Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

An Effective Ensemble of Deep and Machine Learning Methods for Classifying the Expertise Shape of CQA Users

S. Nemati *

Department of Computer Engineering, Shahrekord University, Shahrekord, Iran.

Article Info

Article History:
Received 22 January 2024
Reviewed 23 March 2024
Revised 09 April 2024
Accepted 15 April 2024

Keywords:
Shape of expertise
Deep learning
Machine learning
Ensemble method
Community question answering

*Corresponding Author's Email Address: *s.nemati@sku.ac.ir*

Abstract

Background and Objectives: Community question-answering (CQA) websites have become increasingly popular as platforms for individuals to seek and share knowledge. Identifying users with a special shape of expertise on CQA websites is a beneficial task for both companies and individuals. Specifically, finding those who have a general understanding of certain areas but lack expertise in other fields is crucial for companies who are planning internship programs. These users, called dash-shaped users, are willing to work for low wages and have the potential to quickly develop into skilled professionals, thus minimizing the risk of unsuccessful recruitment. Due to the vast number of users on CQA websites, they provide valuable resources for finding individuals with various levels of expertise. This study is the first of its kind to directly classify CQA users based solely on the textual content of their posts.

Methods: To achieve this objective, we propose an ensemble of advanced deep learning algorithms and traditional machine learning methods for the binary classification of CQA users into two categories: those with dash-shaped expertise and those without. In the proposed method, we used the stack generalization to fuse the results of the dep and machine learning methods. To evaluate the effectiveness of our approach, we conducted an extensive experiment on three large datasets focused on Android, C#, and Java topics extracted from the Stack Overflow website.

Results: The results on four datasets of the Stack Overflow, demonstrate that our ensemble method not only outperforms baseline methods including seven traditional machine learning and six deep models, but it achieves higher performance than state-of-the-art deep models by an average of 10% accuracy and F1-measure.

Conclusion: The proposed model showed promising results in confirming that by using only their textual content of questions, we can classify the users in CQA websites. Specifically, the results showed that using the contextual content of the questions, the proposed model can be used for detecting the dash-shaped users precisely. Moreover, the proposed model is not limited to detecting dash-shaped users. It can also classify other shapes of expertise, such as T- and C-shaped users, which are valuable for forming agile software teams. Additionally, our model can be used as a filter method for downstream applications, like intern recommendations.

Introduction

Community question-answering (CQA) websites have become increasingly popular as platforms for individuals to seek and share knowledge. Notable examples include Stack Overflow and Quora, which have experienced

significant success in the realm of CQA websites [1]. These platforms allow users to ask questions and offer answers to queries posed by other users. To improve the overall content quality, users can also comment and vote on both questions and answers. Additionally, these websites incorporate competitive elements such as reputation scores and badges to encourage active participation from users [1].

Recently, there has been a significant focus on conducting various research studies to identify proficient individuals within the domain of CQA platforms [2]. The prime aim of these investigations is to locate and rank users who possess the necessary knowledge and expertise to effectively address the questions being raised. Providing expert recommendations makes it possible to improve the quality of answers and reduce the waiting time for receiving responses. Additionally, in platforms such as Stack Overflow that involve job positions, the exploration of the most suitable individual for a specific job role serves as an additional motivation for expert-finding studies [3].

Apart from identifying experts, it is also crucial to understand the nature of their expertise. Researchers have proposed various expertise classifications based on the breadth and depth of an expert's knowledge across different fields [4]-[6]. These classifications include (see Fig. 1):

- I-shaped: Experts with advanced knowledge limited to a single field.
- T-shaped: Experts with advanced knowledge in one field and a broad understanding of other fields.
- C-shaped or M-shaped: Experts with advanced and broad knowledge spanning multiple fields.
- Dash-shaped or Hyphen-shaped: Individuals lacking advanced knowledge in any field, but have general knowledge in some fields.

These expertise shapes allow for a better understanding and characterization of experts within CQA platforms. Also, this understanding can help companies and organizations identify and hire trainees for their positions. In recent years, with the development of emerging technologies, companies have become more interested in hiring interns and using on-the-job training methods to prepare them for professional positions. The internship is defined in different ways in different sources, but the most general definition is the conditional employment of people on a part-time or full-time basis for a limited period with a focus on learning specific skills [7].

Every company has its own set of criteria when it comes to choosing an intern. However, in general, an ideal intern should possess the basic knowledge necessary for fulfilling the company's requirements and should be capable of handling the specific work areas (represented by the Dash-shaped users in Fig. 1).

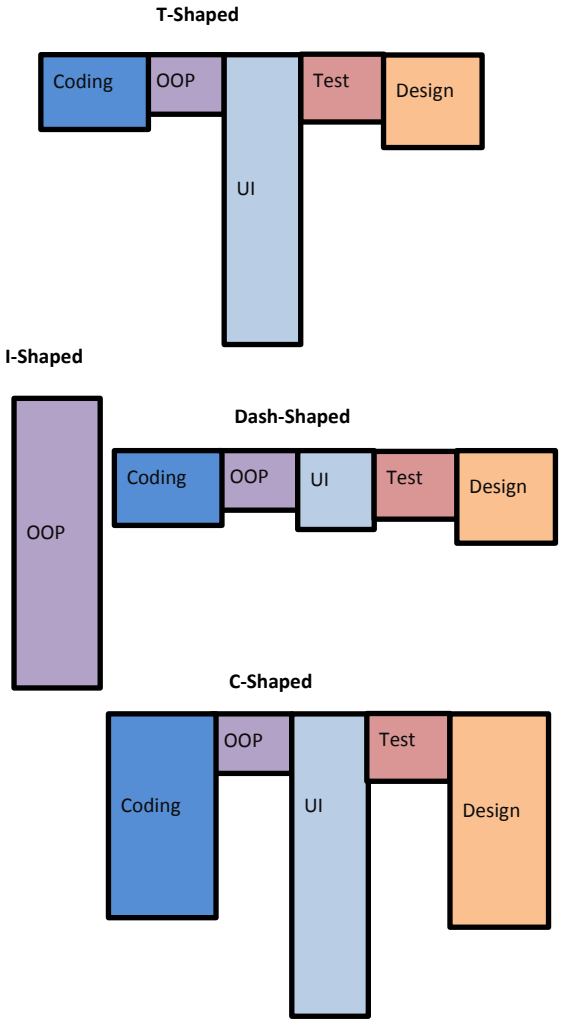


Fig. 1: The types of users in CQAs based on their breadth of knowledge.

Moreover, the internship period should not significantly burden the company financially, as there is a possibility that the intern may not end up being hired and may leave the company. Consequently, it is not advisable to select interns from individuals who are already experts or have extensive experience (represented by the I-, T-, or C-shaped users in Fig. 1), as these experienced individuals typically demand higher wages.

Previous research has primarily focused on identifying T-shaped users in CQA platforms. However, there is only one study that specifically addresses the issue of identifying suitable candidates for internship positions based on their expertise shape [1]. This study suggests that users with dash-shaped expertise have potential and are suitable choices for internship programs. However, the study does not clearly define the expertise shape and proposes statistical features to identify suitable users. Specifically, they propose two methods that utilize the

concept of entropy and the number of skills possessed by the candidates to identify dash-shaped users as suitable candidates for internships. However, their approach has two main limitations. Firstly, it only identifies users who have previously posted comments and does not apply to new users. Secondly, it ignores the most valuable aspect of a post, which is the textual content, when evaluating the expertise shape of users.

To address the issues identified in previous studies, we initially frame the problem of identifying dash-shaped users in CQA platforms as a binary classification problem. By focusing solely on the content of posts, we eliminate the need for user profiles and other statistical features associated with their questions and answers. This approach ensures the effectiveness of our proposed method even in the presence of the cold start problem. To tackle this problem, we introduce a novel ensemble method combining deep learning and traditional machine learning (ML) models. This fusion approach aims to enhance the accuracy and reliability of our solution. To demonstrate the effectiveness of our proposed model, we conduct experiments on three extensive datasets consisting of Stack Overflow questions. The primary contributions of our study can be summarized as follows:

- We introduced the problem of classifying users' shape of expertise only based on their comments.
- We proposed an ensemble method that utilizes the power of both traditional and deep learning models.
- We conducted extensive experiments and compared our method with seven machine learning and twelve deep models using three extensive datasets comprising Stack Overflow questions.

The remainder of the paper continues as follows. In the next section, a brief overview of related studies will be presented. Then, the proposed model will be described. Finally, experimental results, conclusions, and directions for future work will be discussed in the last section.

Literature Review

This section concisely summarizes relevant studies and is divided into three subsections as outlined below. Initially, we examine a selection of related studies regarding CQAs. Subsequently, we explore previous research that delves into the identification of expertise shapes. Finally, we briefly review deep learning models for the expert-finding problem.

