Stochastic Gradient Descent	0.71214	0.71026	0.71344	0.71365	0.72298	0.71269	0.71492	0.70891	0.71121	0.71615
Decision Tree	0.72171	0.72114	0.72160	0.72051	0.72415	0.72191	0.72212	0.72583	0.72068	0.72290
Random Forest	**0.77569**	**0.77728**	**0.77803**	**0.77545**	**0.78092**	**0.77473**	**0.77837**	**0.77918**	**0.78042**	**0.77826**
AdaBoost	0.73220	0.73676	0.72765	0.73408	0.73429	0.73389	0.74032	0.73421	0.73672	0.73933
Extra Trees	0.76806	0.77068	0.77414	0.76866	0.77598	0.76883	0.77569	0.77402	0.77600	0.77378
Gradient Boosting Machine	0.75418	0.75188	0.75435	0.75654	0.75813	0.75283	0.75537	0.75590	0.75685	0.75251

TABLE III. CROSS-VALIDATION RESULTS: MEAN VALUES AND STANDARD DEVIATIONS OF MODEL PERFORMANCE

Algorithm	Cross Validation Means	Cross Validation Errors
Naive Bayes	0.67358	0.00201
Logistic Regression	0.70942	0.00180
Stochastic Gradient Descent	0.71364	0.00372
Decision Tree	0.72226	**0.00156**
Random Forest	**0.77783**	0.00197
AdaBoost	0.73494	0.00344
Extra Trees	0.77258	0.00304
Gradient Boosting Machine	0.75486	0.00195

FCN model architecture features dense layers with different activation functions and dropout layers to prevent overfitting, created using an Adam optimizer with binary cross-entropy function and dropout layers as dropout layers to avoid overfitting. The LSTM architecture employed a recurrent network with LSTM cells. Data was reshaped according to input requirements for an LSTM model and dense layers were added similar to an FCN; dropout layers were also implemented to enhance generalization. Finally, this model was constructed and trained using an optimization algorithm, loss function, and FCN model as its training environment. Implementing the Bidirectional-LSTM Model An additional bidirectional layer was added to an LSTM architectural model for training of Bidirectional-LSTM model, enabling it to capture data from past and future timesteps while increasing understanding of temporal dependencies. Finally, an LSTM was used as the training medium.

V. RESULTS

A. Evaluation Metrics

- Accuracy, which stands as one of the most fundamental, and intuitive evaluation metrics as it measures the ratio of correctly predicted instances over the total number of evaluated instances as shown in formula (1). It signifies the overall correctness of a model's predictions. While accuracy serves as a valuable initial assessment, it may not be the sole determinant of a model's performance, particularly when dealing with imbalanced datasets where one class predominates over others [37].

$$\boldsymbol{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

- Error Rate, which quantifies the proportion of incorrectly predicted instances within the dataset and provides a clear picture of misclassifications which is the ratio of incorrectly predicted instances over the total number of evaluated instances as shown in formula (2) and is particularly relevant in scenarios where false positives or false negatives bear substantial consequences [37].

$$\boldsymbol{Error\ Rate} = \frac{FP+FN}{TP+TN+FP+FN} \quad (2)$$

- Precision [37], which accentuates the accuracy of positive predictions and quantifies the proportion of true positive predictions (correctly identified positive instances) relative to the total number of positive predictions (comprising true positives and false positives) as shown in formula (3).

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \quad (3)$$

- Recall (sensitivity or the true positive rate) assesses a model's capability to correctly identify all relevant instances from a dataset [37]. From the proportion of true positive predictions in relation to the total number of actual positive instances (encompassing true positives and false negatives) as shown in formula (4).

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} \quad (4)$$

TABLE IV. COMPARISON TABLE OF ML BASED MODELS

	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score	Specificity	Matthew Correlation Coefficient	Cohen Kappa	ROC Score	Loss Log
Naive Bayes	0.51866	0.5175	0.42993	0.9409	0.59017	0.26966	0.25797	0.1689	0.605	17.39107
Logistic Regression	0.63082	0.63075	0.00000	0.00000	0.00000	1.00000	0.00000	0.00000	0.5	13.3092
Stochastic Gradient Descent	0.5944	0.5908	0.46496	0.7177	0.56432	0.51652	0.2283	0.21049	0.617	14.74902
Decision Tree	0.68679	0.67717	0.57045	0.509	0.53799	0.77561	0.29225	0.29114	0.642	11.6358
Random Forest	0.69127	0.68942	0.6417	0.3598	0.46106	0.88239	0.28845	0.26644	0.621	11.19449
AdaBoost	0.77972	0.73435	0.66278	0.5712	0.61359	0.82987	0.41553	0.41288	0.701	9.57503
Extra Trees	0.70752	0.70098	0.65113	0.4098	0.50298	0.87147	0.32135	0.30464	0.641	10.7777
Gradient Boosting Machine	0.69441	0.69088	0.6805	0.307	0.42308	0.91563	0.28832	0.25117	0.611	11.14189

