Journal of Information Technology Education: Innovations in Practice

An Official Publication of the Informing Science Institute
InformingScience.org

JITEiip.org

Volume 24, 2025

COMPARING ACCURACY OF BIDIRECTIONAL ENCODERS THROUGH ANALYSIS OF COMPUTER SCIENCE STUDENT SKILLSETS FOR EARLY DETECTION OF STUDENT COMPETENCIES AND WEAKNESSES

Jasmin Nizar*	Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India	jasminnizar.nizar@gmail.com
Sharmila R	Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, Tamil Nadu, India	sharmi.saravanan0521@gmail.com

* Corresponding author

ABSTRACT

Aim/Purpose	Skill assessment of students is a critical process that evaluates their proficiency, ranging from novice to mastery, in various domains. This paper aims to develop a framework for predicting students' multi-skillsets effectively.
Background	The global IT industry recognizes that for computer science students to excel, they must acquire and demonstrate proficiency in Soft Skills, Life Skills, and Technical Skills. This study addresses the need for skill-based education in software engineering to bridge the gap between academic learning and industry requirements.
Methodology	The study employs the Bidirectional Encoder Representations from Transformers (BERT) model to predict students' competencies in Soft, Life, and Technical skills. The effectiveness of the model is evaluated using key performance indicators, including precision, accuracy, F1-score, recall, and the area under the curve (AUC), as discussed in the paper.
Contribution	The paper introduces a multi-skillset framework that leverages AI techniques to predict students' skill levels, ensuring better alignment between academic training and industry expectations.

Accepting Editor Martin D Beer │ Received: May 21, 2025 │ Revised: August 18, September 14, 2025 │ Accepted: October 6, 2025.

Cite as: Nizar, J., & Sharmila, R. (2025). Comparing accuracy of bidirectional encoders through analysis of computer science student skillsets for early detection of student competencies and weaknesses. *Journal of Information Technology Education: Innovations in Practice, 24,* Article 23. https://doi.org/10.28945/5639

Findings	The proposed BERT model achieves a superior prediction accuracy of 96.25%, outperforming conventional machine learning baselines.
Recommendations for Practitioners	Educators and institutions should integrate AI-driven skill assessment models to personalize learning pathways and enhance students' industry readiness.
Recommendations for Researchers	Further exploration of deep learning techniques can refine skill prediction models, leading to more robust assessment frameworks.
Impact on Society	By improving the accuracy of skill assessments, this study contributes to producing future-ready software engineering professionals, ultimately enhancing workforce competency in the IT industry.
Future Research	Future research can focus on integrating BERT with hybrid deep learning models, such as BERT-GRU or BERT-CNN, to enhance the accuracy and robustness of predictions. Additionally, the model can be expanded to incorporate additional skill domains and be adapted for different educational disciplines.

Keywords skill assessment, software engineering, BERT, machine learning, multi-skillset

INTRODUCTION

Software is central to the digital age and continues to drive rapid technological change (Mahanti & Mahanti, 2005). Despite its significance, the software industry faces a persistent challenge – high failure rates in software projects. One contributing factor is the mismatch between the skills of software engineering graduates and the evolving demands of the IT sector. Many companies report that recent graduates often lack the practical knowledge and abilities required to perform software engineering tasks effectively (Radermacher et al., 2014). This issue is further exacerbated by traditional education systems that emphasize theoretical instruction over hands-on, experiential learning, thereby widening the gap between academic preparation and industry expectations.

Bridging this gap has become a critical priority for both educators and industry stakeholders. Structured skill development plays a vital role in addressing students' learning deficiencies and preparing them for successful careers. Beyond technical proficiency, students must also develop essential Soft and Life skills competencies that are increasingly recognized as critical for long-term professional growth. However, current higher education practices often assume that students will acquire these skills on their own, an assumption rarely fulfilled without structured support and targeted training programs.

In the realm of software development, three skill domains are indispensable: Technical skills, Soft skills, and Life skills (Makhoba et al., 2023; Mezhoudi et al., 2023). Technical skills encompass practical knowledge, specialized expertise, and the ability to apply tools, methods, and engineering practices effectively. Soft skills such as teamwork, leadership, problem-solving, and decision-making are essential for effective collaboration in project-based environments. Life skills, including adaptability, time management, and communication, contribute significantly to both individual and team success. Together, these technical and transversal skills form the foundation for thriving in today's competitive IT landscape.

An education system that integrates Technical competencies with Soft and Life skills is necessary to equip students for real-world challenges. Yet, teaching software engineering remains inherently complex due to its abstract concepts and reliance on hypothetical examples within academic curricula. A curriculum that emphasizes analytical thinking, problem-solving, communication, and interpersonal skills – alongside technical instruction – is required to inspire students and better align academic outcomes with industry needs.

This study proposes a predictive framework for evaluating the skill sets of computer science students, incorporating Soft, Life, and Technical competencies. Leveraging natural language processing and deep learning – specifically the Bidirectional Encoder Representations from Transformers (BERT) model – this research offers a robust methodology for assessing and forecasting students' skill levels. By identifying specific skill gaps and recommending targeted interventions, the framework aims to improve the quality of software engineering education, produce industry-ready graduates, and ultimately contribute to higher success rates in software development projects. The highlights of the paper are presented below:

(1) A comprehensive multi-skillset framework is proposed to predict Life skills, Technical skills, and Soft skills.
(2) The BERT model is applied to effectively predict the software engineering skill sets of computer science graduates with high accuracy.
(3) A customized dataset is developed and used for training and assessing the model.
(4) The performance of the BERT model is evaluated using metrics such as precision, accuracy, F1-score, recall, and AUC value.

The document is organized into separate sections. The initial section provides a concise summary of techniques and approaches for estimating students' skill levels. The motivation behind the study and a summary of related works are then discussed, followed by an overview of the datasets used and the methods employed. The proposed framework used for comparison is presented. Finally, the investigation's findings, along with recommendations for future research and an outline of its future scope, are presented.

MOTIVATION

Career success in the rapidly evolving field of software engineering depends on possessing a diverse range of skills. However, there is frequently a disconnect between academic training and business expectations, as traditional educational systems struggle to provide students with the required blend of Technical, Soft, and Life skills. The necessity to close this gap by using machine learning methods, more specifically BERT, to forecast software engineering students' multi-skills is what spurred this study. Personalized learning pathways are made possible by accurately forecasting individuals' skill sets, which enables educational institutions to create focused interventions and curriculum improvements. This ensures that students not only acquire technical knowledge but also develop critical problem-solving, collaboration, flexibility, and other essential skills needed in work settings.

Multi-skillset prediction provides teachers with data-driven insights into their students' areas of strength and weakness, enabling them to develop structured training programs and adaptive teaching strategies. By ensuring a closer match between graduate competencies and job requirements, this predictive capability helps industries reduce onboarding time and improve workforce readiness. Additionally, recognizing and developing various skill sets early on can have a significant impact on students' career paths, especially as software engineering projects increasingly emphasize cooperation and agile approaches. This study helps create a workforce prepared for the future by utilizing AI-driven skill assessment, ensuring that graduates meet the ever-changing demands of the IT sector.

RELATED WORKS

The IT industry stands as a cornerstone of progress, with software engineers serving as pivotal contributors behind the scenes. Their diverse skill sets are essential for driving innovation, addressing customer requirements, ensuring quality and security, boosting operational efficiency, adapting to technological advancements, fostering teamwork, and solving intricate challenges. As technology continues to evolve, the demand for software engineering expertise in the IT sector is only expected to rise. Consequently, building a strong foundation in software engineering is not merely advantageous but a critical requirement for those aiming to excel in this ever-evolving domain. Numerous studies have focused on identifying the skills required for securing software development roles in the IT sector.

To better align with the needs of the software industry, Mahanti and Mahanti (2005) analyzed the state of software engineering education in India and suggested improvements. Nizar et al. (2024a, 2024b) developed prediction frameworks that combine Random Forest, CNN, PCA, and Explainable AI (XAI) to assess students' technical, soft, and life skills. Fávero and Casanova (2021) utilized BERT to identify relevant phrases in software engineering using a requirements classification task. Sventekova and Lovecek (2012) aimed to identify students' weaknesses in project-based learning to bridge the academia-industry gap. Similarly, Gnatz et al. (2003) advocated for SE education that goes beyond theory and fosters practical and non-technical skills. Petrovska et al. (2025) designed exercises to improve students' code evaluation abilities using human and AI-generated samples. Al-Ahmad et al. (2020) proposed a hybrid model combining feature selection and classification to evaluate students' performance throughout the development lifecycle. Siddique et al. (2024) examined the challenges in the freelance sector and offered training recommendations.

Noveski et al. (2024) evaluated the accuracy and impact of LLMs on educational content, contributing to understanding AI's role in future learning systems. Khan et al. (2021) highlighted conceptual gaps between CS I and CS II courses. Mahon et al. (2024) offered adaptable guidelines for integrating generative AI into CS education. Sengul et al. (2024) found a misalignment between academic education and industry needs. Garousi et al. (2019a, 2019b) emphasized this disconnect, showing how it creates early-career challenges for graduates. Akdur (2022) and Scaffidi (2018) underlined the importance of Soft and Technical skills for workplace readiness. Gnatz et al. (2003) described a course structure focused on imparting practical skills to students. Marjan et al. (2023) introduced an EDM system for assessing student programming proficiency. Mezhoudi et al. (2023) identified employability factors aligned with modern workforce demands. Makhoba et al. (2023) proposed using skill traits to match students to suitable programs and identify at-risk learners. Kirova et al. (2024) emphasized the need to adapt SE education to keep pace with advances in AI and LLM.

Employers frequently cite interaction, cooperation, and adaptability as essential for success in the job (Lizzio & Wilson, 2004). According to the World Health Organization (1999), life skills include psychological skills. Employability still depends on technical skills, which are based on disciplinary knowledge and problem-solving frameworks, such as Bloom et al.'s (1964) taxonomy; however, these are insufficient on their own. According to research, graduates who possess strong technical capabilities but lack interpersonal or adaptability skills may struggle to transition into industry employment (Heckman & Kautz, 2012). To better align education with industry needs, recent studies have advocated for the adoption of integrated approaches that consider multiple dimensions of competency (Zawacki-Richter et al., 2019). AI provides a promising avenue in this regard – ML models have been effectively used to detect at-risk students, forecast student performance, and assist individualized learning interventions.