A. Community Question Answering (CQA)

CQA platforms such as Stack Overflow are valuable repositories of knowledge. In recent times, there has been significant focus on the task of identifying experts within these platforms. The main concern lies in the low participation rate of users. To tackle this issue, various question routing methods have been devised to determine and suggest the most appropriate answer for new inquiries. A notable example is the work of Fu et al. [8], who introduced a recurrent memory reasoning network. This network utilizes the implicit relevance of the question and the history of the candidate user to locate experts. Another approach, proposed by Wang et al. [9], involves employing user profiles as input for a convolutional neural network. This network predicts the ideal candidate who can provide an answer to a new question. Furthermore, Kundu et al. [10] devised a method to estimate expertise scores by considering factors such as expert knowledge, reputation, and authority. Lastly, Sorkhani et al. [11] introduced a learning-to-rank framework for question routing. This framework incorporates a set of content-based and social-based features to rank and recommend suitable answers.

Researchers have also focused on studying the time-dependent and changing aspects of expertise. In a study by Neshati et al. [3], they introduced the concept of "future experts finding." This concept leverages existing evidence of expertise to predict the likelihood of users becoming experts in the future. The study explored four groups of features, including user behavior, emerging topics, topic similarity, and topic transitions. Another study by Zhang et al. [12] examined the temporal dynamics of answering behaviors in question routing. They developed a context-aware representation for each individual answering a question, taking into account the temporal context. Expertise was estimated by measuring the similarity between the representation of the answerer and the encoding of the question. In more recent research conducted by Liu et al. [13], a user-interest drift model was proposed. This model aimed to capture the dynamic nature of user interests over different periods.

B. Shape of Expertise

Over the past few years, there has been a growing focus on the idea of finding experts who possess specific forms and depths of expertise. This has become an important aspect of the overall problem of identifying and locating experts in various fields. In a study conducted by Rostami and Neshati in 2021, they introduced two retrieval models that are designed to effectively locate and rank individuals who possess dash-shaped expertise [1]. These individuals have an intermediate knowledge that matches the requirements of specific internship programs.

In 2018, Gharebagh and colleagues utilized a clustering method to analyze and extract various skill areas from the tags used in Stack Overflow [6]. They proposed two probabilistic models that are based on entropy calculations, which help in identifying T-shaped users within specific skill domains. In another study conducted by Rostami and Neshati in 2019, they developed two

retrieval models that focus on creating agile teams consisting of T-shaped experts [6]. These models aim to bring together individuals who possess a deep level of expertise in one area (the vertical part of the T) while also having a broader range of knowledge in other related fields (the horizontal part of the T).

In 2023, Rostami and Shakery introduced a deep learning algorithm that evaluates the likelihood of a candidate being a good fit for a particular role within an agile team [14]. Additionally, they implemented an integer linear programming model to identify the optimal members for an agile team with T-shaped experts, selecting them from a pool of highly qualified candidates.

Unlike previous studies in this specific domain, our research endeavors to classify individuals possessing specialized expertise with dash-shaped to fill an internship position. To the best of our knowledge, this particular aspect has not yet been explored or examined.

C. *Deep* Learning

In previous years, methods for identifying experts mainly relied on probabilistic language models [15]-[17], link analysis [18], [19], latent topic modeling [20]-[22], and other approaches. However, with the rise of deep learning, current expert-finding methods predominantly leverage deep learning techniques [12], [23].

In recent times, there has been considerable focus on the application of deep learning in the field of expert finding. Researchers such as Zhao et al. have developed frameworks that utilize random walk and LSTM neural networks to effectively rank candidates who can provide answers to specific questions [24]. Wang et al. have proposed a model based on CNN, which aims to identify experts on platforms like Stack Overflow [9]. Azzam et al. have generated a list of candidates ranked according to their ability to answer a given question by evaluating the cosine similarity between latent semantic vectors associated with each candidate and the question [25]. They have employed fully connected neural networks to learn these latent semantic vectors. Dehghan et al. have utilized an LSTM neural network that processes the breadth-first and depth-first traversal of candidates' expertise tree to find T-shaped experts who specialize in a specific skill area [26].

Li et al. have introduced a model called NeRank, which initially generates embedding representations of answerers and a given question using an LSTM-based model, and then uses a convolutional recommender system to compute the rank of answerers [27]. Tang et al. have proposed an attention-based factorization machine that generates a ranked list of experts in CQAs [28]. Lastly, Dehghan et al. have presented a CNN-based model that generates a ranked list of T-shaped experts who possess expertise in a particular skill area [26].

In a recent study conducted by Nikzad-Khasmakhi et al., they introduced BERTERS, a model that uses transformers and graph embedding techniques to identify potential expert candidates [23]. Similarly, our approach also involves deep learning, but with a different research objective. Unlike previous methods that focused on ranking experts or T-shaped experts with expertise in a particular query, we aim to use deep learning techniques to identify dash-shaped experts who are suitable for internship programs. Hence, the approaches discussed earlier are not applicable to address the specific problem we are trying to solve.

Problem and Data

A. *Problem Statement*

In the previous section, we discussed how most research has focused on finding people who have expertise in a particular field. However, our study is different because we are trying to identify users who have a variety of skills that would be useful in an internship. This is a binary classification challenge where we categorize the data based on the user's proficiency, which is represented by (1) in our investigation.

$$class(u_i) = \begin{cases} 0 & shape(u_i) \in \{I, T, C\} \\ 1 & otherwise \end{cases} \quad (1)$$

where, u_i is the *i*-th user, $shape(u_i)$ is the expertise shape based on the category shown in Fig. 1, and $class$ represents the user class label. It should be noted that the initial labeling of the dataset was done manually by Gharebagh et al. [6] and used in [1]. In the current research, the collection of all the texts related to the answers of the users is in the form of:

$$D = \bigcup_{i \in U} D_i \quad (2)$$

where, $D_i = \{d_{i1}, d_{i2}, \dots, d_{in}\}$ represents the set of answers' texts of the *i*-th user. Each d_i can be shown as:

$$d_i = \bigcup_{j=1}^{k} d_{sa_j,i} \quad (3)$$

to where $sa_j \in S = \{sa_1, sa_2, \dots, sa_m\}$ is a skill area. In order to identify the dash-shaped users the following probability is estimated [1]:

$$P(H = 1, i) \quad (4)$$

This shows the probability of user i being dash-shaped and can be estimated as:

$$P(H = 1, i) \propto \frac{Entropy(i)}{log\,|D_i + 1|} \quad (5)$$

where $Entropy(i)$ can be determined only based on the documents written by the *i-th* user as follows.

$$Entropy(i) = -\sum_{j=1}^{k} P_{sa_j,i}\, log\, P_{sa_j,i} \quad (6)$$

where k denotes the number of skill areas as before, and $P_{sa_j,i}$ is calculated using the following equation:

$$P_{sa_j,i} = \frac{|D_{sa_j,i}|}{|D_i|} \quad (7)$$

As discussed in [1], in addition to dash-shaped users, C-shaped users also have high values for the probability obtained using (6) due to their high diversity of documents while dash-shape users have lower values for the denominator of (5). Consequently, this distinction in probability values can serve as a means to differentiate dash-shaped users from those with T-, I-, and C-shaped behaviors.

B. Dataset

This research employs three datasets originally introduced by Gharebagh et al. [6]. These datasets are derived from queries made on the Stack Overflow website between August 2008 and March 2015, focusing on three specific categories: C#, Java, and Android. Each category comprises questions with relevant tags and their corresponding answers. Each dataset has identified a set of key skill areas based on the 200 most frequent tags within that particular set. Additionally, the level of knowledge possessed by users in each skill area has been established by considering the number of accepted answers provided by the users in those areas. The users within each dataset are then categorized based on their knowledge level in each skill area.

We had to obtain the text of users' questions and answers from the Stack Exchange Data Explorer database since the original datasets didn't have it. We used the user ID, question ID, and answer ID to get this information. Once we had the text, we added it to the datasets. To assess the impact of different category types on the models' performance, we combined the three datasets into one, which we call the All dataset. We also made sure that the datasets were balanced by adjusting the number of records based on the minority class. Specifically, we used random under-sampling to remove samples from the training dataset that belong to the majority class to balance the class distribution. This was done to ensure fair comparison results. You can find more details about the datasets in Table 1.

Table 1: Specification of datasets used in the current study. Max length and Avg. length represent the maximum and average length of texts in number of words

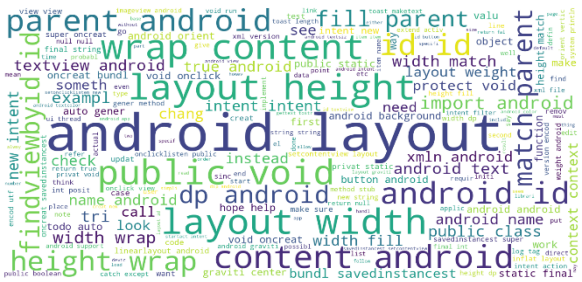
Dataset	# of Rec	Max length	Avg. length	# of unique words
Android	22902	247	95.959	70918
C#	31522	189	102.732	116203
Java	31302	214	100.689	102076
All	85726	278	100.138	237092

To show an overall view of the content of datasets based on their words, word clouds of datasets are shown in Fig. 2.

