Neural Computing and Applications (2025) 37:3077–3110
https://doi.org/10.1007/s00521-024-10753-7

ORIGINAL ARTICLE

Advancing mental health detection in texts via multi-task learning with soft-parameter sharing transformers

Dheeraj Kodati[1,2] · Ramakrishnudu Tene[1]

Received: 4 April 2024 / Accepted: 23 October 2024 / Published online: 11 December 2024

Abstract

In recent years, mental health issues have profoundly impacted individuals' well-being, necessitating prompt identification and intervention. Existing approaches grapple with the complex nature of mental health, facing challenges like task interference, limited adaptability, and difficulty in capturing nuanced linguistic expressions indicative of various conditions. In response to these challenges, our research presents three novel models employing multi-task learning (MTL) to understand mental health behaviors comprehensively. These models encompass soft-parameter sharing-based long short-term memory with attention mechanism (SPS-LSTM-AM), SPS-based bidirectional gated neural networks with self-head attention mechanism (SPS-BiGRU-SAM), and SPS-based bidirectional neural network with multi-head attention mechanism (SPS-BNN-MHAM). Our models address diverse tasks, including detecting disorders such as bipolar disorder, insomnia, obsessive-compulsive disorder, and panic in psychiatric texts, alongside classifying suicide or non-suicide-related texts on social media as auxiliary tasks. Emotion detection in suicide notes, covering emotions of abuse, blame, and sorrow, serves as the main task. We observe significant performance enhancement in the primary task by incorporating auxiliary tasks. Advanced encoder-building techniques, including auto-regressive-based permutation and enhanced permutation language modeling, are recommended for effectively capturing mental health contexts' subtleties, semantic nuances, and syntactic structures. We present the shared feature extractor called shared auto-regressive for language modeling (S-ARLM) to capture high-level representations that are useful across tasks. Additionally, we recommend soft-parameter sharing (SPS) subtypes-fully sharing, partial sharing, and independent layer-to minimize tight coupling and enhance adaptability. Our models exhibit outstanding performance across various datasets, achieving accuracies of 96.9%, 97.4%, and 98%, and F1 scores of 93.8%, 94%, and 94.6% for distinct mental health-related datasets, respectively. We conduct a thorough comparative analysis to evaluate the models' applicability and effectiveness across diverse contexts and platforms, supported by ablation tests highlighting essential components and confirming their superiority over state-of-the-art models in the MTL context.

Keywords Mental health detection · Multi-task learning · Soft-parameter sharing · Shared feature extractor · Deep learning · Transformers

1 Introduction

Mental health encompasses a multifaceted realm, intertwining emotional, psychological, and social well-being, which necessitates nuanced approaches for effective understanding and intervention. Amidst the persistent global burden of mental health challenges, the integration of multi-task learning (MTL) becomes imperative. This integration involves various tasks such as disorder detection, suicide and non-suicide classification, and emotion detection, all contributing to a comprehensive mental health framework. Mental health extends beyond the spectrum of disorders, encompassing a diverse range of emotional experiences, coping mechanisms, and resilience in the face of life's challenges. From mood disorders to considerations of suicidal tendencies, mental health manifests as a dynamic and complex interplay, requiring flexible and responsive methodologies. MTL facilitates a holistic understanding of mental health by concurrently addressing

✉ Dheeraj Kodati
dheerajkodaticse@student.nitw.ac.in

Ramakrishnudu Tene
trk@nitw.ac.in

[1] Department of Computer Science and Engineering, National Institute of Technology, Warangal, India

[2] Department of Computer Science and Engineering, Mahindra University, Hyderabad, India

specific disorders like bipolar disorder, classifying suicide risk, and detecting emotional states. This comprehensive approach ensures a nuanced perspective that transcends singular diagnostic labels, supporting a personalized approach to mental health care. By simultaneously assessing disorders, suicidal tendencies, and emotional states, interventions can be tailored to individual needs, acknowledging the unique spectrum of mental health challenges. The integration of suicide classification with disorder detection aims for early intervention and risk mitigation. Identifying potential indicators of suicidal tendencies within the broader mental health context enables proactive measures to prevent crises. MTL recognizes the significant role of emotions in mental health and helps us understand the complex relationships between emotions, disorders, and suicide risks.