From this review, several research gaps emerge. While numerous studies, such as those by Sengul et al. (2024) and Garousi et al. (2019a, 2019b), stress the disconnect between academic curriculum and industry demands, few propose specific frameworks to bridge this gap, particularly in integrating LLMs and AI. Although tools like those discussed by Noveski et al. (2024) and Marjan et al. (2023) explore AI's educational potential, more research is needed to embed them meaningfully into SE curricula to enhance both Soft and Technical skills. Although frameworks by Al-Ahmad et al. (2020) and Nizar et al. (2024a, 2024b) offer valuable prediction and assessment models, a comprehensive system that evaluates student readiness across Technical, Soft, and Life skills in real-world contexts remains lacking.

BERT (Devlin et al., 2019) and other NLP models have demonstrated an exceptional capacity to capture contextual meaning, surpassing more conventional deep learning models, such as RNNs, CNNs, and GRUs. Although academic prediction tasks have been the primary focus of AI applications in education, little research has been conducted on the comprehensive prediction of multi-skillsets that incorporate Life, Soft, and Technical skills. The current study fills this gap by placing skill prediction at the nexus of AI-driven analytics and educational theory, providing a framework for more thorough and useful insights into student growth. This study introduces a BERT-based multi-skillset prediction model to assess Life, Soft, and Technical competencies among computer science students, aiming to bridge the industry-academia divide.

METHODS EMPLOYED

BERT

The BERT approach was used to develop the suggested model. A potent transformer-based model called Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) was created for a variety of Natural Language Processing (NLP) applications. The two main components of its framework are pre-training and fine-tuning, and using self-supervised tasks, such as Masked Language Models (MLM), which involve masking randomly chosen tokens in a sentence and predicting them based on their context, along with Next Sentence Prediction (NSP), which entails assessing whether one sentence logically follows another. BERT is trained on large unlabelled datasets during the pre-training phase. In contrast to conventional unidirectional models, BERT's bidirectional training allows it to comprehend a word's context by concurrently examining its left and right surroundings.

Using labeled information, the previously trained BERT model is adjusted to specific downstream tasks during the fine-tuning stage. The system is initialized using pre-trained settings, and during this phase, all settings are fine-tuned. Although they all start with the same pre-trained parameters, each downstream task, such as text categorization, question answering, or sentiment analysis, has its own optimized model. Fine-tuning allows BERT to perform exceptionally well on diverse tasks by leveraging its deep contextual understanding. BERT's design incorporates special tokens like [CLS] (used for sequence classification tasks) and [SEP] (to separate sentences), transformer encoder layers for self-attention, and embedding layers that combine token, segment, and positional embeddings. These components enable it to capture complex linguistic relationships and nuances effectively. By supporting bidirectional attention and context-rich embeddings, BERT offers cutting-edge performance in tasks involving language comprehension and reasoning, marking a substantial improvement in natural language processing. Figure 1 (Devlin et al., 2019) shows the summary of BERT's processes for pre-training and fine-tuning.

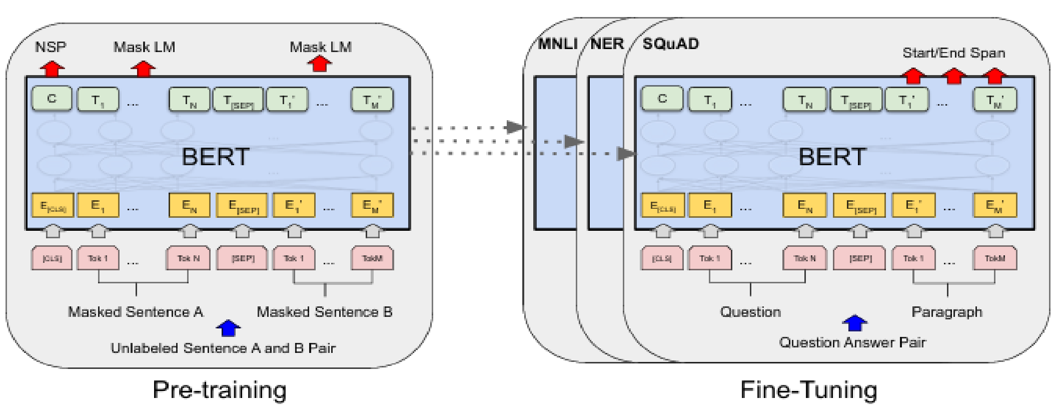

Figure 1. Summary of BERT's pre-training and fine-tuning processes

COMPARISON MODELS

The models used for comparison in this study are the gated recurrent unit (GRU), long short-term memory (LSTM), convolutional neural networks (CNNs), recurrent neural networks (RNNs), and the transformer model.

Convolutional neural networks

Convolutional neural networks (CNNs) (Younesi et al., 2024) are mainly used for processing structured grid data. Inspired by the visual processing in the human brain, CNNs utilize convolutional layers that apply filters to input data in a sliding window fashion. A standard CNN structure comprises several convolutional layers, followed by pooling layers, and fully connected layers that generate predictions using the extracted features. By employing weight sharing within convolutional layers, CNNs minimize the number of parameters, enabling efficient learning from extensive datasets. Due to their success in computer vision, CNNs have also been adapted for applications in natural language processing and other domains requiring spatial feature extraction. Figure 2 (Nizar et al., 2024b) shows the structure of the CNN.

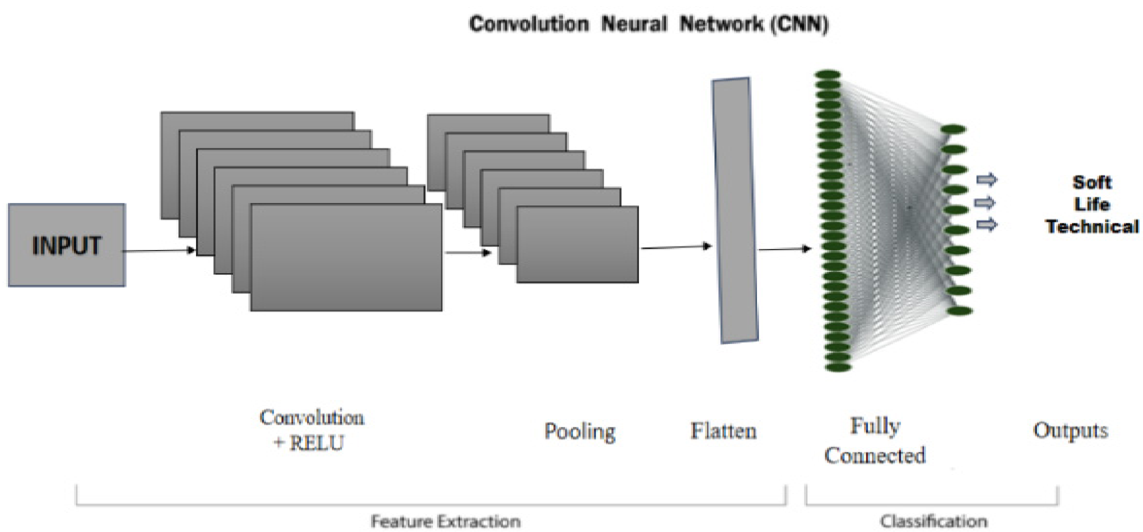

Figure 2. Structure of CNN

Recurrent neural networks

Recurrent neural networks (RNNs) (Yu et al., 2019) are designed to handle sequential data by retaining information about previous inputs within their hidden states. RNNs employ loops in their architecture, which enables information to survive over time steps. This makes them especially useful for applications such as speech recognition and natural language processing. However, studying long-term dependencies is challenging for standard RNNs due to issues like vanishing gradients. By adding gating techniques that control the information flow, variants like LSTM and GRU solve these problems and improve the RNN's ability to capture long-term dependencies. At each time step, input vectors (x1, x2, x3, …, xn) are supplied into the network. As a memory system, the hidden states (h0, h1, h2, …, hn) record details about the sequence up to the present step. The current input (x_t) and the previous hidden state (ht-1) are used to update each hidden state (ht). The outputs (y1, y2, y3, …, yn) are generated at each time step, typically derived from the corresponding hidden states. Figure 3 (Prestes et al., 2021) shows the architecture of an RNN.

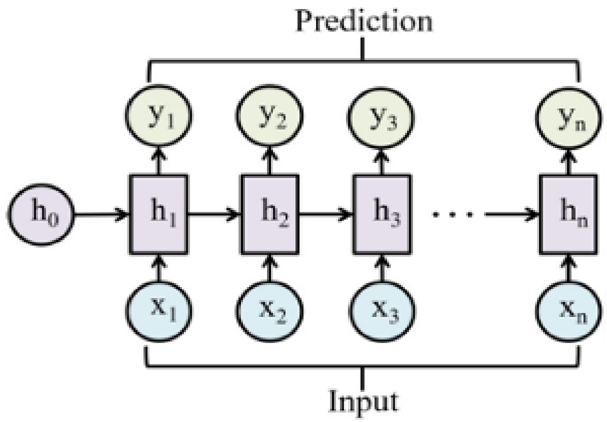

Figure 3. Architecture of RNN

Long short-term memory

An RNN type called long short-term memory (LSTM) (Ghojogh & Ghodsi, 2023) was developed to address the gradient disappearance issue that conventional RNNS encounter when attempting to learn long-term relationships in sequential input. Memory cells with the ability to retain information for an extended period are part of the special architecture of LSTMs. Three key gates govern these memory cells: the input gate (i_t) manages what data is allowed to be added to the cell state; the gate that outputs (o_t) the data controls what data is passed on to the next layer; the forget gate (ft) determines which information from the previous cell state is unnecessary and should be discarded. LSTMS are especially useful for jobs involving sequential data. Figure 4 demonstrates the LSTM layout (Jaseena & Kovoor, 2021).