Proposed Model

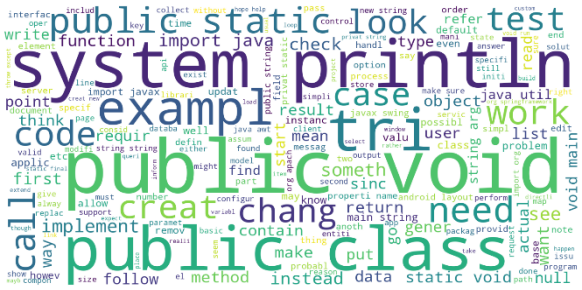
As part of our investigation, we have used a combination of advanced techniques, including deep learning, pre-training, and machine learning. These techniques are classified into three groups of diverse classifiers, aimed at producing more precise results. Our model is based on the concept of meta-learning, where a set of initial classifiers are trained using the given training data to generate predictions. These predictions are then used to train a meta-classifier, which makes the final prediction. Fig. 3 contains a diagram that illustrates the overall structure of our proposed model.

(a)



(b)

(c)



(d)

Fig. 2: Word clouds of the positive class for the (a) All, (b) Android, (c) C#, and (d) Java datasets.

In our proposed approach, we have utilized three different types of classifiers. These include classical machine learning classifiers, popular deep learning models, and a pre-trained transformer-based Bert model. The subsequent sections provide a brief overview of these models.

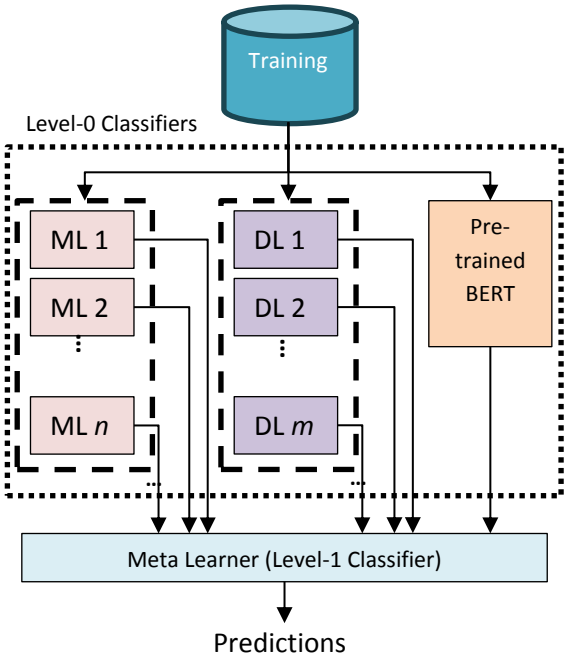


Fig. 3: The overall structure of the proposed ensemble model.

A. Classical Machine Learning Models

In the classical machine learning part of the proposed model, we utilized a total of seven techniques (i.e., n = 7 in Fig. 3): random forest (RF), support vector machine (SVM), decision tree (DT), logistic regression (LR), multi-layer perceptron (MLP), and two boosting classifiers including Adaboost (Ada), and XGBoost (XGB). Presented below is a brief outline of each of these methodologies.

- RF is a powerful methodology that synergizes the strengths of numerous decision trees, each trained on different subsets of data. This approach significantly boosts the precision and reliability of a specific dataset by leveraging the collective consensus derived from these trees. Instead of relying on a single tree's prediction, the RF algorithm calculates the average result generated by an ensemble of trees, thereby ensuring enhanced accuracy [29].
- SVM has gained extensive usage across various domains over a significant period for its ability to forecast outcomes and tackle classification and regression challenges. This technique effectively ascertains the optimal hyperplane to divide data into two distinct classes [29].
- DT is a type of supervised learning classifier that operates without any predetermined parameters. It comprises internal nodes responsible for making decisions, while the outcome is depicted by the leaf nodes [29]. In the current study, we used the CART (Classification And Regression Tree) variants of a decision tree that uses a greedy approach to split the data at each node.
- Logistic regression is a popular algorithm utilized in supervised learning. It aims to estimate the probability and forecast the result of a categorical dependent variable by establishing a connection between independent variables and the dependent variable [30].
- MLP is a type of feedforward neural network, consisting of three layers: input, output, and hidden. It uses a linear activation function [30].
- AdaBoost, short for adaptive boosting, is a boosting technique derived from the boosting algorithm. Its objective is to merge several weak classifiers into a powerful classifier [30].
- XGBoost is a powerful approach to gradient boosting, which encompasses a range of machine learning algorithms. It combines several weak learning models, particularly decision trees, to create a high-performing and reliable predictive model [30].

A review of deep and classical ML methods for classification tasks was presented in [31].

B. Deep Learning Models

In the deep learning part of the proposed model, we exploited five methods (i.e., m = 5 in Fig. 3): dense, GRU, CNN, BiLSTM, and CNN-LSTM models. The details of these models are as follows.

- Dense: This type of deep model is commonly used in various deep learning tasks, such as image classification, natural language processing, and speech recognition [32]. In the current study, we implemented a dense model shown in Fig. 4. It contains five fully connected dense layers with sizes shown in the figure.
- CNN: This type of deep model is primarily used for image processing tasks, but it can also be applied to text classification tasks [33]. To this aim, CNNs can be used to extract meaningful features from textual data. In the current research, we used the CNN model shown in Fig. 5. Here, the model can learn to automatically extract relevant features from the text data and capture important patterns using the convolutional layer followed by Maxpooling which is used for dimensionality reduction.
- GRU: Gated Recurrent Unit (GRU) deep models are suitable for text classification due to their ability to capture sequential dependencies, handle variable-length inputs, and efficiently process text data [34]. In the current study, we used the GRU model shown

in Fig. 6. It can effectively capture the contextual information and dependencies between words in a sentence using three GRU layers.

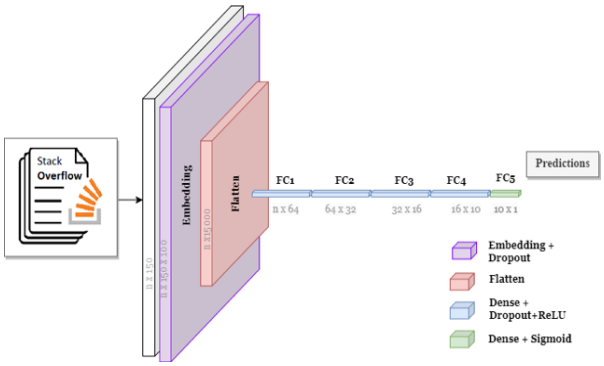


Fig. 4: The overall structure of the dense model.

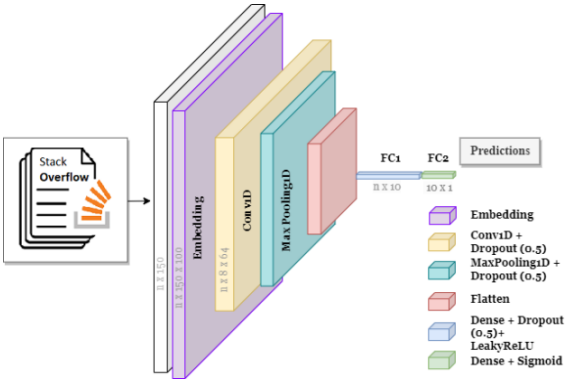


Fig. 5: The overall structure of the CNN model.

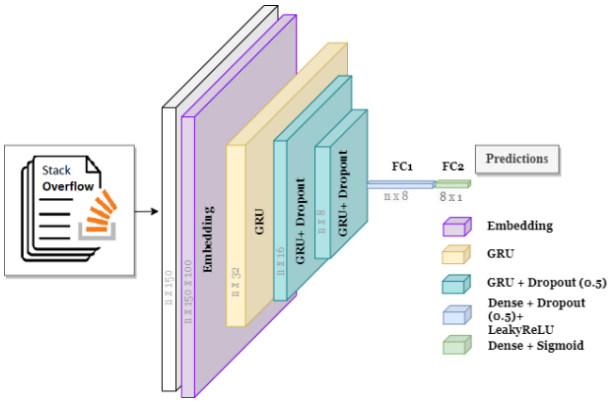


Fig. 6: The overall structure of the GRU model.