Most existing approaches rely on single tasks to comprehend individual behavior, limiting their ability to capture the complexity of mental health conditions [1–4]. This limitation arises because tasks with distinct data distributions or structures may not effectively utilize the same set of shared parameters, resulting in less-than-optimal performance. Our research emphasizes the adoption of SPS. In this approach, features are shared from a designated input vector, and the entire layer is shared when necessary, while the remaining layers are kept as independent as possible. The objective is to concurrently perform interconnected tasks that share vital contextual features. To extract these features, we introduce various subtypes such as fully sharing (FS), partial sharing (PS), and independent layer (IL) to uphold loose coupling to the greatest extent possible. We present various strategies for addressing different tasks. The first strategy incorporates a comprehensive attention mechanism. The following approach utilizes a partial sharing attention mechanism, specifically integrating a self-attention component.

The third strategy combines a partial sharing multi-head attention (MHA) component with a bidirectional long short-term memory (BiLSTM). Auto-regressive-based permutation and enhanced permutation language modeling techniques are suggested to capture contextual word embeddings and sequential dependencies effectively. The approach involves predicting the next token in a sequence based on preceding tokens, considering the interdependence of all prior tokens. This encompasses the exploration of all possible permutations of tokens within a sequence. A fully shared embedding representation is employed to construct the encoder. Following an auto-regressive approach, attention is allocated to each input position. The objective is to uphold long-term dependencies within transformers. In addition, most research works focused on primary or basic emotions such as anger, anxiety, and sadness [5, 6]. However, our research focuses on in-depth emotion mining, such as secondary and tertiary emotion detection ("Parrott's emotion classification, 2001"), which identifies further emotions instead of primary emotions in mental health-related texts.

1.1 Motivation and contributions of the research

The work in [7] introduced a BERT-based model designed for the primary task of emotion detection. Additionally, a deep MTL framework is presented in [8], addressing both sentiment and emotion analysis simultaneously. Similarly, the study in [9] suggested the utilization of a transformer model for MTL. While these prior works offer valuable insights, they exhibit limitations regarding emotion detection within textual sequences, do not emphasize the preservation of long-term dependencies, and primarily focus on auxiliary tasks. However, these prior contributions have served as motivation for our main contributions, which are as follows:

- Early mental health detection is crucial for understanding individuals' psychological well-being. Prior studies focused on single tasks, limiting comprehensive assessment. Our approach includes disorder detection, categorizing suicide or non-suicide texts, and the main task is emotion detection in suicide notes.
- In our study, we present various task-performing methodologies. The initial methodology incorporates an LSTM-based transformer with complete attention. The second methodology utilizes a bidirectional gated neural network with partial sharing attention, encompassing self-attention. The third methodology integrates a MHA component with a bidirectional neural network.
- We use fully shared embedding representation for encoder construction, following an auto-regressive approach to attend to each input position and maintain long-term dependencies with transformers.
- We recommend incorporating shared auto-regressive language modeling along with enhanced permutation language modeling for effective contextual word embeddings and sequential dependencies.
- The proposed models address long-term dependencies in texts using various strategies. These include LSTM with attention for sequence dependencies, BiGRU with self-head attention for bidirectional context, and bidirectional networks with MHA for diverse text focus. Additionally, shared auto-regressive (S-ARLM) language modeling enhances high-level representation and understanding of long-term context.

Figure 1 illustrates the proposed overall MTL framework incorporating soft-parameter sharing (SPS) for