- F1-score, as shown in formula (5) represents the harmonic mean of precision and recall. This metric strikes a balance between precision and recall, offering a consolidated score that accounts for both false positives and false negatives [37].

$$\boldsymbol{F1-score} = \frac{\boldsymbol{2\times Precision\times Recall}}{\boldsymbol{Precision-Recall}} \quad (5)$$

B. Ensemble Learning on ML Results

At this research facility, an in-depth evaluation was conducted to select the ideal model. To do so, several classifiers were used such as Gaussian Naive Bayes (GNB), Logistic Regression (LR), Stochastic Gradient descent Decision Tree Random Forest AdaBoost Extra Trees to select an effective one.

As part of this evaluation of each classifier's performance, first conducted a stratified cross-validation with 10-folds to assess their performance and select the most accurate models for further study, cross-validation scores for each fold were then computed as illustrated in Table II which offers a side-by-side comparison of the variability and average performance of several machine learning methods. The best mean accuracy is demonstrated by Random Forest and Extra Trees, although numerous other algorithms, including Logistic Regression and Decision Tree, also perform consistently.

Calculation and summarizing the mean and standard deviation of cross-validation results for each model as provided in Table III.

After selecting classifiers, an in-depth analysis was conducted using grid search to fine-tune their hyperparameters. This involved optimizing each model's parameters using cross-validation with the GridSearchCV feature; hyperparameters were chosen carefully based on empirical evidence and prior knowledge for each classifier.

After optimizing hyperparameters, every classifier was trained using the entire training dataset. Standard scaling was applied both during training and testing to ensure unbiased evaluation, predictions were made using training data, predictions were made from both sources simultaneously while accuracy and computational time were recorded.

Reports were prepared on the optimal settings and scores for each classifier, along with grid scores from parameter tuning processes, to give an insight into performance variations between hyperparameter combinations. After using independent test data to assess the generalization abilities of classifiers, their generalization abilities were evaluated using various performance metrics. The results of the performance metrics of machine learning models as presented in Table IV reveal that Naive Bayes obtained moderate accuracy but demonstrated a trade-off between precision and recall.

Table V illustrated the additional results presents the performance metrics of ensemble learning techniques applied to the previously mentioned machine learning models. The two ensemble methods used are Simple Average and Voting (both Hard-Voting and Soft-Voting).

Simple Average, Hard-Voting Ensemble, and Soft-Voting Ensemble were among the ensemble approaches used to enhance overall prediction accuracy. The results of applying ensemble learning techniques to the previous machine learning models show improved performance compared to individual models. The Simple Average ensemble achieved good accuracy and precision, but its recall rate was relatively low. The Hard-Voting Ensemble achieved the highest train accuracy and a balanced performance between precision and recall. The

Soft-Voting Ensemble showed a good overall performance, with higher recall but slightly lower precision compared to the Hard-Voting Ensemble. Both ensemble methods demonstrated better performance metrics compared to the individual models, suggesting the effectiveness of combining multiple models in improving predictions.

By leveraging the strengths of individual models and combining their predictions, ensemble learning techniques have the potential to enhance the performance of machine learning models.

TABLE V. ENSEMBLE ON SIMPLE BASE MODELS

	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score	Specificity	Matthew Correlation Coefficient	Cohen Kappa	ROC Score
Simple Average	0.74303	0.69804	0.73163	0.28782	0.41312	0.93819	0.30955	0.25852	0.61301
Hard-Voting Ensemble	0.79982	0.76133	0.67724	0.67563	0.67643	0.81150	0.48737	0.48737	0.74357
Soft-Voting Ensemble	0.78274	0.75000	0.63556	0.75703	0.69100	0.74588	0.48896	0.48374	0.75145

In order to optimize these ensemble methods and determine their applicability to other datasets, additional analysis and experimentation are required. The Hard-Voting Ensemble outperformed the individual models in terms of accuracy, precision, memory, and discriminating ability. The Soft-Voting Ensemble exhibited good accuracy and recall as well, but with somewhat lower precision than the Hard-Voting Ensemble. Individual models were outperformed by both ensemble techniques, with the Soft-Voting Ensemble having the greatest ROC score, suggesting higher discriminating abilities.