- CNN-LSTM: The combination of CNNs and Long Short-Term Memory (LSTM) networks is a popular approach for text classification tasks. This combination allows the model to capture both local and global dependencies in the text data [35]. In the current study, we used CNNs for feature extraction and LSTMs for sequence modeling as shown in Fig. 7. This allows for a more comprehensive understanding of the text data and can improve the accuracy of text classification tasks.
- BiLSTM: Bidirectional LSTM is a type of Recurrent Neural Network (RNN) that is commonly used for text classification tasks [36]. It is particularly effective in capturing contextual information from both past and future words in a sequence. In the current study, we used BiLSTM models as shown in Fig. 8.
-

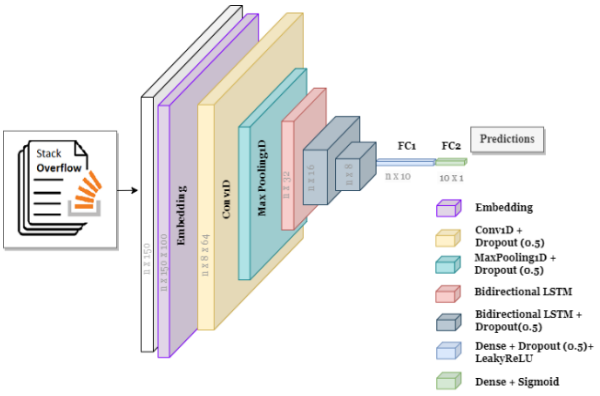


Fig. 7: The overall structure of the CNN-LSTM model.

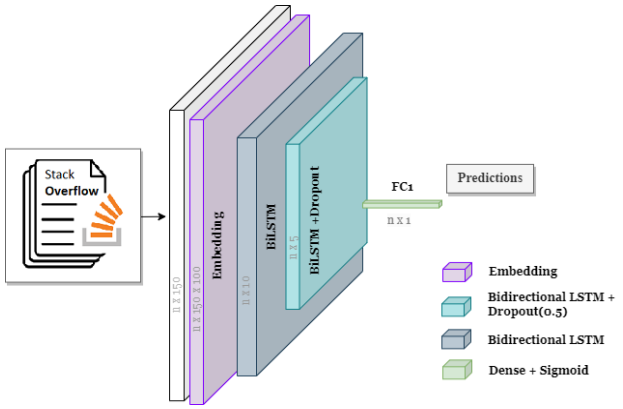


Fig. 8: The overall structure of the BiLSTM model.

C. *Bert Model*

The third level-0 classifier used in the proposed method is the pre-trained Bert classifier which is a transformer-based multi-layered encoder [37]. It uses an attention mechanism to learn the relationship between all words in a sentence. Specifically, it contains three embedding modules and 12 transformer layers each containing a dense layer and an attention layer. In the current study, we adopted the Huggingface[1] implementation of the Bert model. Bert has been previously used for text classification and a comparison of Bert and ML methods was provided in [38].

[1]https://huggingface.co/transformers/v2.10.0/model_doc/bert.html

D. Ensemble of Models

The main rationale behind employing the aforementioned three learning models in the present study lies in their utilization of different text features and their distinct mechanisms for generating predictions. This diversity holds great importance for meta-learning and stacking models [39]. In the field of data analysis, stacking is a useful technique that leverages the diverse predictions generated by base models to capture various aspects of the data, ultimately improving the accuracy of predictions. Each base model has its strengths and weaknesses, but by combining their predictions, the ensemble model can benefit from the collective expertise of these models [40]. Additionally, stacking helps to reduce bias and variance, leading to better accuracy by consolidating predictions from multiple models [41]. In addition, it assists in capturing detailed connections and patterns in the data that individual models might fail to notice. In our proposed model, we implemented a variant of the stack generalization technique, as depicted in Algorithm 1.

Algorithm 1: Pseudo-code for the stack generalization algorithm adopted from [15].

Data: Training dataset
$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$
Level-0 learning algorithms $\mathcal{L}_1, \cdots, \mathcal{L}_M$
Level-1 learning algorithms $\mathcal{L}$
Test dataset $\mathcal{X}' = \{x'_1, x'_2, \cdots, x'_T\}$
Result: Prediction vector $\mathcal{Y}' = \{y'_1, y'_2, \cdots, y'_T\}$

begin
  Randomly split $\mathcal{D}$ into I almost equal folds: $\mathcal{D}_1, \cdots, \mathcal{D}_I$
  $\mathcal{D}' = \varnothing$
  for $i = 1, \cdots, I$ **do**
    $\mathcal{D}^{-i} = \mathcal{D} - \mathcal{D}_i$
    $h = \varnothing$
    for $m = 1, \cdots, M$ **do**
      $h_m = \mathcal{L}_m(\mathcal{D}^{-i})$
    end
    $z = \varnothing$
    for $k = 1, \cdots, |\mathcal{D}^i|$ **do**
      $d = \varnothing$
      for $m = 1, \cdots, M$ **do**
        $d_m = h_m(\mathcal{D}^i_k[x])$
      end
      $z_k = (d, \mathcal{D}^i_k[y])$
    end
    $\mathcal{D}' = \mathcal{D}' \cup z$
  end
  $h' = \mathcal{L}(\mathcal{D}')$
  $\mathcal{Y}' = \varnothing$
  for $k = 1, \cdots, T$ **do**
    $z = \varnothing$
    for $m = 1, \cdots, M$ **do**
      $z_m = \mathcal{L}_m(x'_k)$
    end
    $\mathcal{Y}'_k = h'(z)$
  end
  return $\mathcal{Y}'$
end

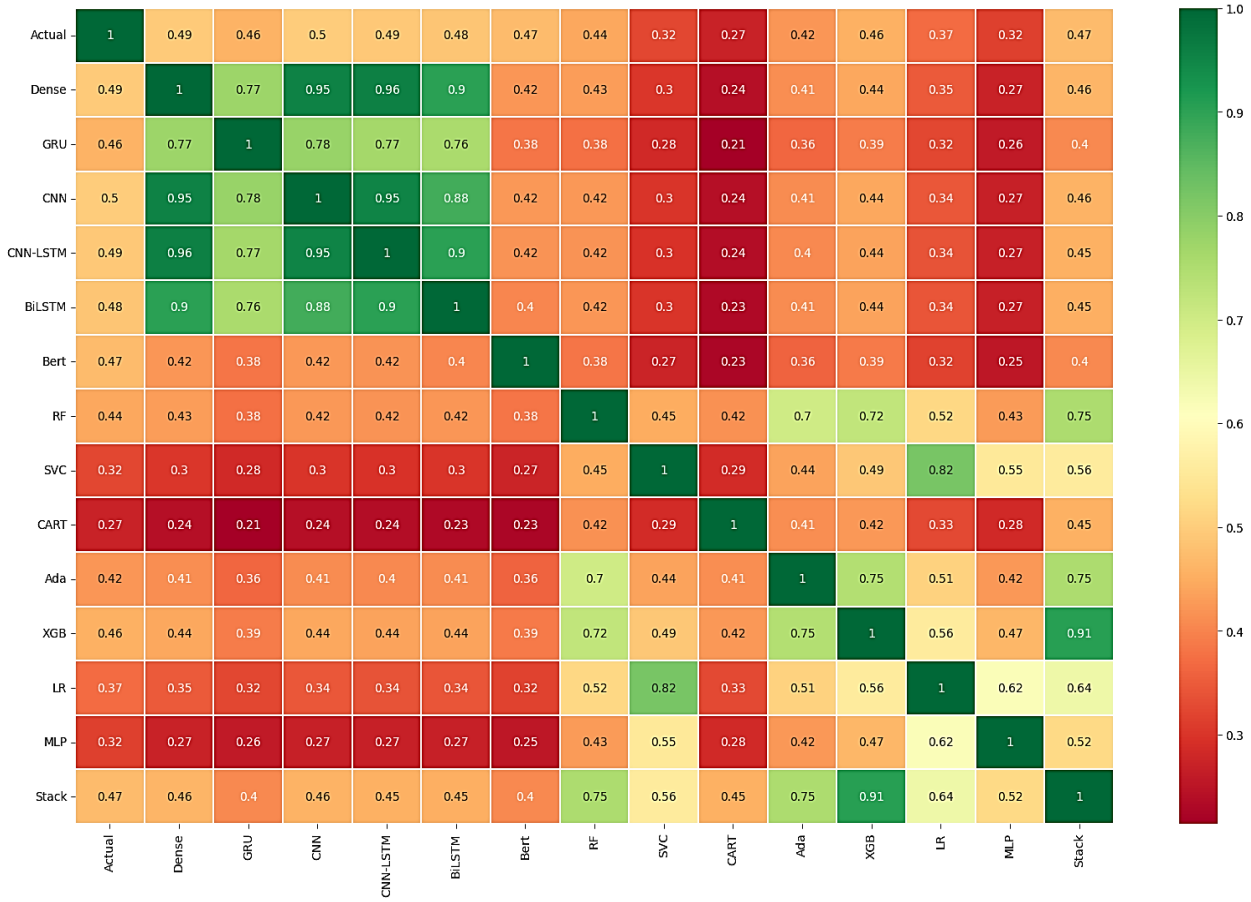


Fig. 9: Correlation between classical ML, BERT, and other deep learning models.