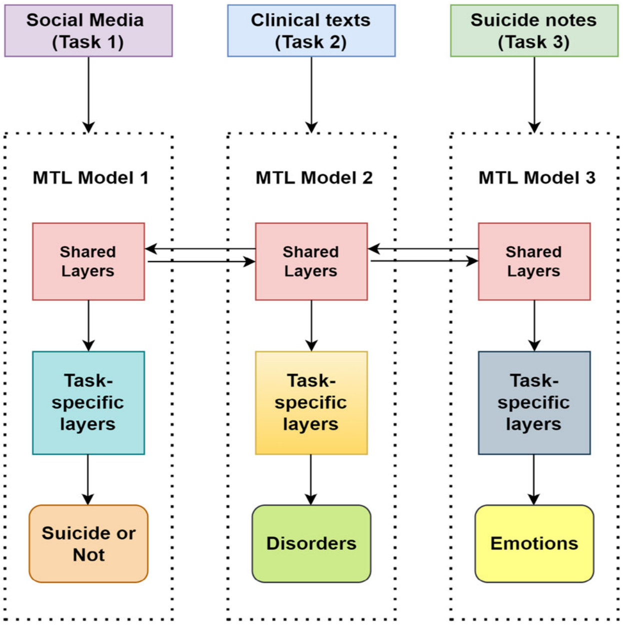


Fig. 1 Proposed framework for MTL with SPS for mental health detection

mental health detection. In Figs. 1 and 2, the models are defined as follows: model 1 is SPS-BiGRU-SAM, model 2 is SPS-LSTM-AM, and model 3 is SPS-BNN-MHAM. These models process three distinct datasets with different input texts. The SPS mechanism facilitates joint learning by sharing layers across models and datasets, thereby enhancing performance across diverse tasks. The different proposed approaches, like LSTM-based transformers with complete attention, are adept at capturing sequential patterns and long-term dependencies in textual data. It focuses on understanding contextual information crucial for identifying mental health-related nuances. The bidirectional gated neural network with partial sharing attention excels in discerning nuanced patterns in texts. It allows the model to weigh different parts of the input text, enabling a nuanced understanding of semantic information. The MHA component with a bidirectional neural network captures sequential relationships and contextual features in textual data. It addresses intricate expressions and relationships within the text associated with mental health.

The remaining part of the paper is divided as follows: Sect. 2 is about the literature review, Sect. 3 is related to the suggested framework, and Sect. 4 is about proposed methodologies. Section 5 is related to dataset collection and experimental results. Section 6 is related to the conclusion and future scope.

2 Literature survey

Initially, delving into prior research methodologies in text mining for mental health-related tasks, the utilization of deep learning within the framework of MTL is examined. This analysis aims to identify gaps in existing research, leading to the proposal of new perspectives and considerations for further exploration.

2.1 Mental health-related text mining

The work in [10] focused on extracting lexical features from suicide notes for emotion detection, aiming to enhance understanding of suicidal tendencies. This approach was further refined using principal component analysis to manage feature complexity and improve detection accuracy [11]. However, it relied solely on lexical features, which might miss contextual nuances. Similarly, [12] employed deep learning techniques for classifying suicide-related texts, integrating advanced neural networks to improve accuracy. Despite this, the method struggled with the variability in data and less scope in capturing semantic dependencies. To address these challenges, [13, 14] explored the Word2vec with BiLSTM model to analyze user behavior from social media, focusing on capturing semantic dependencies. Nevertheless, this approach faced limitations in handling the variability of user-generated content. Additionally, [15] utilized a deep learning model for detecting human behavior in online texts, expanding the focus on behavioral cues. However, its narrow scope limited its ability to fully address the complexities of diverse online interactions.

In the study [16], a convolutional neural network (CNN) was employed to detect suicidal threats among teenagers, demonstrating the utility of convolutional networks in identifying key indicators of mental distress. However, this approach had limited capacity for capturing nuanced contexts in longer texts. Similarly, the work in [17] used an RNN-based transformer model with GloVe embeddings for emotion mining from social media, but this model also struggled with capturing relevant contextual dependencies. A skip-gram model with LSTM was applied in [18] to detect emotions in patients' comments. The main limitation of this model was its difficulty in accurately capturing long-range dependencies and contextual features. BiLSTM with an attention mechanism and XGBoost models were used to identify emotions in social media texts [19]. Despite being designed to handle long-range dependencies, these models faced challenges in effectively managing highly unstructured data and maintaining performance consistency across diverse social media texts, particularly in capturing contextual features.