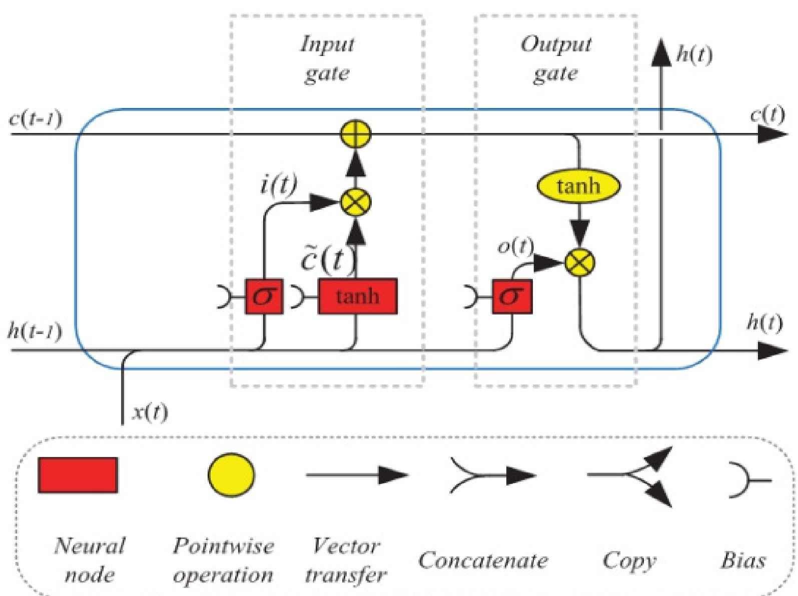

Figure 4. Layout of LSTM

Gated recurrent unit

One type of RNN architecture, called the gated recurrent unit (GRU) (Chung et al., 2014), was developed to address some of the limitations of conventional RNNs, specifically the challenge of learning long-term dependencies. GRU is a more straightforward version of LSTM networks. To avoid problems such as vanishing and exploding gradients during training, they employ gating techniques to regulate the information flow. The update gate (z_t) and the reset gate (r_t) are the two main gates of the GRU. While the reset gate regulates how much of the prior data should be forgotten, the update gate decides how much of the previous hidden state should be retained. These gates make the GRU effective for tasks involving sequential data by enabling it to determine which information is crucial for the present prediction adaptively. Figure 5 displays the schematic for GRU (Yu et al., 2019).

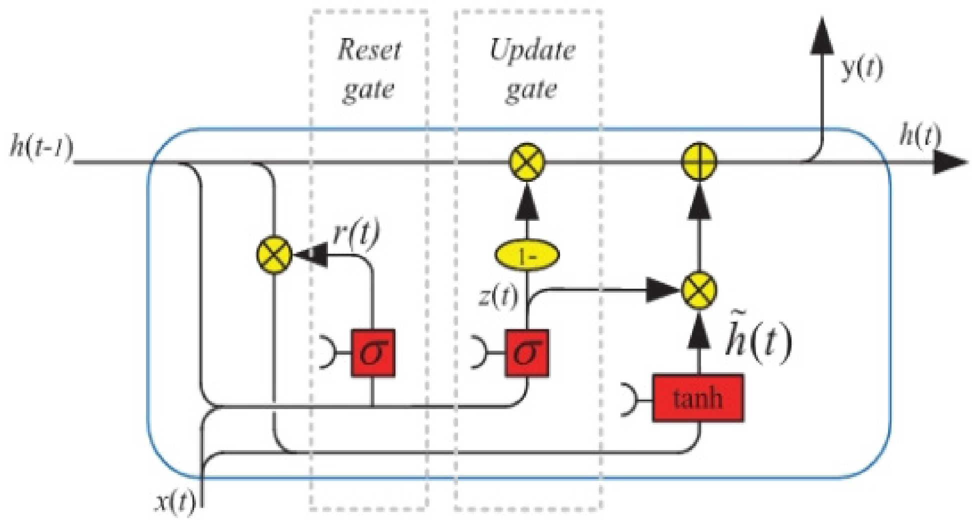

Figure 5. Schematic of GRU

Transformer model

NLP was transformed by the vanilla transformer model (Vaswani et al., 2017), a deep learning architecture that substituted self-attention mechanisms for RNN and CNN. The transformer model handles all input tokens simultaneously, enabling parallelization and significantly increasing training efficiency compared to the RNN, which processes data in a sequential manner. The transformer's self-attention system enables the model to evaluate the relative importance of words in a sequence, independent of their order. Because this attention method is used at both the encoder and decoder levels, it works especially well for tasks like language modeling, text summarization, and machine translation. Figure 6 (Vaswani et al., 2017) displays the layout of the transformer model.

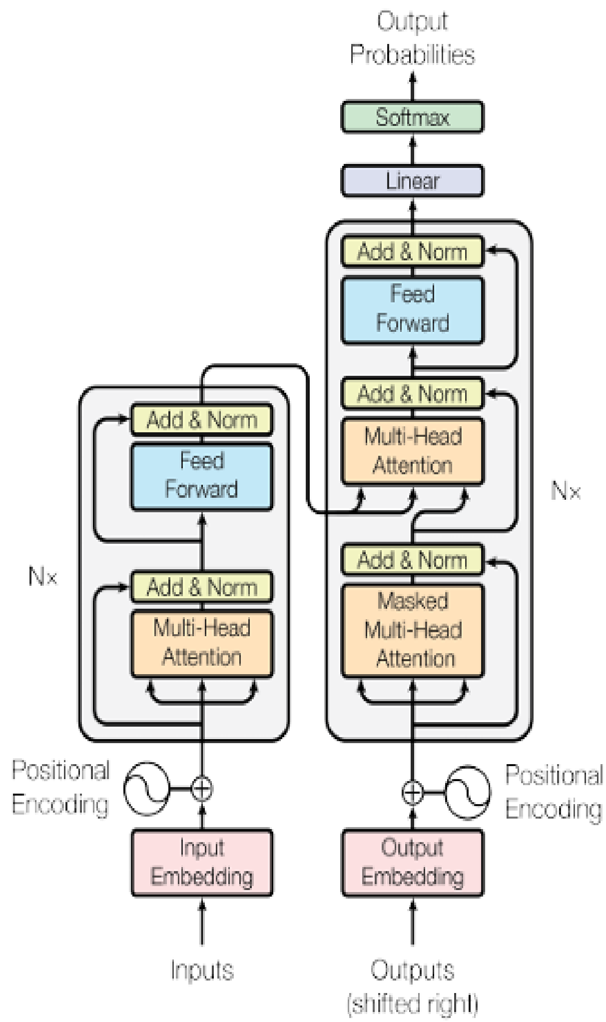

Figure 6. Layout of transformer model

PROPOSED FRAMEWORK

In order to forecast different software engineering skill sets among students studying computer science, this study uses the BERT algorithm. Data collection, preprocessing, feature extraction, model selection, training, and evaluation are some of the essential processes in the methodology. Figure 7 provides a summary of the suggested methods.

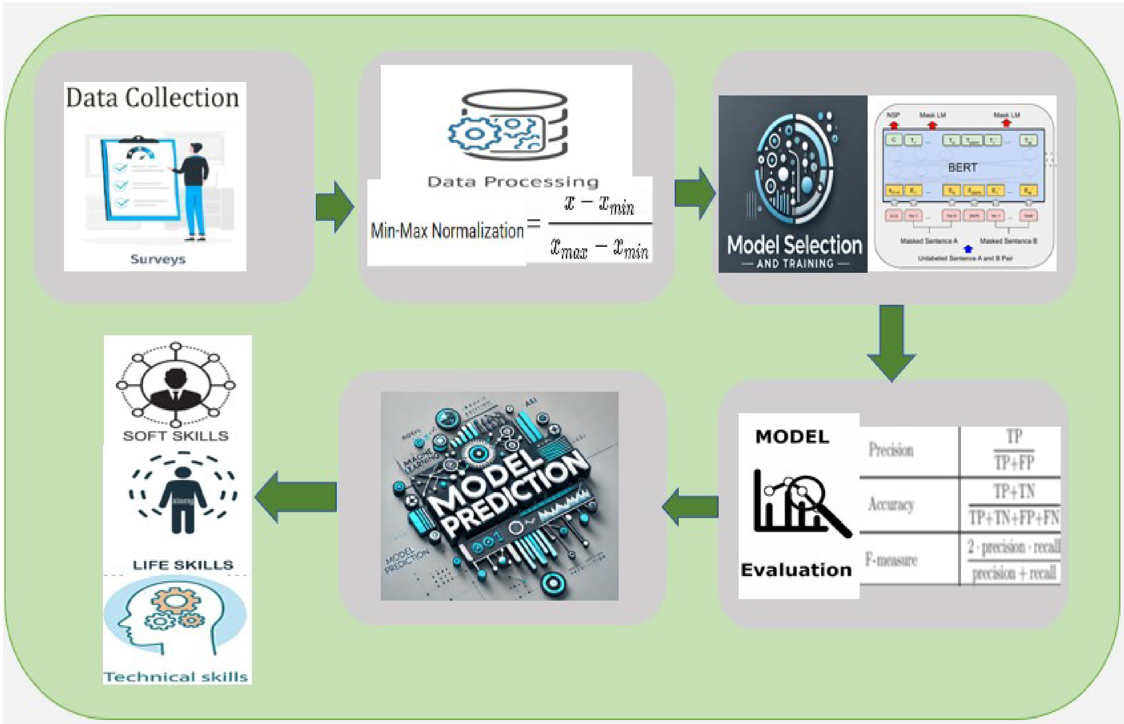

Figure 7. Proposed methodology

Data Collection and Preprocessing

The effectiveness of any prediction model significantly depends on the reliability and strength of the dataset. A thorough survey of computer science students from different colleges in Kerala, India, served as the main source of data for this research. Students who have taken software engineering as part of their coursework were the target population of the survey. The study focused on evaluating three core skill categories – soft skills, life skills, and technical skills – identified through a comprehensive literature review (Akdur, 2022; Lizzio & Wilson, 2004; Nizar et al., 2024a, 2024b; Robles, 2012; Scaffidi, 2018; Sunindijo, 2015) as critical for software engineering readiness. The questionnaire was meticulously designed with three distinct sections. To comprehensively assess each skill area, including a total of 54 questions, with each question offering six possible answers, scored on a scale from 0 to 5.