To showcase the effectiveness of the three different models used in the proposed method, we created a correlation chart for these methods, as shown in Fig. 9. As evident in the diagram, there is a strong correlation observed among the deep models, while the correlation between the deep and ML methods, as well as the Bert model, is relatively weaker. This suggests that combining the models in an ensemble can produce more precise outcomes as compared to using each learning method individually.

Experimental Settings

A. Compared Baselines

Our study aims to classify users' expertise based on their shape, which we frame as a binary text classification task. This differs from previous research that focused on expert identification and intern retrieval, such as the works of Gharebagh et al. [6] and Rostami and Neshati [1]. Hence, we cannot directly compare our findings with theirs due to the dissimilarity between the problems. To demonstrate the effectiveness of our proposed approach, we utilized seven deep-learning techniques that are commonly used for binary text classification:

- CRNN [42]: In this approach, every sentence is regarded as a region, and a regional CNN is utilized on the input word vectors. Subsequently, max pooling is employed to decrease the dimensionality of the local features. Finally, an LSTM layer is utilized to capture long dependencies, and a linear decoder is used to make predictions.
- IWV [43]: This model comprises three convolution layers, a max pooling layer, and a fully connected layer stacked sequentially for sentiment polarity classification.
- SS-BED [44]: This model utilizes two parallel LSTM layers on two distinct word embedding matrices to acquire knowledge about semantic and sentiment feature representations. The results obtained from the LSTM layers are then inputted into a fully connected network with one hidden layer to make the predictions.
- HAN [45]: This model comprises four essential components: a word sequence encoder, which is a bidirectional GRU, a word-level attention layer that calculates weighted sentence vectors, a sentence encoder, which is another bidirectional GRU, and a sentence-level attention layer that rewards sentences for making accurate classifications.
- ARC [46]: In this model, a one-layer bidirectional GRU is applied to the word vectors, and the outcomes are fed into an attention layer. The output of the attention mechanism is then passed through a CNN layer, followed by a max-pooling layer and a fully connected layer.
- AC-BiLSTM [47]: This model has a one-dimensional CNN layer consisting of CNNs of different filter sizes. This layer is employed for localized feature extraction. The output of the CNN layer is then fed into a bidirectional LSTM layer, followed by an attention mechanism. The output layer of this model consists of a dropout layer and a softmax layer.
- ABCDM [39]: This method utilizes a unique combination of two bidirectional LSTM and GRU layers to effectively capture contextual information from preceding and forthcoming contexts. This allows ABCDM to consider the sequential progression of information in both forward and backward directions. Additionally, ABCDM seamlessly integrates an attention mechanism within the bidirectional layers, allowing it to selectively emphasize specific words based on their varying levels of significance. Furthermore, ABCDM incorporates convolution and pooling mechanisms to reduce the complexity of features and extract localized features more efficiently.

B. Environment Setting

In our comparative analysis, we investigated the utility of the proposed model against the baseline learning models used in the proposed ensemble model as well as against seven state-of-the-art deep models described in the previous section. All the implementations were carried out using Tensorflow 2.14.0, Numpy, Sklearn, and Pandas in Python3 (version: 3.12), and Transformer (version: 4.36.2). All the models were implemented in the Google Colab environment with an Intel Xeon CPU accompanied by a 13 GB RAM, a Tesla K80 accelerator, and 12 GB GDDR5 VRAM.

C. Evaluation Criteria

To evaluate the effectiveness of models, we employed *Precision* (π), *recall* (ρ), *accuracy*, *F1*, and Area Under Curve (AUC) evaluation criteria in the experiments [39].

$$F1 = \frac{2 \times \pi \times \rho}{(\pi + \rho)} \tag{8}$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

$$\pi = \frac{TP}{TP + FP} \tag{10}$$

$$\rho = \frac{TP}{TP + FN} \tag{11}$$

$$AUC = \frac{\sum Rank(+) - \left(|+| \times \frac{|+|+1}{2}\right)}{|+|+|-|} \tag{12}$$

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively and $\sum Rank(+)$ is the sum of the ranks of all positively classified samples, |+| and |-| are the number of positive and negative samples in the dataset, respectively.

Results

A. Preliminary Results

In our first round of experiments, we compared seven different machine learning methods and their ensemble (named as SG-ML) using the stack generalization method outlined in Algorithm 1 in Table 2.

Table 2: Comparison of results obtained using ML methods and their ensemble (SG-ML). Bold values indicate the best-performed models

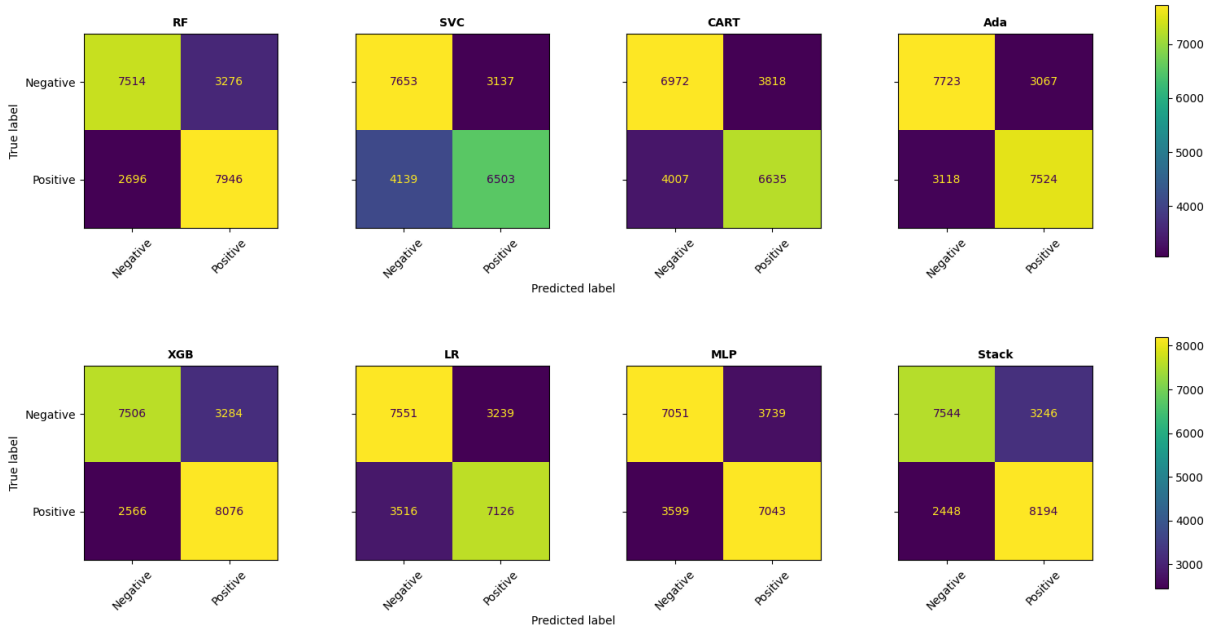
		Acc	π	ρ	F1
Android	RF	68.91	66.91	74.68	70.58
	SVM	62.54	63.97	57.19	60.39
	CART	61.49	61.50	61.18	61.34
	Ada	68.08	68.20	67.58	67.88
	XGB	68.58	67.81	70.58	69.17
	LR	64.77	65.19	63.20	64.18
	MLP	65.68	64.23	70.58	67.26
	SG-ML	**69.52**	**68.38**	**72.47**	**70.37**
C#	RF	73.28	70.74	78.43	74.39
	SVM	68.53	69.29	65.38	67.27
	CART	65.60	65.09	65.71	65.40
	Ada	73.04	72.39	73.56	72.97
	XGB	74.83	72.11	80.10	75.89
	LR	70.05	70.01	69.04	69.52
	MLP	66.69	66.01	67.38	66.68
	SG-ML	**75.32**	**72.84**	**79.92**	**76.21**
Java	RF	72.67	70.70	76.42	73.45
	SVM	65.55	66.52	61.16	63.72
	CART	64.82	64.41	64.59	64.50
	Ada	72.28	71.99	72.00	71.99
	XGB	72.94	71.11	76.29	73.61
	LR	67.29	67.40	65.63	66.50
	MLP	73.36	71.59	76.52	73.97
	SG-ML	**73.36**	**71.59**	**76.52**	**73.97**
All	RF	72.14	70.81	74.67	72.69
	SVM	66.05	67.46	61.11	64.13
	CART	63.49	63.47	62.35	62.91
	Ada	71.14	71.04	70.70	70.87
	XGB	72.70	71.09	75.89	73.41
	LR	68.48	68.75	66.96	67.84
	MLP	65.76	65.32	66.18	65.75
	SG-ML	**73.43**	**71.63**	**77.00**	**74.21**

Our analysis revealed that the RF, Ada, and XGB classifiers scored higher in terms of accuracy and F1 scores than the other methods. Additionally, the SG-ML model outperformed all level-0 models across all four datasets. We also compared the deep models and their ensemble using the same stack generalization method shown in Algorithm 1 in Table 3. The results indicate that the CNN model and the ensemble model achieved higher scores overall, but the differences between the individual model performances and their ensemble were less pronounced compared to the ML algorithms. This suggests that the variance of the deep models is lower than that of ML models.