C. Deep Learning Results

The provided results present the performance metrics of deep learning models applied to the Quora question pairs dataset. The evaluated models consist of Fully Connected Network (FCN), Long Short-Term Memory (LSTM), and Bidirectional LSTM (Bi-LSTM).

The results in Table VI show how three deep learning models, FCN, LSTM, and Bi-LSTM, performed in a classification test. The FCN model obtained 68.47% train accuracy and 68.54% test accuracy, suggesting consistent performance across training and testing periods.

Moreover, through the application of cross-validation, the initial test accuracy of the FCN model, which stood at 68.54%, was significantly improved to 0.81 % as demonstrated in Table VII.

1) Deep learning using cross validation: The results given in Table VIII showed the performance metrics of the FCN model when cross validation was done for a binary classification task. The evaluation of the model was carried out in two classes, which were labeled as 0 and 1. Precision, recall and F1 score were computed for each class.

Results from a binary classification model that was applied to a dataset with two separate classes labelled as 0 and 1. The outcomes are performance measures for a binary classification model for the classes labelled 0 and 1. The model obtains 86% accuracy, 84% recall, and an F1 score of 85% for class 0. It achieves 73% accuracy, 77% recall, and a 75% F1 score for class 1. Collectively, these measures show that the model performs very well at categorizing cases into class 0, while also achieving better for class 1 examples, even though with significantly lower precision.

VI. DISCUSSION

The outcomes and performance metrics of different machine learning and deep learning classifiers in the current study are analyzed to evaluate their effectiveness in predicting target classes.

The results of the performance metrics of machine learning models reveal that Naive Bayes obtained moderate accuracy but demonstrated a trade-off between precision and recall. These results are in consistence with [13] where they proposed to use of ANN's minimal cost architecture and the selection of highly dominating attributes from the questions make it an excellent model for detecting duplicate questions and subsequently finding high-quality replies to queries in Q&A forum. They obtained 0.673 % of accuracy from Multinomial Naive Bayes model.

Logistic Regression performed poorly; however Stochastic Gradient Descent displayed balanced precision and recall in agreement with [16] study article adopted by Sharma et al. who employed Natural Language Processing knowledge and different classification and boosting techniques to determine which is more useful. Then they examined the accuracy of various models to determine which method is best suited for the task. The same has been done with the aid of multiple graphs and tables to highlight the differences in the accuracy of various algorithms. By comparing the two questions, Sharma et al. were able to determine whether they were identical. They used a dataset provided by Quora to solve this hard challenge and trained multiple machine learning algorithms on four entries to determine whether two questions are identical or not. They used multiple techniques after cleaning and preparing the data as needed. First, they used logistic regression, which produced unsatisfactory results. Therefore, they attempted xG boosting with Count Vectorizer and TF-IDF Vectorizer, and they got an accuracy of more than 80%. With 125 trees, Random Forest produced the best results, yielding an accuracy of 83%, which is quite impressive. The performance of the Decision Tree and Random Forest models was comparable, with the latter obtaining slightly greater accuracy and specificity in consistence with results obtained by [13] and [14], which showed percent of specificity of 0.683% and 73.2% respectively. AdaBoost worked admirably, displaying strong accuracy, precision, recall, and discriminating abilities. Extra Trees and Gradient Boosting Machine performed rather well, with a trade-off between various models to determine which method is best suited for the task. The same has been done with the aid of multiple graphs and tables to highlight the differences in the accuracy of various algorithms. By comparing the two questions, Sharma et al. were able to determine whether they were identical. They used a dataset provided by Quora to solve this hard challenge and trained multiple machine learning algorithms on four entries to determine whether two questions are identical or not. They used multiple techniques after cleaning and preparing the data as needed. First, they used logistic regression, which produced unsatisfactory results. Therefore, they attempted XG boosting with Count Vectorizer and TF-IDF Vectorizer, and they got an accuracy of more than 80%. With 125 trees, Random Forest