The survey and quiz were administered separately but analyzed together, featuring five questions from each skill category, totaling 15 questions on the quiz. The survey measured students' self-assessed proficiency in each skill category, while the quiz evaluated their demonstrated applied knowledge. The results from both instruments were combined to provide a composite skill assessment. Students were classified as "competent" in a skill category only if they achieved ≥60% in the quiz and also scored above the average for that category in the survey. The survey and quiz combined gathered responses from 2,004 students, resulting in a dataset with over 2,000 samples. The features of soft, life, and technical skills are detailed in the Appendix (Tables A1, A2, and A3). The complete software engineering skillset dataset used in this study has been published in the Mendeley dataset (Nizar et al., 2024c). The dataset, central to this study, plays a vital role in assessing students' readiness for the IT industry, and the accuracy of the prediction model is largely dependent on its integrity and completeness. Figure 8 (Nizar et al., 2024c) illustrates a sample screenshot of the dataset used in this research. The following are sample survey queries that are intended to assess students' skill sets in three different areas:

Soft skills assessment

1. Do you feel confident managing a group of more than 20 people?
2. How at ease are you conversing with people in your close social circle?
3. Can you effectively contribute to group discussions to stimulate creativity?
4. Do you prefer delegating decision-making responsibilities to others?
5. Are you adaptable enough to adjust your schedule when faced with unexpected events?

Life skills assessment

1. How skilled are you in resolving conflicts within a team setting?
2. Do you actively listen during group discussions or collaborative tasks?
3. Are you confident speaking in front of bigger crowds?
4. What level of interpersonal communication proficiency do you possess?
5. Do you consider yourself an effective leader in inspiring and building a cohesive team?

Technical skills assessment

1. Would you rather study technical and managerial disciplines together in a curriculum than just one?
2. Are you willing to explore and learn new tools and techniques?
3. How confident are you in completing a project independently?
4. Do you think time, cost, or resource management is more critical than technical challenges in team projects?
5. Has quality management been a continuous feature of your project work?

(1-18 QUESTIONS FO	2.Are you confident to n	3.Have you struggled to	4.Have you faced any str	5.How confident are you	6.How confident are you	7.Do you multitask in yo	8.Do you tend to work o	9.How do you stay moti	10.How do you deal with	11.Would you pre
SOMETIMES	SOMETIMES	SOMETIMES	SOMETIMES	SOMETIMES	SOMETIMES	NEVER	NEVER	SOMETIMES	OFTEN	SOMETIMES
RARELY	NEVER	SOMETIMES	OFTEN	MOSTLY	RARELY	MOSTLY	ALWAYS	OFTEN	OFTEN	MOSTLY
ALWAYS	MOSTLY	MOSTLY	OFTEN	ALWAYS	OFTEN	RARELY	MOSTLY	SOMETIMES	SOMETIMES	MOSTLY
MOSTLY	MOSTLY	OFTEN	SOMETIMES	MOSTLY	OFTEN	SOMETIMES	SOMETIMES	SOMETIMES	OFTEN	SOMETIMES
OFTEN	NEVER	MOSTLY	SOMETIMES	ALWAYS	MOSTLY	NEVER	MOSTLY	RARELY	MOSTLY	SOMETIMES
SOMETIMES	SOMETIMES	SOMETIMES	ALWAYS	ALWAYS	SOMETIMES	RARELY	NEVER	SOMETIMES	MOSTLY	ALWAYS
MOSTLY	OFTEN	SOMETIMES	SOMETIMES	MOSTLY	OFTEN	OFTEN	SOMETIMES	RARELY	OFTEN	RARELY
MOSTLY	ALWAYS	MOSTLY	SOMETIMES	ALWAYS	SOMETIMES	RARELY	SOMETIMES	MOSTLY	OFTEN	NEVER
OFTEN	SOMETIMES	SOMETIMES	SOMETIMES	RARELY	SOMETIMES	RARELY	MOSTLY	SOMETIMES	RARELY	SOMETIMES
MOSTLY	ALWAYS	MOSTLY	MOSTLY	OFTEN	OFTEN	OFTEN	ALWAYS	OFTEN	SOMETIMES	RARELY
MOSTLY	MOSTLY	OFTEN	OFTEN	ALWAYS	OFTEN	RARELY	RARELY	SOMETIMES	MOSTLY	SOMETIMES
MOSTLY	MOSTLY	SOMETIMES	SOMETIMES	OFTEN	SOMETIMES	ALWAYS	ALWAYS	MOSTLY	MOSTLY	OFTEN
MOSTLY	NEVER	MOSTLY	MOSTLY	SOMETIMES	SOMETIMES	ALWAYS	OFTEN	RARELY	SOMETIMES	SOMETIMES
ALWAYS	OFTEN	OFTEN	SOMETIMES	MOSTLY	SOMETIMES	RARELY	MOSTLY	SOMETIMES	SOMETIMES	SOMETIMES
SOMETIMES	RARELY	ALWAYS	MOSTLY	OFTEN	SOMETIMES	MOSTLY	ALWAYS	OFTEN	OFTEN	OFTEN

Figure 8. Dataset screenshot

The dataset includes 36 features obtained from questionnaire and quiz responses, covering various aspects of soft, life, and technical skills, based on inputs from over 2,000 students. The class distribution of the multi-skillset dataset is summarized in Table 1.

Table 1. Class distribution of multi-skillset

Class	Label
0	No Skill
1	Life Skill
2	Soft Skill
3	Technical Skill
4	Life and Soft Skills
5	Life and Technical Skills
6	Soft and Technical Skills
7	All Skills

The multi-skillset classes (0–7) were created based on whether students achieved competence in life skills, soft skills, and technical skills. Each student was assigned to one of eight mutually exclusive categories representing all possible combinations of these three skills. The numbering from 0 to 7 was used only for dataset processing. The order follows a binary combination logic, where Class 0 (000) means no skills, Class 1 (100) means only life skills, Class 2 (010) means only soft skills, Class 3 (001) means only technical skills, Class 4 (110) means life and soft skills, Class 5 (101) means life and technical skills, Class 6 (011) means soft and technical skills, and Class 7 (111) means competence in all three skills.

Effective data preprocessing techniques are essential for preparing a dataset before developing machine learning models. Properly processed data ensures that models can efficiently detect patterns and relationships. Training a model with incomplete or missing data is not feasible, as such data cannot provide reliable insights. Data preparation takes up a large amount of time during the process of building a model, about 90% of it. To guarantee the dataset's completeness, missing values were filled in using the corresponding feature's mean value. For normalization, various techniques were explored to scale the data. Initially, the standard scaler was applied, which provided reasonable outcomes. However, the Min-Max normalization method delivered superior performance, which improved the model's capacity to absorb information from the dataset.

Min-Max normalization is a broadly used data preprocessing technique employed in this study to standardize numerical features. It is particularly popular in machine learning for rescaling data to a specific range, between 0 and 1. This method adjusts the dataset such that the smallest value becomes 0 and the largest value becomes 1, effectively transforming all features to fall within the range [0, 1]. This is the most common choice due to its simplicity and compatibility with various machine learning models.

Min-Max normalization ensures that the scale of features does not disproportionately influence the learning process, particularly in distance-based algorithms like K-Nearest Neighbors or clustering techniques. Maintaining proportional relationships between data points helps models converge more quickly and improve their performance. The transformation is achieved using a mathematical formula, as shown in Eq. (1), where x' represents the normalized value, x is the original data point, and the minimum and maximum values of x determine the scaling factor. This technique is not only efficient but also preserves the original distribution of data (Jaseena & Kovoor, 2021).

$$\boldsymbol{x'} = \frac{\boldsymbol{x - min}}{\mathbf{max} - \boldsymbol{min}} \quad (1)$$

MODEL SELECTION AND TRAINING

BERT is a transformer-based model developed by Google, designed for query response and classification of texts. In the context of multi-skillset prediction, BERT processes text through multiple stages to achieve a nuanced understanding and effective classification. Initially, BERT employs a specialized tokenizer to split input text into subword units, adding special tokens such as [CLS] (indicating the start of the sequence) and [SEP] (marking sentence boundaries). For instance, the sentence "Student has teamwork experience rated 4" would tokenize into ['[CLS]', 'Student', 'has', 'team', '##work', 'experience', 'rated', '4', '[SEP]']. These tokens are transformed into dense vector representations, known as embeddings, which capture the context of each token. BERT combines three types of embeddings: token embeddings, which represent individual tokens; segment embeddings, which distinguish between different sentences in a pair; and positional embeddings, which retain the order of tokens within the sequence.

The foundation of BERT's design consists of transformers that analyze word relationships in both directions using self-attention techniques. Because of this bidirectional attention, BERT is able to fully understand terms like "teamwork experience," taking into account both the words that come before and after it at the same time. The two stages of BERT's framework are pre-training and fine-tuning. Using tasks like NSP, which evaluates whether a pair of sentences logically follow one another, and MLM, which masks specific textual tokens and predicts them based on their context, BERT is trained on sizable, unlabeled datasets during the pre-training stage. BERT gains a profound comprehension of language through these exercises.

BERT is modified for particular purposes, such as skill-set classification, during the fine-tuning stage. Labeled datasets are used to refine the model after it has been initialized using pre-trained parameters. To create probabilities for predetermined class labels, for instance, the final embedding of the [CLS] token is passed through a classifier layer after processing text describing student skill ratings (e.g., "teamwork: 4, multitasking: 5"). Making use of its prior training, BERT effectively learns patterns and relationships from the labeled data during fine-tuning. Table 2 summarizes the key parameters that influence the training process.

Table 2. The optimum values for the parameters used by the model

Parameter	Value	Description
Number of Examples	1603	Total number of training trials used in the dataset.
Number of Epochs	25	The total count of iterations in which the training dataset is processed by the system.
Instantaneous Batch Size (per device)	8	Number of samples per batch on each device during training.
Learning Rate	2.00E-05	Learning rate used during fine-tuning.
Optimizer	AdamW	Optimizer with weight decay specifically designed for transformer models.
Total Train Batch Size (with parallel, distributed & accumulation)	8	Effective batch size considering parallelism, distribution, and gradient accumulation.
Gradient Accumulation Steps	13	Number of gradient accumulation steps before a backward pass.
Total Optimization Steps	400	Total number of optimization steps in the training process.
Count of Trainable Parameters	10,94,88,392	Total count of parameters that are trainable in the model.

Hyperparameter tweaking was meticulously carried out for all models, including BERT, CNN, LSTM, RNN, GRU, and Transformer-based techniques, to maximize model performance and increase the algorithm's efficiency. To achieve the best results, several important hyperparameters were carefully changed, including the number of hidden layers, the number of processing nodes per layer, the maximum number of training epochs, and activation functions. While the number of layers and neurons in hidden layers was adjusted for LSTM, RNN, GRU, and Transformer models, the best kernel size, pooling layer configuration, and number of dense layers were chosen for CNN models. To optimize performance in the instance of BERT, other hyperparameters, including learning rate, batch size, and optimizer selection, were also investigated.