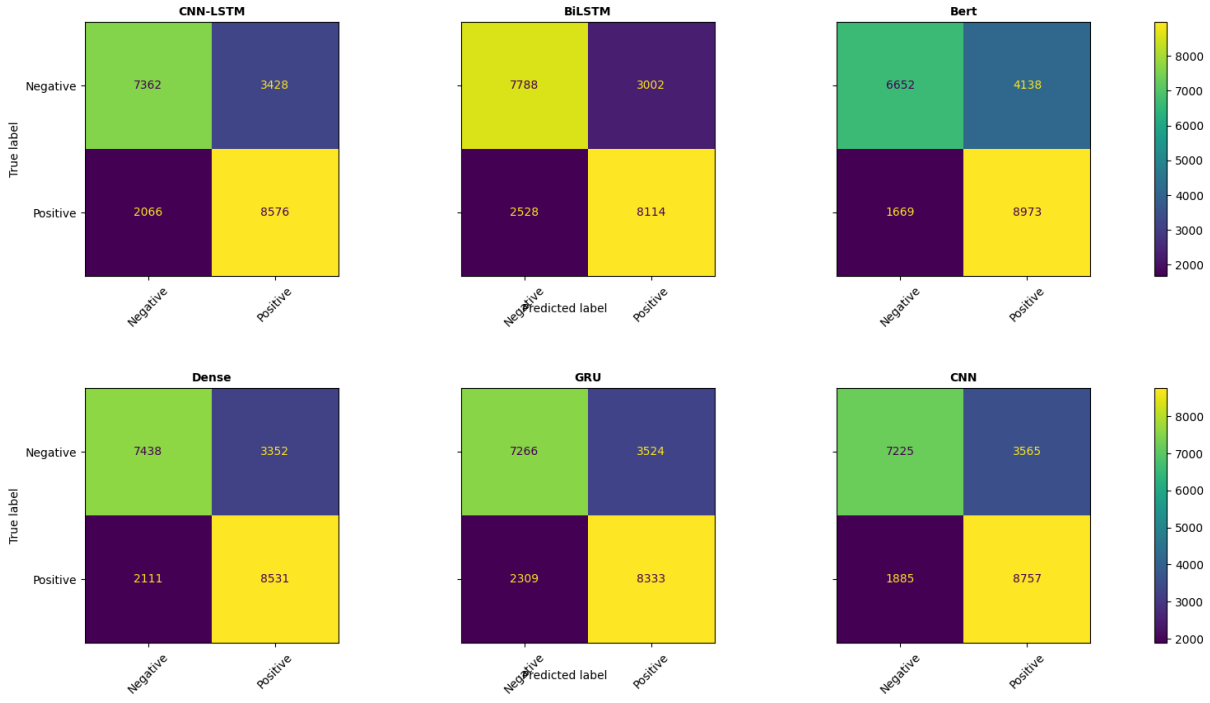
Table 3: Comparison of results obtained using deep methods and their ensemble (SG-Deep). Bold values indicate the best-performed models

		Acc	π	ρ	F1
Android	Dense	71.90	71.87	71.84	71.86
	GRU	71.92	71.07	**73.80**	72.41
	BiLSTM	**72.09**	70.67	75.41	**72.96**
	CNN	71.83	70.40	75.20	72.72
	CNN-LSTM	71.50	**73.53**	67.05	70.14
	SG-Deep	72.02	72.10	72.00	72.02
C#	Dense	76.58	73.71	81.84	77.56
	GRU	76.48	72.10	85.56	78.25
	BiLSTM	75.93	72.03	83.94	77.53
	CNN	**76.74**	72.64	**85.00**	**78.34**
	CNN-LSTM	76.08	**73.89**	79.89	76.77
	SG-Deep	76.64	77.42	76.64	76.52
Java	Dense	75.02	71.77	**81.61**	76.37
	GRU	74.83	72.13	80.06	75.89
	BiLSTM	73.15	69.20	82.44	75.24
	CNN	74.81	71.17	82.52	**76.43**
	CNN-LSTM	74.39	73.77	74.87	74.31
	SG-Deep	**75.13**	**75.55**	75.13	75.07
All	Dense	74.51	71.79	80.16	75.75
	GRU	72.78	70.28	78.30	74.07
	BiLSTM	74.20	72.99	76.25	74.58
	CNN	**74.57**	71.07	**82.29**	**76.27**
	CNN-LSTM	74.37	71.44	80.59	75.74
	SG-Deep	74.54	**74.99**	74.54	74.46

To provide more detailed information about the performance of different models on positive and negative classes, we have presented the confusion matrix of the All dataset in Fig. 10. We obtained similar results for the other three datasets, but we could not show them due to space limitations. As indicated in the figure, the XGB method had the best true positive result among the ML methods, while the Ada method had the best true negative. The Ada and XGB methods had the best false positive and false negative results, respectively, highlighting their effectiveness for classification tasks. Among deep models, the CNN model provided the best true positives, while the BiLSTM model provided the best true negatives. The BiLSTM and Bert models had the best false positives and false negatives, respectively.



(a)

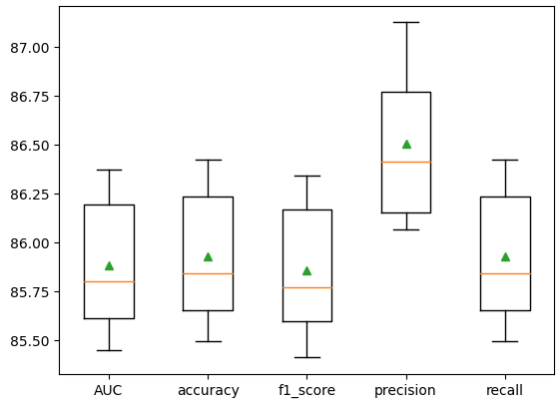


(b)

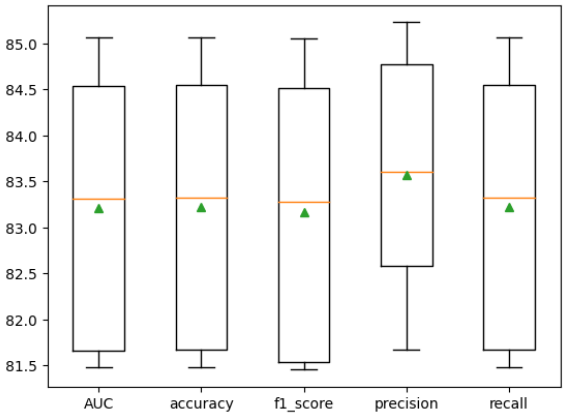
Fig. 10: Confusion matrix for (a) classical ML, and (b) Bert and other deep learning models on the All dataset.

B. *Main Results*

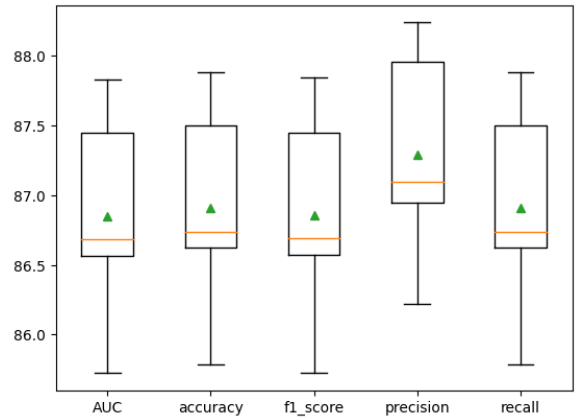
In the second part of our experiments, we presented the performance of our proposed model through 5-fold cross-validation, as displayed in Fig. 11 as box plots. The results indicate that the model's performance on the Android dataset is comparatively lower than that of other datasets, possibly due to the nature of the texts in this dataset or its fewer records when compared to other datasets (refer to Table 1 for more information).