produced the best results, yielding an accuracy of 83%, which is quite impressive. The performance of the Decision Tree and Random Forest models was comparable, with the latter obtaining slightly greater accuracy and specificity in consistence with results obtained by [13] and [14], which showed percent of specificity of 0.683% and 73.2% respectively. AdaBoost worked admirably, displaying strong accuracy, precision, recall, and discriminating abilities. Extra Trees and Gradient Boosting Machine performed rather well, with a trade-off between precision and recall. These results are in accordance with [14] and [15] where they investigated the task of Natural Language Understanding (NLU) by examining duplicate question identification in the Quora dataset. They conducted extensive investigation of the dataset and employed various machine-learning models, including linear and tree-based models. The researchers discovered that a simple Continuous Bag of Words neural network model outperformed more complex recurrent and attention-based models with accuracy of 83.4 %.

TABLE VI. COMPARISON TABLE OF DL BASED MODELS

	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score	Specificity	Matthew Correlation Coefficient	Cohen Kappa	ROC Score
FCN	0.68469	0.68537	0.79293	0.20021	0.31970	0.96939	0.28151	0.20071	0.58480
LSTM	0.70532	0.70494	0.78143	0.27896	0.41114	0.95432	0.33279	0.26915	0.61664
Bi-LSTM	0.65856	0.66013	0.83982	0.09832	0.17603	0.98902	0.20726	0.10691	0.54367

TABLE VII. FCN MODEL WITH CROSS VALIDATION

Evaluation metrics	FCN Model
Train Accuracy	0.95
Test Accuracy	0.81
Precision	0.81
Recall	0.81
F1 Score	0.81
Specificity	0.84
Matthew Correlation Coefficient	0.6
Cohen Kappa	0.6

TABLE VIII. BINARY CLASSIFICATION RESULTS FOR CLASS 0 AND 1

Class	Precision	Recall	F1 Score
0	0.86	0.84	0.85
1	0.73	0.77	0.75

The machine learning models' performance on the Quora question pairs dataset produced diverse results as the Logistic Regression model had a challenging time classifying positive instances, showing limited success along with the Naive Bayes and Stochastic Gradient Descent models. The Gradient Boosting Machine and AdaBoost models achieved the highest train accuracies, outperforming the Decision Tree, Random Forest, and Extra Trees models. These results are in agreement with authors of [17] research, where their goal was to find the best machine learning technique for removing all duplicate questions and increasing user satisfaction. Using a real-time dataset, this work trained and tested four machine learning models to recognize duplicate inquiries. The raw dataset was discovered to be 7GB in size. PL/SQL was used to pre-process data before it was stored in the database. PL/SQL loads the full dataset only once, and data is acquired directly from the database whenever a query is conducted, making this procedure quick and efficient. In one hour, the complete dataset was cleaned and pre-processed effectively. While existing solutions use python methods available in python libraries to pre-process massive datasets, it takes four times as long as PL/SQL. Four distinct machine learning models were applied, and their results were evaluated to determine which model performed the best. Following execution, the error parameters referred from the log loss function for the random model, logistic regression model, linear SVM, and XGBoost are 0.887, 0.521, 0.654, and 0.357, respectively. Because efficiency is inversely related to error function, it can be concluded that XGBoost is the optimal model, delivering highest accuracy in the shortest amount of time, which is supplemented by the unique pre-processing procedures performed using PL/SQL, hence improving overall response time.

The results of the Simple Average ensemble had a middling accuracy but a better specificity, suggesting its ability to recognize negative events in agreements with [22], which offered the use of base ensemble consists of two participants in soft voting mode, but multiple classifiers combined into an ensemble method to improve predictive performance. In addition, the experimental results demonstrated by [26] and [27] showed that ensemble classifiers perform significantly better than standalone classifications or majority voting ensembles for sentiment classification purposes. Furthermore, that research explored how feature representation and preprocessing affect sentiment classification performance in consistent with current data results.