Previous studies have demonstrated that training and cross-validation times are inherently longer for large datasets (Kumar et al., 2021). A comparison study of several model configurations was carried out to assess the influence of these hyperparameters, as seen in Table 3. Larger datasets inherently require longer training and cross-validation times; hence, computational cost and model training time were also evaluated. According to the study, BERT outperformed other models in terms of multi-skillset prediction, but it also came at a higher computational cost. The findings draw attention to the trade-off between training efficiency and accuracy, highlighting the significance of choosing the best configuration given the resources at hand.

Table 3. The algorithms' comparative factors

Classification algorithms	Factors	Count or type
CNN	1D Convolution layer	(kernel_size=3,
		activation='relu')
	Pool layer	(pool_size=2)
	No of dense layers	5
	Number of neurons in Dense layers	1024, 512, 256,128,64
	Activation function of the dense layers	Softmax
	Activation function of the output layer	Linear
	Maximum iteration of epochs	200
LSTM, RNN, GRU, TRANSFORMER	Number of layers	5
	Number of neurons in the hidden layer	1024, 512, 256,128,64
	Activation function of the hidden layers	Softmax
	Activation function of the output layer	Linear
	Maximum iteration of epochs	200

MODEL EVALUATION

To gauge the efficiency of the suggested framework, the metrics of precision, accuracy, recall, and F1-score are employed, as they are among the most commonly used for classification tasks. These metrics provide valuable insights into model effectiveness. Higher values for these indicators generally reflect better model performance. In the case of an ideal classifier, both precision and recall would achieve a value of one.

As specified by Equation (2), accuracy is the ratio of correctly identified instances to all occurrences in the dataset. Precision evaluates the percentage of relevant results among all predictions made by the model, as indicated in Equation (3). According to Equation (4), recall measures the model's capacity to accurately recognize real positive cases. Last, the F1-score, which is specified in Equation

(5), offers an evaluation of the model's performance by combining accuracy and recall into a single metric by computing their harmonic mean (Zhu et al., 2019).

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

$$Precision = \frac{Number\ of\ retrieved\ and\ relevant\ observations}{Total\ retrieved\ observations} = \frac{TP}{TP+FP} \tag{3}$$

$$Recall = \frac{Number\ of\ retrieved\ and\ relevant\ observations}{Total\ relevant\ observations} = \frac{TP}{TP+FN} \tag{4}$$

$$F1\ Score = \ 2 * \frac{(recall * precision)}{(recall + precision)} \tag{5}$$

When the model accurately classifies pertinent data as such, it is referred to as a true positive (TP). When the model wrongly classifies irrelevant data as relevant, it produces false positives (FP). False negatives (FN), on the other hand, are pertinent occurrences that the model misses and labels as irrelevant. Measures that range from 0 to 1 are the true positive rate (TPR) and the false positive rate (FPR). Better model performance is indicated by a greater area under the curve (AUC).

The AUC, a critical parameter that assesses the model's ability to distinguish between positive and negative classes across a range of threshold values, is obtained from the ROC curve. The trade-off between TPR and FPR is shown visually by the ROC curve. Generally speaking, AUC values fall between 0.5 and 1, with 0.5 denoting no discriminative capacity and a number nearer 1 denoting nearly flawless classification performance.

RESULTS AND DISCUSSIONS

OPTIMAL FEATURES USING XAI

A critical step in improving model performance is feature selection, which finds the most pertinent properties while removing unnecessary or inconsequential ones. Thirty-six elements that represent different facets of soft, life, and technical skills (Appendix - Tables A1, A2, and A3) were extracted from questionnaire and quiz responses and used in this study. However, not every feature had an equivalent impact on prediction accuracy; therefore, a methodical reduction procedure was necessary. XAI techniques, such as SHAP, LIME, and LRP, were employed to rank the significance of the features and exclude less significant qualities, thereby refining the feature set. By offering transparency in feature selection and emphasizing the characteristics that most strongly influenced skill prediction, XAI played a crucial part. This promoted confidence in the outcomes by guaranteeing that the selection procedure was both data-driven and interpretable. SHAP, LIME, and LRP provide insights into the connections between the chosen features and skill categories by quantifying the contribution of each feature.

The SHAP analysis (Figure 9) generated a summary plot visualizing global feature importance, identifying key features that had the most significant influence on predictions. The LIME interpretation (Figure 10) provided localized explanations by ranking features based on their impact on Class 1 (high skill level) vs. Class 0 (low skill level) predictions. Additionally, the LRP heatmap (Figure 11) further highlighted the most influential features in each skill category, identifying the top-ranking attributes for soft, life, and technical skills. Following these analyses, the 19 most predictive features were selected, representing the core competencies across all three skill domains. Features with low SHAP values or weak contributions in LIME and LRP were eliminated, reducing redundancy and enhancing the model's generalizability. The selection process ensured that only the most informative attributes were retained, improving both interpretability and predictive accuracy.

To leverage these predictive features effectively, the BERT technique was employed. BERT's ability to understand contextual relationships in data enabled the development of a highly accurate prediction model. The combined use of XAI and BERT ensured a balance between model interpretability and performance, enhancing the overall reliability of the study. This advanced machine learning approach not only predicted students' competencies with precision but also offered actionable insights into their skill gaps, empowering educators and students to make informed decisions.

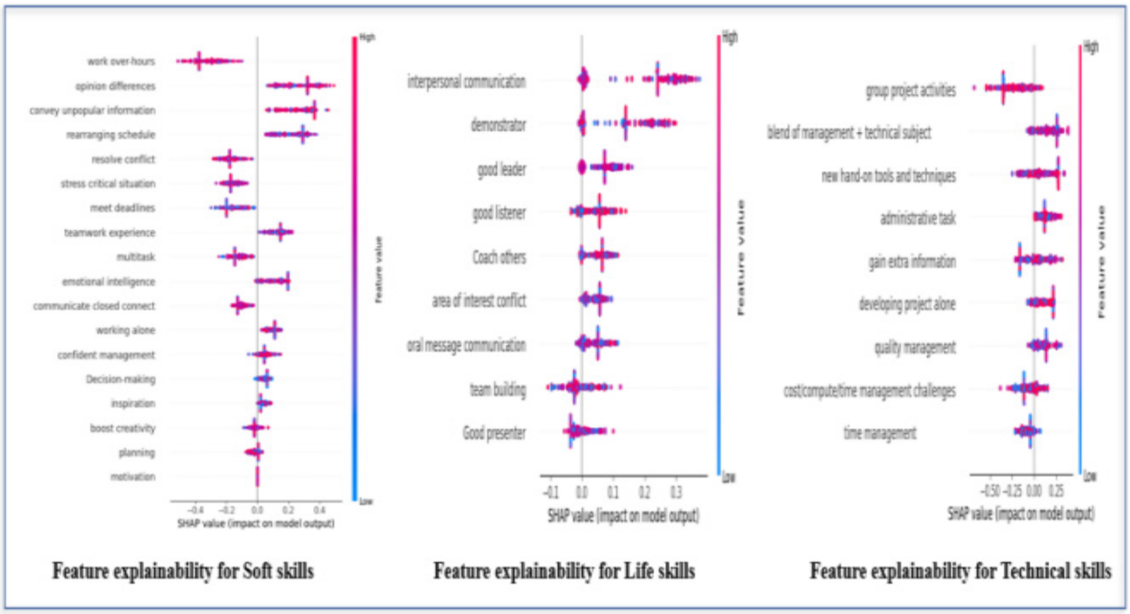

Figure 9. Summary plots for soft, life, and technical skills (Nizar et al., 2024b)

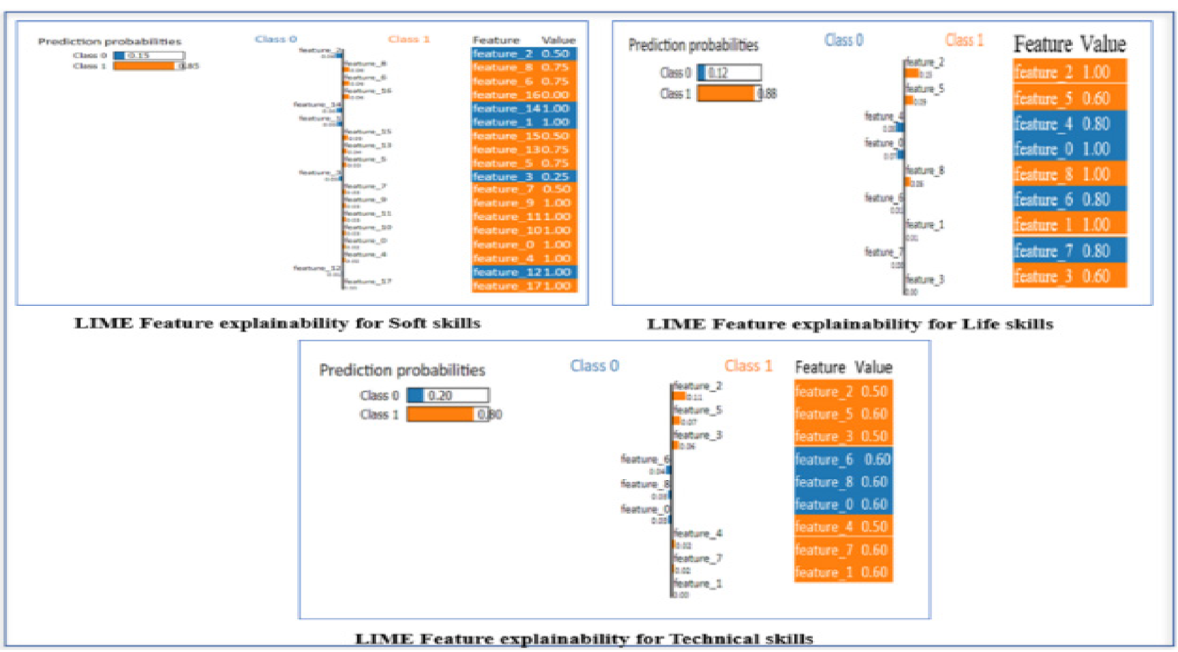

Figure 10. LIME visualization for soft, life, and technical skills (Nizar et al., 2024b)