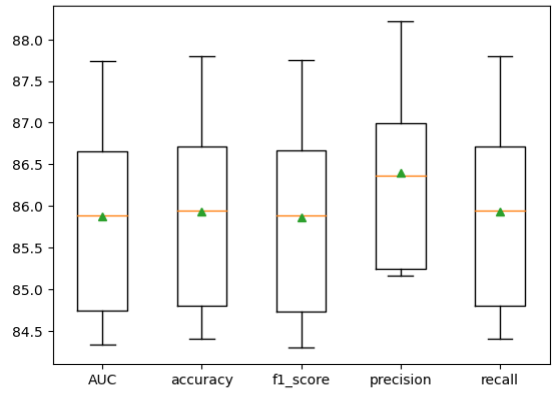
(a)



(b)



(c)



(d)

Fig. 11: Comparison of the results obtained using the proposed method with 5-fold cross-validation on the (a) All, (b) Android, (c) C#, and (d) Java datasets.

Table 4: Comparison of results obtained using the proposed model and state-of-the-art binary text classification models. Bold values indicate the best-performed models

		Acc	π	ρ	F1
Android	SS-BED	74.36	71.24	81.12	75.86
	ACBiLSM	71.99	67.15	85.32	75.15
	IWV	67.24	67.18	66.51	66.84
	HAN	64.30	64.03	64.12	64.08
	CRNN	74.17	70.15	83.53	76.25
	ARC	74.53	71.58	80.78	75.90
	ABCDM	74.36	71.24	81.12	75.86
	Proposed	**85.93**	**86.51**	**85.93**	**85.86**
C#	SS-BED	75.32	71.03	84.64	77.24
	ACBiLSM	72.00	70.48	74.66	72.51
	IWV	71.40	70.06	73.69	71.83
	HAN	66.78	65.25	70.27	67.67
	CRNN	76.06	70.87	87.61	78.36
	ARC	76.23	71.34	86.84	78.33
	ABCDM	66.87	64.48	73.53	68.71
	Proposed	**86.91**	**87.29**	**86.91**	**86.85**
Java	SS-BED	73.35	70.77	78.59	74.47
	ACBiLSM	70.19	68.41	73.84	71.02
	IWV	68.48	66.78	72.21	69.39
	HAN	63.61	62.11	67.82	64.84
	CRNN	73.77	69.23	84.56	76.13
	ARC	74.30	69.69	85.05	76.61
	ABCDM	65.14	64.81	64.64	64.73
	Proposed	**85.93**	**86.40**	**85.93**	**85.87**
All	SS-BED	75.32	71.03	84.64	77.24
	ACBiLSM	72.00	70.48	74.66	72.51
	IWV	71.40	70.06	73.69	71.83
	HAN	66.78	65.25	70.27	67.67
	CRNN	76.06	70.87	87.61	78.36
	ARC	76.23	71.34	86.84	78.33
	ABCDM	66.87	64.48	73.53	68.71
	Proposed	**86.91**	**87.29**	**86.91**	**86.85**

Furthermore, we compared our proposed model's performance with other state-of-the-art binary text classification methods mentioned above, and the results are shown in Table 4. The proposed model outperforms all the other models significantly. Interestingly, the CRNN, ARC, and ABCDM models, which all utilize convolutional layers in their architecture, delivered the best results, as shown in Table 3 for comparison of deep models.

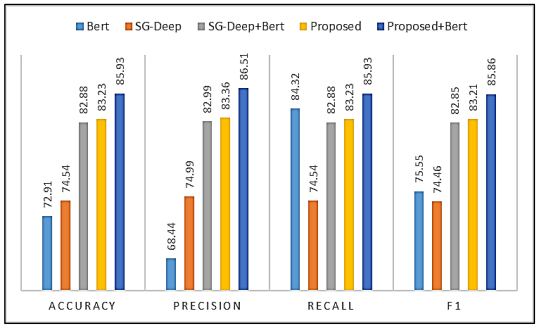
C. Ablation Study

To demonstrate the effectiveness of our proposed model, we carried out an ablation study. This involved eliminating different components of the model and evaluating the performance of the resulting models. We compared the performance of five models, as shown in Fig. 12. The first model, called Bert, only included the Bert branch of our proposed model and omitted the ML and Deep branches. The second model, SG-Deep, only included the deep learning branch and omitted the Bert and ML branches. The third model, SG-Deep+Bert, preserved the Bert and Deep branches but omitted the ML branch. The fourth model, Proposed, only preserved the ML and Deep branches and omitted the Bert branch. The fifth and final model, Proposed+Bert, utilized all three branches of our proposed model.

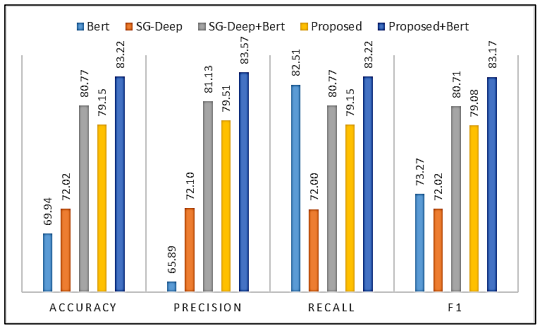
As shown in the figure, the performance of the Bert and SG-Deep models was significantly lower than the other models. However, the proposed model and SG-Deep+Bert models had similar performance, which demonstrates the effectiveness of the ensemble technique used in our proposed model. The diversity of algorithms in SG-Deep and the structural differences between deep models and Bert models make the ensemble results more accurate. Finally, the Proposed+Bert model achieved the best performance in all datasets, showing the effectiveness of using all three branches of our proposed model.

Conclusion

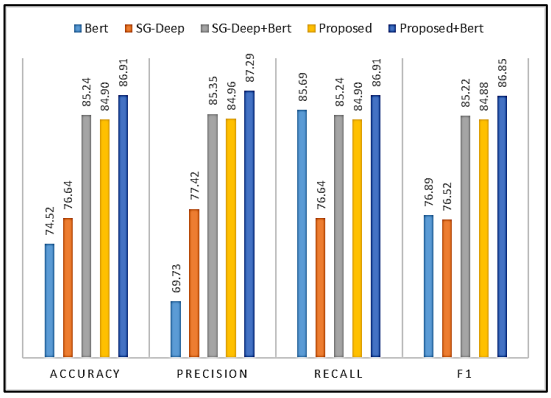
Our study proposes an ensemble model that combines deep learning and machine learning methods to detect the expertise shape of users based on their answers in Stack Overflow's CQA. To achieve this, we used seven ML models, five deep models, and a pre-trained transformer-based Bert model. Our model was able to process user answers and identify dash-shaped users. We conducted extensive experiments to evaluate our model's effectiveness, and the results across four different datasets of Stack Overflow answers demonstrate that our model outperforms both the ML and deep models used as its building blocks, as well as state-of-the-art deep models for binary classification of textual data. Our model is not limited to detecting dash-shaped users. It can also classify other shapes of expertise, such as T- and C-shaped users, which are valuable for forming agile software teams.



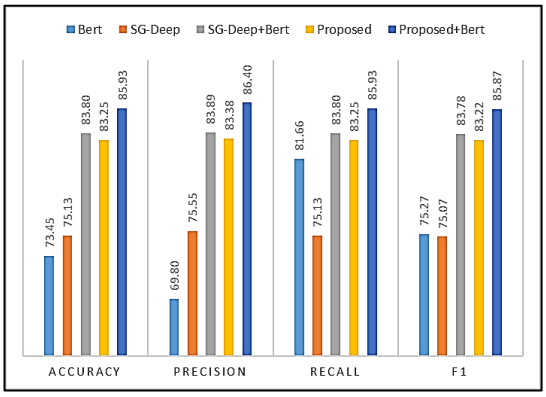
(a)



(b)



(c)



(d)

Fig. 12: Comparison of the performance of five models in the ablation study on the (a) All, (b) Android, (c) C#, and (d) Java datasets.

Additionally, our model can be used as a filter method for downstream applications, like intern recommendations. In future work, we plan to evaluate our model on similar problems in CQA texts and explore other deep ensemble models to further improve the performance of expertise shape classification problems.

Author Contributions

S. Nemati designed the experiments, analyzed the data, interpreted the results, and wrote the manuscript.

Acknowledgment

This work was financially supported by the research deputy of Shahrekord University (grant number 0GRD34M44264).

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviations

CQA	Community Question-Answering
BiLSTM	BiDirectional Long Short-Term Memory
RF	Random Forest
SVM	Support Vector Classifier
DT	Decision Tree
LR	Logistic Regression
MLP	Multi-Layer Perceptron
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network
AUC	Area Under Curve
CART	Classification And Regression Tree
ML	Machine Learning

References

[1] P. Rostami, M. Neshati, "Intern retrieval from community question answering websites: A new variation of expert finding problem," Expert Syst. Appl., 181: 115044, 2021.

[2] S. Yuan, Y. Zhang, J. Tang, W. Hall, J. B. Cabotà, "Expert finding in community question answering: a review," Artif. Intell. Rev., 53: 843-874, 2020.

[3] A. Dargahi Nobari, S. Sotudeh Gharebagh, M. Neshati, "Skill translation models in expert finding," in Proc. the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval: 1057-1060, 2017.