The Deep Learning model has a comparatively high accuracy of 79.29%, indicating its ability to categorize positive events reliably. The reduced recall of 20.02%, on the other hand, indicates that the model struggled to catch many positive events. When compared to the FCN model, the LSTM model performed better in terms of accuracy, precision, and recall. It obtained 70.53% train accuracy and 70.49% test accuracy, with a precision of 78.14% and a recall of 27.89% in consistency with [18] study, which used a Siamese LSTM to assess the semantic similarity of two queries in order to improve prediction. The Siamese network is an architecture composed of parallel neural networks, namely LSTM units, for the parallel processing of two questions, with each question passing through an Embedding Layer, an LSTM unit, and then a dense layer. Following that, the outputs of two networks were integrated and compared, yielding a similarity score reflecting

how similar two queries are. The log loss metric was used as the major statistic in this study to evaluate alternative models. The main addition is that the Siamese network is utilized to process two questions in parallel and find vector representations for each. The vectors produced by this technology enable more effective similarity detection than existing models. The GloVe word embedding method was used to determine the semantic similarity of two queries. As the basis model, a random classifier was developed, then logistic regression, linear SVM, and the XGBoost model were utilized to reduce log loss. Finally, a Siamese LSTM was proposed, which significantly minimizes the loss. The XGBoost model accurately identified 69% of question pairings as duplicate, resulting in a recall rate of 0.69. The precision rate was 0.79, and the F1- score was 0.73. Finally, as compared to individual models, ensemble techniques performed better in the classification challenge. The Hard-Voting Ensemble and Soft-Voting Ensemble performed better in terms of accuracy, precision, memory, and discrimination, highlighting the value of mixing various models. These findings extend machine learning approaches for categorization problems by emphasizing the potential benefits of ensemble methods in improving prediction performance. Furthermore, current results are in agreement with [26] study which proposed a model that identifies duplicate question pairs by integrating three word embedding feature extraction techniques (Google News Vector, FastText Crawl, and FastText Crawl Subword), which results in significantly higher accuracy than these embeddings independently. Furthermore, this study developed a novel Siamese MaLSTM model that uses the Manhattan distance to determine semantic similarity among questions with 95% accuracy, far outperforming previous studies. Looking closely at the manhattan values, the manhattan score classifies the question pairings more accurately than any other embedding in a blend of different word embedding predictions; the duplicate question score is nearly one, while the non-duplicate pair values are nearly zero.

VII. Conclusion

In this study, a thorough investigation of deep learning (DL) and machine learning (ML) models using the dataset of Quora question pairings. To ensure a reliable analysis, the dataset was put through ten folds of cross-validation. A variety of machine learning (ML) models were trained, such as Naive Bayes, Logistic Regression, Stochastic Gradient Descent, Decision Tree, Random Forest, AdaBoost, Extra Trees, and Gradient Boosting Machine, and evaluated the performance of each model using a variety of evaluation criteria.

These findings showed that the ML models performed at various levels. While models like Decision Tree, Random Forest, AdaBoost, Extra Trees, and Gradient Boosting Machine performed better, models like Naive Bayes and Logistic Regression had little success. Following that, ensemble learning strategies like Simple Average, Hard Voting, and Soft Voting were used to improve the performance of the ML models. These ensemble approaches significantly increased F1 scores, accuracy, and precision, demonstrating their efficacy in combining predictions from many models.

In addition, this study investigated how well DL models like the Fully Connected Network (FCN), Long Short-Term Memory (LSTM), and Bidirectional LSTM (Bi-LSTM) performed. The DL models' performance was assessed using precision, recall, and F1 scores after they were trained on the identical dataset of Quora question pairs. Cross-validation was used to evaluate the FCN model with two classes, and the results showed precision, recall, and F1 scores of 0.81 for both classes. These results show that the FCN model has performed well overall.

This study discovered that the ensemble learning techniques applied to the ML models produced competitive results when comparing their performance to that of the DL models. The DL models, on the other hand, showed off their capacity to identify intricate patterns and connections in the dataset. With high precision, recall, and F1 scores for both classes, the FCN model showed promise.

In summary, the current study emphasizes the effectiveness of using ensemble learning techniques to improve the performance of machine learning models. Moreover, this study has observed that deep learning models, the FCN model demonstrate great potential in accurately categorizing pairs of questions. These discoveries contribute to the progress of natural language processing. Offer valuable insights for enhancing question pair classification tasks. Moving forward, it would be beneficial to concentrate on refining these models exploring different architectures and examining their applicability to diverse datasets and real-world scenarios.

VIII. Future Work

In light of current study's results in employing ensemble learning techniques to enhance deep learning models, several avenues for future work emerge, broaden this ensemble learning approach to include a broader range of deep learning architectures, improve computational efficiency, and improve interpretability. Will also examine applications in domains with limited labelled data, assess generalization skills further, and develop adaptive ensemble weight techniques for dynamic data distributions.

Acknowledgment

The authors would like to thank all those who made contributions towards this work.