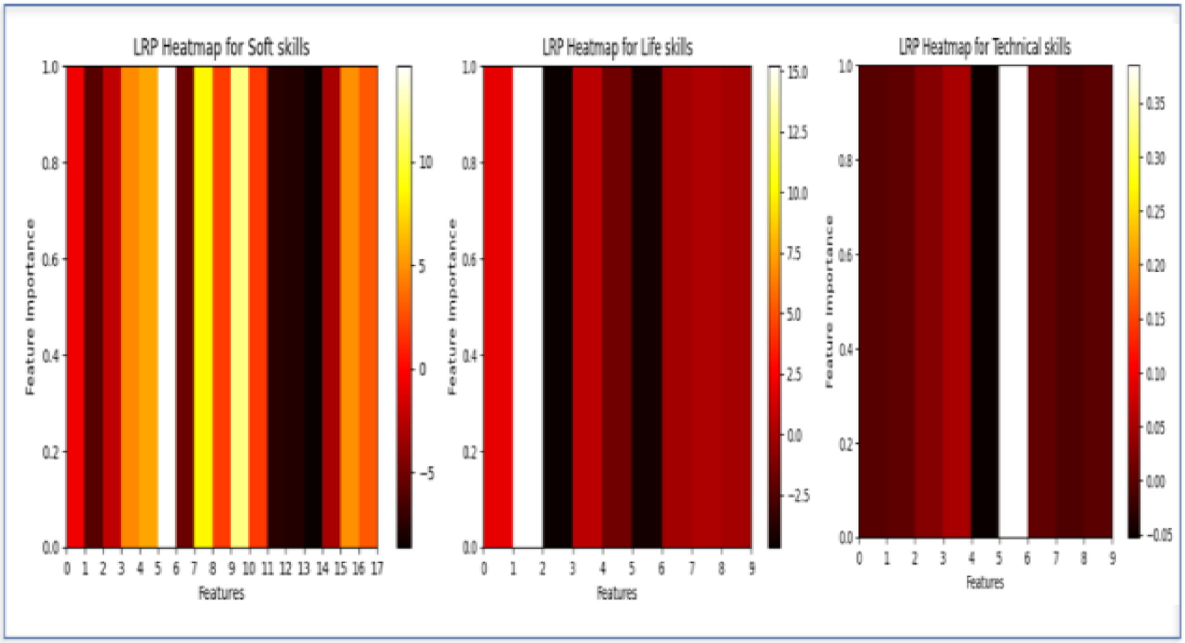

Figure 11. Heatmaps of LRP for soft, life, and technical skills (Nizar et al., 2024b)

The dataset is designed to evaluate, predict, and classify students' strengths and potential areas for development based on the three skill sets. It employs a multiclass classification framework, with class labels ranging from 0 to 7, representing varying levels of proficiency. Detailed criteria used for predicting Soft, Life, and Technical skills are outlined in Table 4. The dataset's diversity ensures its applicability across a range of educational scenarios, making it a robust tool for identifying gaps in students' skill sets. This comprehensive approach supports targeted interventions to enhance student capabilities and better align academic outcomes with industry expectations.

Table 4. Features utilized to predict multi-skillset

Sl no.	Features to predict soft skills
1	Teamwork Experience
2	Confident Management
3	Stress Critical Situations
4	Communicate Closed Connect
5	Multitask
6	Working Alone
7	Boosting Creativity
8	Rearranging Schedules
9	Area of Conflict Interest
10	Good Presenter
11	Oral message communication
12	Demonstrator
13	Team Building
14	Coach Others
15	Blend of Management plus Technical Subject
16	New Hands-On Tools

Sl no.	Features to predict soft skills
17	Time Management
18	Cost or Time Management Challenges
19	Gaining Extra Information

Feature Selection and Correlation Map

By determining which characteristics are most pertinent for prediction, feature selection is essential to enhancing model performance. The direction and intensity of correlations between various features in a dataset are graphically represented by a correlation map (Prestes et al., 2021). The correlations between various skill-related variables are examined in this study using a correlation map, which helps choose the most predictive attributes for multi-skillset analysis. The linkages between the strongest predictive qualities for multi-skillset analysis are graphically depicted in the correlation map for 19 features in Figure 12. Key attributes, including oral communication, managerial confidence, and teamwork experience, are highlighted in this heatmap. Increases in one ability tend to coincide with advances in another, as indicated by strong correlations between related traits, which are displayed in lighter shades. As an illustration of how interwoven they are, the collaboration experience is highly correlated with team-building and management confidence. The dataset's interpretability is improved by removing redundant or weakly linked properties by concentrating on the top 19. Because the most important features are kept for training, this condensed feature set enables predictive models like BERT to attain greater accuracy and generalisability.

A wider view of the connections between soft, life, and technical abilities is provided in Figure 13, which shows the correlation map for 36 features. The darker areas of the heatmap, which display weak or insignificant correlations, suggest that although this thorough depiction captures more complex interdependencies, it also introduces possible noise. Strong correlations between some clusters, like confidence in management and leadership or teamwork and team building, reaffirm their significance in predictive modeling. The requirement for a meticulous feature selection procedure is further supported by the possibility that the dataset's increased complexity with all 36 features may decrease model efficiency. This study maximizes the prediction potential by combining feature selection and correlation analysis to guarantee that only the most significant characteristics are included in the training process, optimizing the predictive power of BERT while reducing computational overhead.

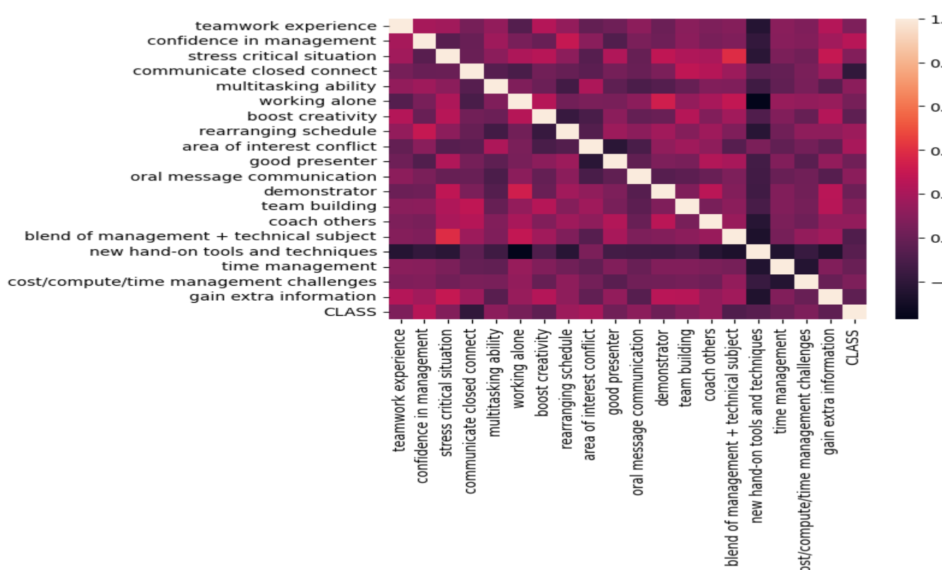

Figure 12. Correlation map of 19 multi-skillset

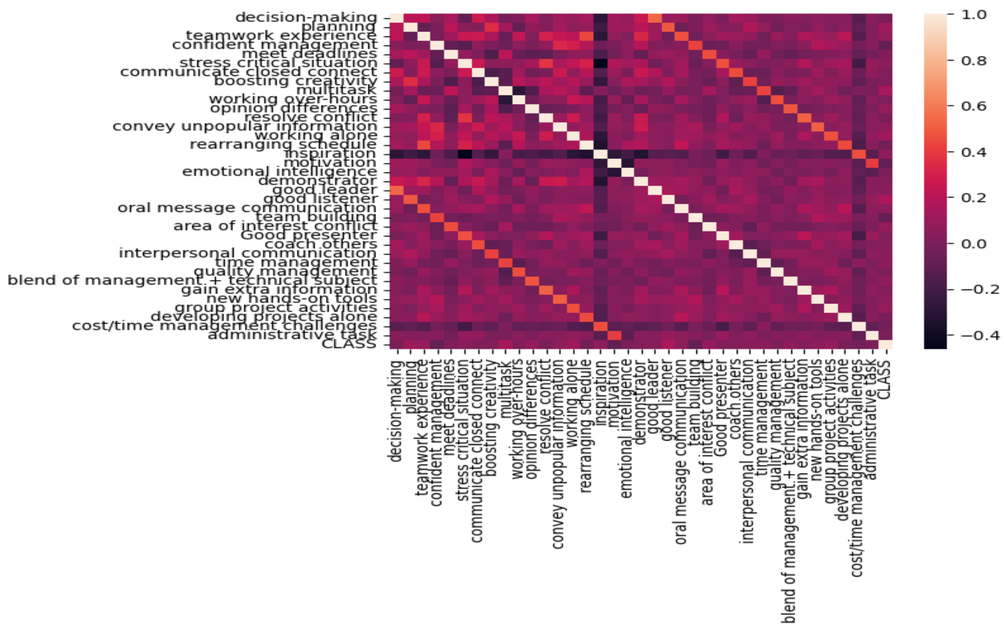

Figure 13. Correlation map of 36 multi-skillset

EXPERIMENTAL ANALYSIS

This study explores the application of BERT for predicting students' skill sets. To evaluate its performance, metrics such as precision, accuracy, recall, and F1-score were used to compare several models, including GRU, LSTM, CNN, RNN, and the Transformer model. The analysis was conducted using a dataset comprising 36 features related to Soft Skills, Life Skills, and Technical Skills. By leveraging Explainable AI techniques, the 19 most predictive features were identified, significantly contributing to the enhancement of the models' performance. A detailed comparison of results was conducted between the full set of 36 features and the top 19 features, covering soft, life, and technical skills. Among the models evaluated, the BERT model with 19 features demonstrated the highest accuracy, achieving 96.25%. This performance surpassed that of other models, including GRU, LSTM, CNN, RNN, and Transformer, across both feature sets. As presented in Table 5, BERT with 19 features exhibited remarkable robustness in predicting multi-skillsets, with consistently high F1-score (0.9621), precision (0.9601), and recall (0.9624). These values underscore their exceptional generalizability and predictive capability.