[4] H. Demirkan, J. Spohrer, "T-shaped innovators: Identifying the right talent to support service innovation," Res. Technol. Manage., 58(5): 12-15, 2015.

[5] V. Kumar, N. Pedanekar, "Mining shapes of expertise in online social Q&A communities," in Proc. the 19th ACM conference on Computer Supported Cooperative Work and Social Computing Companion: 317-320, 2016.

[6] S. S. Gharebagh, P. Rostami, M. Neshati, "T-shaped mining: A novel approach to talent finding for agile software teams," in Proc. Advances in Information Retrieval: 40th European Conference on IR Research: 411-423, 2018.

[7] C. P. Maertz Jr, P. A. Stoeberl, J. Marks, "Building successful internships: lessons from the research for interns, schools, and employers," Career Dev. Int., 19(1): 123-142, 2014.

[8] X. Fu, X. Sun, H. Wu, L. Cui, J. Z. Huang, "Weakly supervised topic sentiment joint model with word embeddings," Knowl. Based. Syst., 147: 43-54, 2018.

[9] H. Wang, K. Guo, "The impact of online reviews on exhibitor behaviour: evidence from movie industry," Enterp. Inf. Syst., 11(10): 1518-1534, 2017.

[10] D. Kundu, D. P. Mandal, "Formulation of a hybrid expertise retrieval system in community question answering services," Appl. Intell., 49: 463-477, 2019.

[11] S. Sorkhani, R. Etemadi, A. Bigdeli, M. Zihayat, E. Bagheri, "Feature-based question routing in community question answering platforms," Inf. Sci. (N Y), 608: 696-717, 2022.

[12] X. Zhang et al., "Temporal context-aware representation learning for question routing," in Proc. the 13th International Conference on Web Search and Data Mining: 753-761, 2020.

[13] H. Ding, Q. Liu, G. Hu, "TDTMF: A recommendation model based on user temporal interest drift and latent review topic evolution with regularization factor," Inf. Process. Manage., 59(5): 103037, 2022.

[14] P. Rostami, A. Shakery, "A deep learning-based expert finding method to retrieve agile software teams from CQAs," Inf. Process. Manage., 60(2): 103144, 2023.

[15] K. Balog, L. Azzopardi, M. de Rijke, "A language modeling framework for expert finding," Inf. Process. Manage., 45(1): 1-19, 2009.

[16] S. Liang, M. de Rijke, "Formal language models for finding groups of experts," Inf. Process. Manage., 52(4): 529-549, 2016.

[17] D. Petkova, W. B. Croft, "Hierarchical language models for expert finding in enterprise corpora," Int. J. Artif. Intell. Tools, 17(01): 5-18, 2008.

[18] M. Bouguessa, B. Dumoulin, S. Wang, "Identifying authoritative actors in question-answering forums: the case of yahoo! answers," in Proc. the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 866-874, 2008.

[19] H. Zhu, E. Chen, H. Xiong, H. Cao, J. Tian, "Ranking user authority with relevant knowledge categories for expert finding," World Wide Web, 17: 1081-1107, 2014.

[20] A. Daud, J. Li, L. Zhou, F. Muhammad, "Temporal expert finding through generalized time topic modeling," Knowl. Based Syst., 23(6): 615-625, 2010.

[21] S. Momtazi, F. Naumann, "Topic modeling for expert finding using latent Dirichlet allocation," Wiley Interdiscip. Rev. Data Min. Knowl. Discov., 3(5): 346-353, 2013.

[22] L. Yang et al., "Cqarank: jointly model topics and expertise in community question answering," in Proc. the 22nd ACM International Conference on Information & Knowledge Management: 99-108, 2013.

[23] N. Nikzad-Khasmakhi, M. Balafar, M. R. Feizi-Derakhshi, C. Motamed, "ExEm: Expert embedding using dominating set theory with deep learning approaches," Expert Syst. Appl., 177: 114913, 2021.

[24] M. Zhao, F. Javed, F. Jacob, M. McNair, "SKILL: A system for skill identification and normalization," in Proc. the AAAI Conference on Artificial Intelligence: 4012-4017, 2015.

[25] A. Azzam, N. Tazi, A. Hossny, "Text-based question routing for question answering communities via deep learning," in Proc. the Symposium on Applied Computing: 1674-1678, 2017.

[26] M. Dehghan, H. A. Rahmani, A. A. Abin, V. V. Vu, "Mining shape of expertise: A novel approach based on convolutional neural network," Inf. Process. Manage., 57(4): 102239, 2020.

[27] Z. Li, J. Y. Jiang, Y. Sun, W. Wang, "Personalized question routing via heterogeneous network embedding," in Proc. the AAAI Conference on Artificial Intelligence: 192-199, 2019.

[28] W. Tang, T. Lu, D. Li, H. Gu, N. Gu, "Hierarchical attentional factorization machines for expert recommendation in community question answering," IEEE Access, 8: 35331-35343, 2020.

[29] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," Artif. Intell. Rev., 52(1): 273-292, 2019.

[30] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown, "Text classification algorithms: A survey," Information, 10(4): 150, 2019.

[31] Q. Li et al., "A survey on text classification: From shallow to deep learning," arXiv preprint arXiv:2008.00364, 2020.

[32] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, J. Gao, "Deep learning–based text classification: a comprehensive review," ACM Comput. Surv. (CSUR), 54(3): 1-40, 2021.

[33] Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882, 2014.

[34] S. Yu, D. Liu, W. Zhu, Y. Zhang, S. Zhao, "Attention-based LSTM, GRU and CNN for short text classification," J. Intell. Fuzzy Syst., 39(1): 333-340, 2020.

[35] M. Zulqarnain et al., "Text classification using deep learning models: A Comparative review," Cloud Comput. Data Sci., 5(1): 80-96, 2024.

[36] A. Ezen-Can, "A comparison of LSTM and BERT for small corpus," arXiv preprint arXiv:2009.05451, 2020.

[37] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[38] S. González-Carvajal, E. C. Garrido-Merchán, "Comparing BERT against traditional machine learning text classification," arXiv preprint arXiv:2005.13012, 2020.

[39] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, U. R. Acharya, "ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis," Future Gener. Comput. Syst., 115: 279-294, 2021.

[40] S. Nemati, "Canonical correlation analysis for data fusion in multimodal emotion recognition," in Proc. 9th International Symposium on Telecommunication: With Emphasis on Information and Communication Technology, IST 2018, 2019.

[41] A. Mohammed, R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," J. King Saud Univ. Comput. Inf. Sci., 35(2): 754-774, 2023.

[42] J. Wang, L. C. Yu, K. R. Lai, X. Zhang, "Dimensional sentiment analysis using a regional CNN-LSTM model," in Proc. the 54th Annual Meeting of the Association for Computational Linguistics, (2: Short papers): 225-230, 2016.

[43] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, H. Veisi, "Sentiment analysis based on improved pre-trained word embeddings," Expert Syst. Appl., 117: 139-147, 2019.

[44] A. Chatterjee, U. Gupta, M. K. Chinnakotla, R. Srikanth, M. Galley, P. Agrawal, "Understanding emotions in text using deep learning and big data," Comput. Human Behav., 93: 309-317, 2019.

[45] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, "Hierarchical attention networks for document classification," in Proc. the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: 1480-1489, 2016.

[46] S. Wen, J. Li, "Recurrent convolutional neural network with attention for twitter and yelp sentiment classification: ARC model for sentiment classification," in Proc, the 2018 International Conference on Algorithms, Computing and Artificial Intelligence: 1-7, 2018.

[47] G. Liu, J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," Neurocomput., 337: 325-338, 2019.

Biographies

Shahla Nemati was born in Shiraz, Iran, in 1982. She received the B.S. degree in Hardware Engineering from Shiraz University, Shiraz, in 2005, the M.S. degree from the Isfahan University of Technology, Isfahan, Iran, in 2008, and the Ph.D. degree in Computer Engineering from Isfahan University, Isfahan, in 2016. Since 2017, she has been an Assistant Professor with the Computer Engineering Department, Shahrekord University, Shahrekord, Iran. She has written several articles in the fields of data fusion, emotion recognition, affective computing, and audio processing. Her research interests include data fusion, affective computing, and data mining.

- Email: s.nemati@sku.ac.ir
- ORCID: 0000-0003-2906-5871
- Web of Science Researcher ID: AAA-3341-2019
- Scopus Author ID: 24512475100
- Homepage: https://www.sku.ac.ir/~snemati#

How to cite this paper:
S. Nemati, "An effective ensemble of deep and machine learning methods for classifying the expertise shape of CQA users," J. Electr. Comput. Eng. Innovations, 12(2): 409-424, 2024.

DOI: 10.22061/jecei.2024.10621.724

URL: https://jecei.sru.ac.ir/article_2097.html