References

[1] A. Mohammed and R. Kora, “A comprehensive review on ensemble deep learning: Opportunities and challenges,” Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 2, pp. 757-774, Feb. 2023, https://doi.org/10.1016/j.jksuci.2023.01.014

[2] Sagi, O., & Rokach, L, “Ensemble learning: A survey,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, Vol. 8, no. 4, pp.1249, 2018, https://doi.org/10.1002/widm.1249.

[3] Y. Freund and R. E. Schapire,“A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” Journal of Computer and System Sciences, vol. 55, pp. 119- 139 , 1997.

[4] D. H. Wolpert, “Stacked Generalization,” Neural Networks, Vol. 5, pp. 241-259, 1992 0893-6080/92.

[5] J. Abellán and C. J. Mantas, “Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring,” Expert Syst Appl, vol. 41, no. 8, pp. 3825–3830, Jun. 2014, doi: 10.1016/j.eswa.2013.12.003.

[6] A. A. Aburomman and M. Bin Ibne Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," Applied Soft Computing Journal, vol. 38, pp. 360–372, Jan. 2016, doi: 10.1016/j.asoc.2015.10.011.

[7] C. Catal, S. Tufekci, E. Pirmit, and G. Kocabag, "On the use of ensemble of classifiers for accelerometer-based activity recognition," Applied Soft Computing Journal, vol. 37, pp. 1018–1022, Dec. 2015, doi: 10.1016/j.asoc.2015.01.025.

[8] C. F. Tsai, Y. C. Lin, D. C. Yen, and Y. M. Chen, "Predicting stock returns by classifier ensembles," in Applied Soft Computing Journal, Mar. 2011, pp. 2452–2459. doi: 10.1016/j.asoc.2010.10.001.

[9] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015. doi: 10.1038/nature14539.

[10] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning."

[11] L. Deng and D. Yu, "Deep learning: Methods and applications," Foundations and Trends in Signal Processing, vol. 7, no. 3–4. Now Publishers Inc, pp. 197–387, 2013. doi: 10.1561/2000000039.

[12] T. Li, Y. Gao, K. Wang, S. Guo, H. Liu, and H. Kang, "Diagnostic assessment of deep learning algorithms for diabetic retinopathy screening," Inf Sci (N Y), vol. 501, pp. 511–522, Oct. 2019, doi: 10.1016/j.ins.2019.06.011.

[13] A. Dhakal, A. Poudel, S. Pandey, S. Gaire and H. P. Baral, "Exploring Deep Learning in Semantic Question Matching," IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), Kathmandu, Nepal, pp. 86-91, 2018, doi: 10.1109/CCCS.2018.8586832.

[14] L. Sharma, L. Graesser, N. Nangia, and U. Evci, "Natural Language Understanding with the Quora Question Pairs Dataset," ArXiv, vol abs/1907.01041, 2019, Corpus ID: 195776066, https://doi.org/10.48550/arXiv.1907.01041.

[15] A. Chandra, and R. Stefanus, "Experiments on Paraphrase Identification Using Quora Question Pairs Dataset,"Computation and Language-arXiv, Jun 2020, https://doi.org/10.48550/arXiv.2006.02648.

[16] A.Sharma, S. Jha, S. Arora, S. Garg, and Sandeep Tayal, "Twin Question Pair Classification," Smart and Sustainable Intelligent Systems, WILEY Online Library, Chapter 16, Book Editors: N. Gupta, P. Chatterjee, and T. Choudhury, March 2021, https://doi.org/10.1002/9781119752134.ch16.

[17] V.K.R. Anishaa, P. Sathvika, and S. Rawat, "Identifying Similar Question Pairs Using Machine Learning Techniques," Indian Journal of Science and Technology, vol. 14, no. 20, pp. 1635-1641, 2021, https://doi.org/ 10.17485/IJST/v14i20.312.

[18] M. Chandra, A. Rodrigues, and J. George, "An Enhanced Deep Learning Model for Duplicate Question Detection on Quora Question pairs using Siamese LSTM," IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, pp.1-5, 2022, DOI: 10.1109/ICDCECE53908.2022.9792906

[19] S.S.T. Gontumukkala, Y.S.V. Godavarthi, B.R.R.T. Gonugunta, D. Gupta, and S. Palaniswamy, "Quora Question Pairs Identification and Insincere Questions Classification," 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, pp. 1-6, 2022, doi: 10.1109/ICCCNT54827.2022.9984492.