The findings highlight that BERT's performance significantly benefits from the feature selection process enabled by Explainable AI, which identified the 19 most predictive features. This streamlined feature set allowed BERT to outperform not only its performance with the full 36 features, but also every other model in terms of overall assessment metrics and correctness. These results emphasize BERT's effectiveness in addressing the nuanced requirements of skillset prediction, demonstrating its suitability for analyzing complex datasets in educational contexts.

BERT is the most robust model for both datasets. The superiority of BERT for skills evaluation tasks comes from its ability to identify contextual linkages in textual and structured data. By using bi-directional attention processes, BERT is able to comprehend word and feature dependencies more thoroughly than typical sequential models such as RNN, LSTM, and GRU, which analyze input in a one-directional way. This improves its ability to extract intricate skill relationships that are difficult for other models to capture. The contextual richness and adaptability of BERT's pre-trained language representations are superior to CNN and Transformers, despite the fact that they both increase feature extraction. Figure 11 determines the ROC curves for five different algorithms used in predicting the multi-skillset of computer science students: GRU, LSTM, CNN, RNN, Vanilla Transformer model, and BERT.

Table 5. Comparison of evaluation metrics across various modes for predicting multi-skillsets

Model employed	Features	Accuracy	Precision	Recall	F1-score	AUC value
GRU	36	0.7655	0.7697	0.7655	0.7659	0.9467
LSTM		0.8678	0.8714	0.8678	0.8676	0.9817
CNN		0.8428	0.8554	0.8428	0.8434	0.9809
RNN		0.5885	0.6092	0.5885	0.5901	0.8596
TRANSFORMER		0.9027	0.8712	0.9678	0.9002	0.829
BERT		0.9226	0.8644	0.9374	0.8888	0.8768
GRU	19	0.9002	0.9035	0.9002	0.8997	0.9872
LSTM		0.9027	0.9037	0.9027	0.9006	0.9795
CNN		0.8952	0.8951	0.8952	0.8942	0.9684
RNN		0.8977	0.8997	0.8977	0.8976	0.9437
TRANSFORMER		0.9077	0.9117	0.9077	0.9059	0.9543
BERT		**0.9625**	**0.9601**	**0.9624**	**0.9621**	**0.9813**

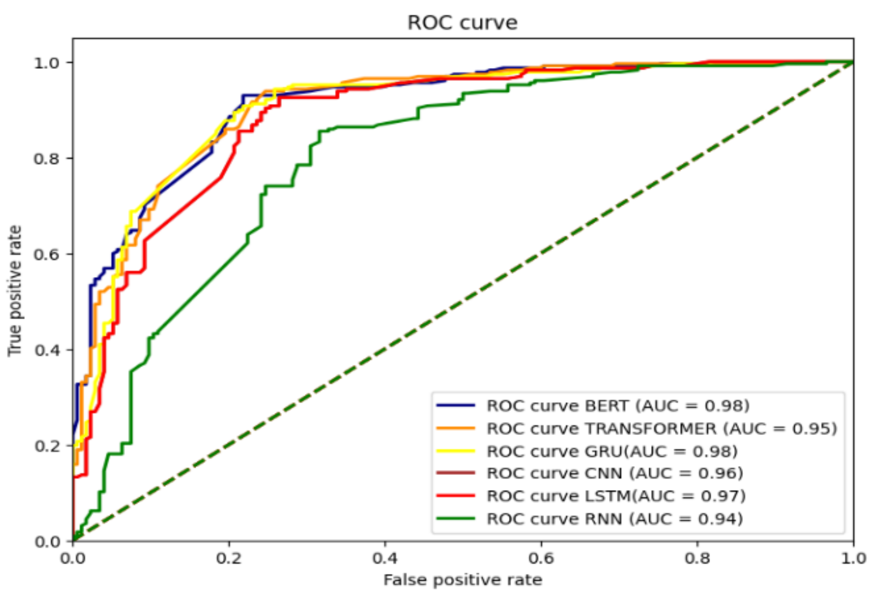

Figure 11. ROC curves for multi-skillset

COMPARISON OF THE PROPOSED MODEL WITH OTHER MODELS

Using a variety of performance criteria, the model's efficacy is carefully assessed, providing a comprehensive review of its overall capabilities. To assess the BERT model thoroughly, this study uses a number of statistical metrics. The BERT model's performance is evaluated by contrasting it with various machine learning models, including the Vanilla Transformer model, CNN, LSTM, GRU, and RNN.

This comparison of several metrics shows how well the BERT model predicts skillsets. Figure 12 illustrates the predictive power of the proposed BERT model, using both 19 and 36 multi-skillsets, by visualizing its performance across a range of evaluation parameters. The graph makes it abundantly evident that the BERT model outperforms all other models, especially when utilizing the 19 most predictive features. This highlights how well the feature selection procedure improved BERT's accuracy and overall performance.

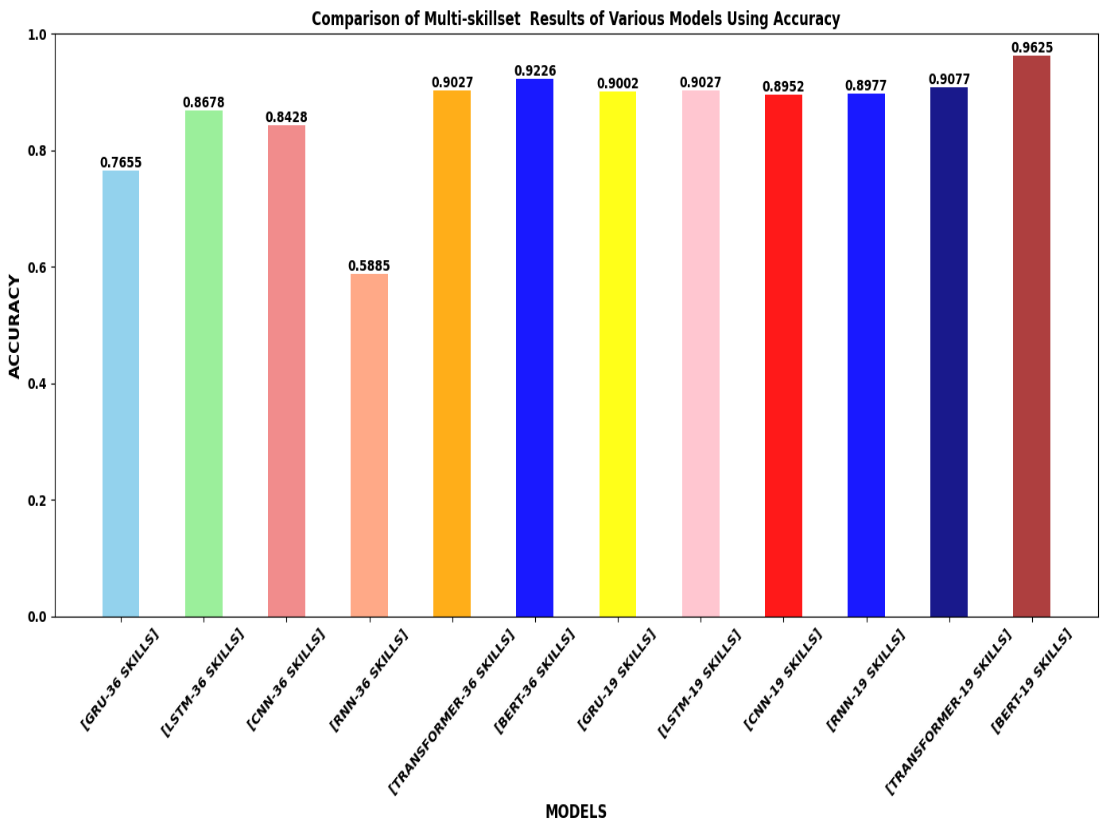

Figure 12. Accuracy values for the prediction of multi-skillsets

CONCLUSION AND FUTURE SCOPE

The study demonstrates the superior capability of BERT's advanced natural language processing techniques in predicting students' multi-skillsets, outperforming traditional and deep learning baselines such as GRU, LSTM, CNN, RNN, and Transformer. By achieving a high accuracy of 96.25% and maintaining strong performance even with limited labeled data, the proposed model proves its suitability for real-world educational environments where data scarcity can be a constraint. The model's ability to identify skill gaps with precision enables educators to design targeted, personalized learning interventions that address the unique needs of each student, thereby fostering both academic growth and career readiness.

From a practical standpoint, educational institutions can incorporate this framework into regular assessment systems to track skill development over time, enabling early identification of weaknesses and timely remedial measures. For instructional designers, the findings provide a data-driven foundation to optimize course content, assessment methods, and teaching strategies so they align more closely with industry requirements. Policymakers can use these insights to prioritize funding for skill-building programs, develop competency-based curricula, and implement performance monitoring systems that improve graduate employability and reduce the industry-academia skill gap.

The broader implication of this research lies in demonstrating how AI-driven analytics can reshape education into a more adaptive, personalized, and outcome-oriented system. However, the study is not without limitations. The dataset used is drawn from a single regional context, which may limit the generalizability of results to other geographic, cultural, or academic settings. The current model focuses on three primary skill domains; expanding this to include cognitive or interdisciplinary skills, could provide a more holistic understanding of student competencies. Another limitation lies in the computational demands of large-scale BERT models, which may pose challenges for resource-constrained institutions.

Future research should validate the proposed framework with diverse datasets from multiple regions and disciplines, explore hybrid architectures such as BERT-GRU or BERT-CNN for enhanced robustness, and integrate advanced Explainable AI techniques to improve model transparency and trustworthiness. By addressing these aspects, the predictive framework can evolve into a scalable, universally applicable tool for data-driven educational policy and instructional design, supporting the global shift toward personalized, skill-oriented learning ecosystems.

DECLARATIONS

Statement on conflicts of interest: The writers claim to have no conflicting interests.

REFERENCES

Akdur, D. (2022). Analysis of software engineering skills gap in the industry. *ACM Transactions on Computing Education*, *23*(1), Article 16. https://doi.org/10.1145/3567837

Al-Ahmad, B., Banitaan, S., & Alkhawaldeh, R. (2020). The impact of software team project measurements on students' performance in software engineering education. *Journal of Education and Practice, 11*(31), 7-17.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1964). *Taxonomy of educational objectives* (Vol. 2). Longmans, Green.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical evaluation of gated recurrent neural networks on sequence modeling*. PsyArXiv. https://doi.org/10.48550/arXiv.1412.3555

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 4171–4186). Association for Computational Linguistics.

Fávero, E. M. D. B., & Casanova, D. (2021). *BERT_SE: A pre-trained language representation model for software engineering*. PsyArXiv. https://doi.org/10.48550/arXiv.2112.00699

Garousi, V., Giray, G., Tüzün, E., Catal, C., & Felderer, M. (2019a). Aligning software engineering education with industrial needs: A meta-analysis. *Journal of Systems and Software*, *156*, 65–83. https://doi.org/10.1016/j.jss.2019.06.044

Garousi, V., Giray, G., Tüzün, E., Catal, C., & Felderer, M. (2019b). Closing the gap between software engineering education and industrial needs. *IEEE Software*, *37*(2), 68–77. https://doi.org/10.1109/MS.2018.2880823

Ghojogh, B., & Ghodsi, A. (2023). *Recurrent neural networks and long short-term memory networks: Tutorial and survey*. PsyArXiv. https://doi.org/10.48550/arXiv.2304.11461

Gnatz, M., Kof, L., Prilmeier, F., & Seifert, T. (2003). A practical approach of teaching Software Engineering. *Proceedings of the 16th Conference on Software Engineering Education and Training* (pp. 120–128). IEEE.

Heckman, J. J., & Kautz, T. (2012). Hard evidence on soft skills. *Labour Economics*, *19*(4), 451–464. https://doi.org/10.1016/j.labeco.2012.05.014

Jaseena, K. U., & Kovoor, B. C. (2021). A wavelet-based hybrid multi-step wind speed forecasting model using LSTM and SVR. *Wind Engineering*, *45*(5), 1123–1144. https://doi.org/10.1177/0309524X20964762

Khan, M. S., Ibrahim, M., & Haider, S. A. (2021, July). *Performance prediction of computer science students in capstone software engineering course through educational data mining.* Paper presented at 2021 ASEE Virtual Annual Conference. https://doi.org/10.18260/1-2--37575

Kirova, V. D., Ku, C. S., Laracy, J. R., & Marlowe, T. J. (2024, March). Software engineering education must adapt and evolve for an LLM environment. *Proceedings of the 55th ACM Technical Symposium on Computer Science Education* (pp. 666–672). Association for Computing Machinery. https://doi.org/10.1145/3626252.3630927

Kumar, P., Batra, S., & Raman, B. (2021). Deep neural network hyper-parameter tuning through twofold genetic approach. *Soft Comput.*, *25*(13), 8747-8771. https://doi.org/10.1007/s00500-021-05770-w

Lizzio, A., & Wilson, K. (2004). Action learning in higher education: An investigation of its potential to develop professional capability. *Studies in Higher Education*, *29*(4), 469–488. https://doi.org/10.1080/0307507042000236371

Mahanti, R., & Mahanti, P. K. (2005). Software engineering education from Indian perspective. *Proceedings of the 18th Conference on Software Engineering Education & Training, Ottawa, ON, Canada,* 111–117. https://doi.org/10.1109/CSEET.2005.30

Mahon, J., Mac Namee, B., & Becker, B. A. (2024). Guidelines for the evolving role of generative AI in introductory programming based on emerging practice. *Proceedings of the 2024 Innovation and Technology in Computer Science Education* (Vol. 1, pp. 10–16). Association for Computing Machinery. https://doi.org/10.1145/3649217.3653602

Makhoba, L., Jadhav, A., Sixhaxa, K., & Ajoodha, R. (2023). Evaluation of student skillsets as predictors of success at higher education institutions. In S. Kumar, S. Hiranwal, S. D. Purohit & M. Prasad (Eds.), *Proceedings of International Conference on Communication and Computational Technologies* (pp. 585–600). Springer. https://doi.org/10.1007/978-981-19-3951-8_45

Marjan, M. A., Uddin, M. P., & Ibn Afjal, M. (2023). An educational data mining system for predicting and enhancing tertiary students' programming skill. *The Computer Journal*, *66*(5), 1083–1101. https://doi.org/10.1093/comjnl/bxab214

Mezhoudi, N., Alghamdi, R., Aljunaid, R., Krichna, G., & Düştegör, D. (2023). Employability prediction: A survey of current approaches, research challenges, and applications. *Journal of Ambient Intelligence and Humanized Computing*, *14*(3), 1489–1505. https://doi.org/10.1007/s12652-021-03276-9

Nizar, J., Sharmila, R., & Jaseena, K. U. (2024a). A random forest model for prediction of software engineering skill set among computer science students through explainable AI. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(21s), 2633–2650.

Nizar, J., Sharmila, R., & Jaseena, K. U. (2024b). Prediction and assessment of software engineering skillset among computer science students using convolutional neural networks through explainable AI. *Journal of Theoretical and Applied Information Technology*, *102*(21), 7948–7968.

Nizar, J., Sharmila, R., & Jaseena, K. U. (2024c). *Software engineering skillset assessment dataset for computer science students.* Mendeley Data. https://doi.org/10.17632/7b68fzgy6b.1

Noveski, G., Jeroncic, M., Velard, T., Kocuvan, P., & Gams, M. (2024). Comparison of large language models in generating machine learning curricula in high schools. *Electronics*, *13*(20), 4109. https://doi.org/10.3390/electronics13204109

Petrovska, O., Pearsall, R., & Clift, L. (2025). Assessing software engineering students' analytical skills in the era of generative AI. *Proceedings of the 9th Conference on Computing Education Practice* (pp. 34–34). Association for Computing Machinery. https://doi.org/10.1145/3702212.3702223

Prestes, P. A. N., Silva, T. E. V., & Barroso, G. C. (2021). Correlation analysis using teaching and learning analytics. *Heliyon*, *7*(11), e08435. https://doi.org/10.1016/j.heliyon.2021.e08435

Radermacher, A., Walia, G., & Knudson, D. (2014). Investigating the skill gap between graduating students and industry expectations. *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 291–300). Association for Computing Machinery. https://doi.org/10.1145/2591062.2591159

Robles, M. M. (2012). Executive perceptions of the top 10 soft skills needed in today's workplace. *Business Communication Quarterly*, *75*(4), 453–465. https://doi.org/10.1177/1080569912460400

Scaffidi, C. (2018). Employers' needs for computer science, information technology and software engineering skills among new graduates. *International Journal of Computer Science, Engineering and Information Technology*, *8*(1), 1–12. https://doi.org/10.5121/ijcseit.2018.8101

Sengul, C., Neykova, R., & Destefanis, G. (2024). Software engineering education in the era of conversational AI: Current trends and future directions. *Frontiers in Artificial Intelligence*, 7, 1436350. https://doi.org/10.3389/frai.2024.1436350

Siddique, A., Butt, G. M., Zahid, A., Naveed, Q. N., & Alouane, M. T.-H. (2024). Analyzing software industry trends to improve curriculum. *IEEE Access, 12,* 22510-22523. https://doi.org/10.1109/ACCESS.2024.3362244

Sunindijo, R. Y. (2015). Project manager skills for improving project performance. *International Journal of Business Performance Management*, *16*(1), 67–83. https://doi.org/10.1504/IJBPM.2015.066041

Sventekova, E., & Lovecek, T. (2012). Project-based teaching practice in the academic environment. *Proceedings of the 11th WSEAS International Conference on Education and Educational Technology*, *Singapore City, Singapore.*

Vaswani, A., Shazeer, N., Parmar, N., Uskoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017, December). Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach*, 6000-6010.

World Health Organization. (1999). *Partners in life skills education: Conclusions from a United Nations inter-agency meeting.* Geneva: Mental Health Promotion.

Younesi, A., Ansari, M., Fazli, M., Ejlali, A., Shafique, M., & Henkel, J. (2024). A comprehensive survey of convolutions in deep learning: Applications, challenges, and future trends. *IEEE Access*, *12*, 41180–41218. https://doi.org/10.1109/ACCESS.2024.3376441

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, *31*(7), 1235–1270. https://doi.org/10.1162/neco_a_01199

Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education – Where are the educators? *International Journal of Educational Technology in Higher Education*, *16*, Article 39. https://doi.org/10.1186/s41239-019-0171-0

Zhu, L., Qiu, D., Ergu, D., Ying, C., & Liu, K. (2019). A study on predicting loan default based on the random forest algorithm. *Procedia Computer Science*, *162*, 503–513. https://doi.org/10.1016/j.procs.2019.12.017

APPENDIX

Table A1. Attributes used for forecasting soft skills

Sl no.	Attributes used for forecasting soft skills
1	Decision-Making
2	Planning
3	Teamwork Experience
4	Confident Management
5	Meeting Deadlines
6	Stress Critical Situations
7	Communicate Closed Connect
8	Boosting Creativity
9	Multitask
10	Working Over Hours
11	Opinion Differences

Sl no.	Attributes used for forecasting soft skills
12	Resolving Conflict
13	Conveying Unpopular Information
14	Working Alone
15	Rearranging Schedules
16	Inspiration
17	Motivation
18	Emotional Intelligence

Table A2. Attributes used for forecasting life skills

Sl no.	Attributes to forecast life skills
1	Demonstrator
2	Good Leader
3	Good Listening
4	Oral message communication
5	Team Building
6	Area Of Conflict Interest
7	Good Presenter
8	Coach Others
9	Interpersonal Communication

Table A3. Attributes used for forecasting technical skills

Sl no.	Attributes to forecast technical skills
1	Time Management
2	Quality Management
3	Blend Of Management Plus Technical Subject
4	Gaining Extra Information
5	New Hands-On Tools
6	Group Project Activities
7	Developing Projects Alone
8	Cost Or Time Management Challenges
9	Administrative Tasks

AUTHORS

Jasmin Nizar received her MCA degree from SCMS School of Technology and Management, Kochi, Kerala, in 2010. She is pursuing a PhD at Karpagam University, Coimbatore, Tamil Nadu, India. Her thesis is related to the software engineering skill set of computer science students. Her research interests include data mining, machine learning, and deep learning.

Dr R Sharmila is Professor and Head of the Department of Computer Applications, Karpagam Academy of Higher Education. She has 20+ years of teaching experience in the field of computer applications. She received her PhD from Manonmaniam Sundaranar University in 2018. Her areas of interest are Data Science, Big Data Analytics, Ad Hoc Networks, and Applications of Machine Learning. She has published three books and four book chapters in Scopus-indexed publications. She has also published more than 20 journal articles in various Scopus/Web of Science journals. She currently has eight research students. She is also a reviewer for reputed journals in computer science.