[20] N. Sendi, N. Abchiche-Mimouni, and F. Zehraoui, "A new transparent ensemble method based on deep learning," in Procedia Computer Science, Elsevier B.V., 2019, pp. 271–280. doi: 10.1016/j.procs.2019.09.182.

[21] A. Mohammed and R. Kora, "An effective ensemble deep learning framework for text classification," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 10, pp. 8825–8837, Nov. 2022, doi: 10.1016/j.jksuci.2021.11.001.

[22] S. Karlos, G. Kostopoulos, and S. Kotsiantis, "A soft-voting ensemble based co-training scheme using static selection for binary classification problems," Algorithms, vol. 13, no. 1, Jan. 2020, doi: 10.3390/a13010026.

[23] C. A. Gonçalves, A. S. Vieira, C. T. Gonçalves, R. Camacho, E. L. Iglesias, and L. B. Diz, "A Novel Multi-View Ensemble Learning Architecture to Improve the Structured Text Classification," Information (Switzerland), vol. 13, no. 6, Jun. 2022, doi: 10.3390/info13060283.

[24] F. Haghighi and H. Omranpour, "Stacking ensemble model of deep learning and its application to Persian/Arabic handwritten digits recognition," Knowl Based Syst, vol. 220, May 2021, doi: 10.1016/j.knosys.2021.106940.

[25] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," Expert Syst Appl, vol. 77, pp. 236–246, Jul. 2017, doi: 10.1016/j.eswa.2017.02.002.

[26] A. Onan, S. Korukoğlu, and H. Bulut, "A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification," Expert Syst Appl, vol. 62, pp. 1–16, Nov. 2016, doi: 10.1016/j.eswa.2016.06.005.

[27] Ankit and N. Saleena, "An Ensemble Classification System for Twitter Sentiment Analysis," in Procedia Computer Science, Elsevier B.V., 2018, pp. 937–946. doi: 10.1016/j.procs.2018.05.109.

[28] Universitas Gadjah Mada, Institute of Electrical and Electronics Engineers. Indonesia Section., and Institute of Electrical and Electronics Engineers, Proceedings, 2019 5th International Conference on Science and Technology (ICST) : 30-31, Eastparc Hotel, Yogyakarta, Indonesia. July 2019.

[29] L. Wang, L. Zhang, and J. Jiang, "Detecting Duplicate Questions in Stack Overflow via Deep Learning Approaches," in Proceedings - Asia-Pacific Software Engineering Conference, APSEC, Dec. 2019, vol. 2019-December, pp. 506–513. doi: 10.1109/APSEC48747.2019.00074.

[30] E. Dadashov, Elkhan, S. Sakshuwong, and K. Yu, "Quora Question Duplication," 2017, https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs.

[31] X. Zhang, X. Sun, and H. Wang, "Duplicate Question Identification by Integrating FrameNet With Neural Networks," Thirty-Second AAAI Conference on Artificial IntelligenceProceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.

[32] H. Zhang, and L. Chen, "Duplicate Question Detection based on Neural Networks and Multi-head Attention," International Conference on Asian Language Processing (IALP), pp. 13-18, 2019.

[33] Z. Imtiaz, M. Umer, M. Ahmad, S. Ullah, G. S. Choi,and A. Mehmood, "Duplicate Questions Pair Detection Using Siamese MaLSTM," IEEE Access, vol. 8, pp. 21932-21942, 2020.

[34] A. Chunamari, M. Yashas, A. Basu, D. K. Anirudh, , and C.S. Soumya,, "Quora question pairs using XG boost," Emerging Research in Computing, Information, Communication and Applications, pp. 715–721, 2021.

[35] H. T. Le, D. T. Cao, T. H. Bui, L. T. Luong and H. Q. Nguyen, "Improve Quora Question Pair Dataset for Question Similarity Task," RIVF International Conference on Computing and Communication Technologies (RIVF), pp. 1-5, 2021.

[36] T.G. Dietterich, "Ensemble Methods in Machine Learning. In: Multiple Classifier Systems," MCS Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol. 1857, 2000, https://doi.org/10.1007/3-540-45014-9_1.

[37] M. K. Elhadad, K. F. Li , and F. Gebali, "Detecting Misleading Information on COVID-19," IEEE Access, vol. 8, pp. 165201-165215, Sep.2020, doi: 10.1109/ACCESS.2020.3022867.