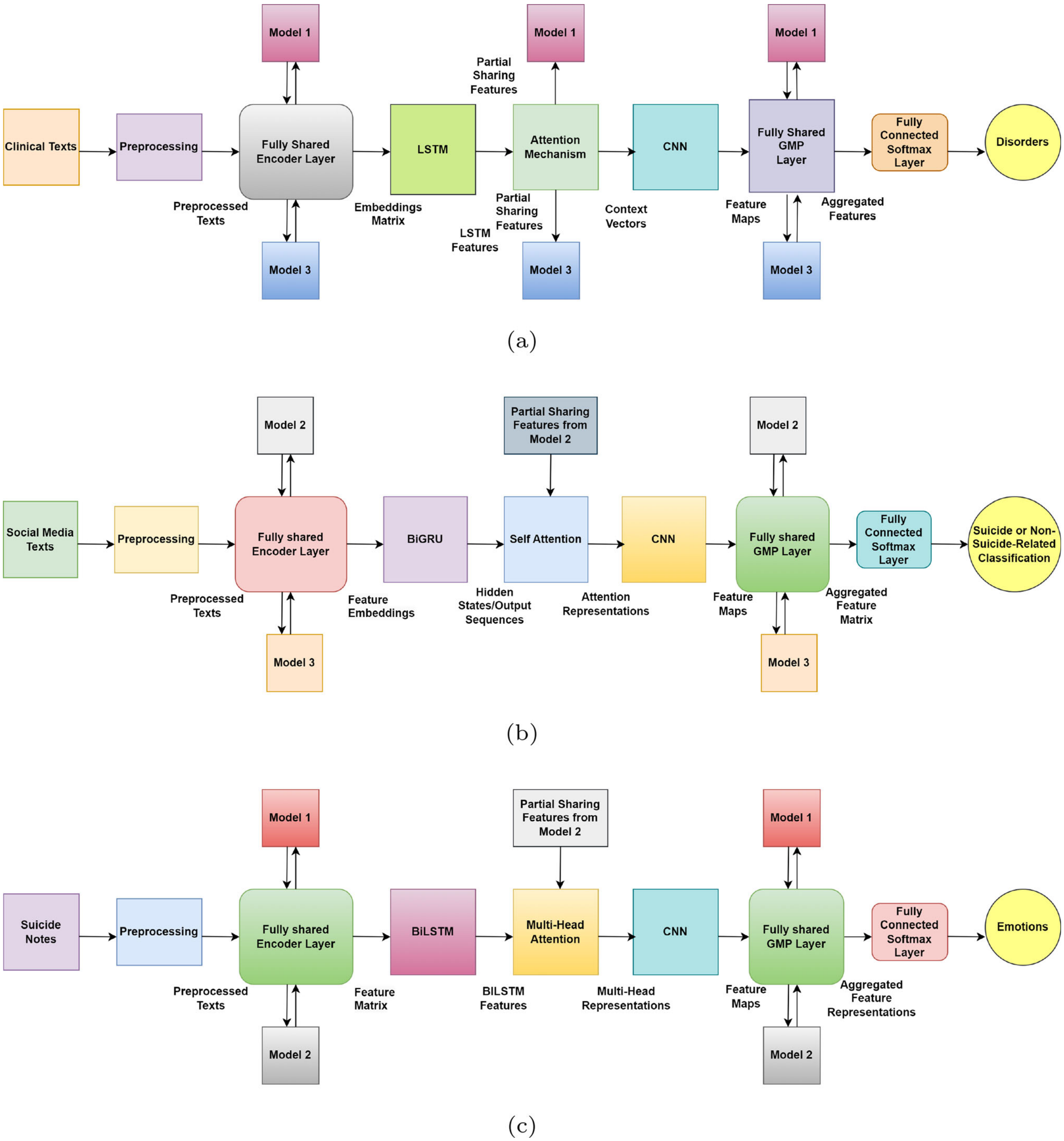


Fig. 2 SPS-based MTL network architecture **a** SPS-LSTM-AM model **b** SPS-BiGRU-SAM model **c** SPS-BiGRU-SAM model

The research in [20] examined Twitter tweets to identify neutral and negative emotions but faced limitations by excluding contextual nuances, restricting its ability to capture a full range of emotional expressions. In contrast, [21] and [22] utilized transfer learning with BERT [23], demonstrating the effectiveness of pre-trained transformers in extracting contextual meanings from text. However, these methods struggled to maintain meaningful context when texts were shortened or divided, potentially losing crucial information. Similarly, [24] recommended the Hierarchical Bi-CuDNNLSTM model for emotion detection in social media, excelling in capturing semantic features but encountering difficulties with long-range dependencies. This issue was also relevant in the Sentiment Information CNN model proposed by [25], effective in sentiment categorization but requiring further refinement to capture nuanced emotions beyond basic positive, negative, or neutral classifications. Additionally, [26] employed a method combining word embeddings, specialized lexicons, and psycholinguistic techniques to enhance classification accuracy in mental health-related content. Nonetheless, this approach's efficacy diminished with longer text sequences, as it was mainly tested on smaller datasets, potentially limiting its scalability and generalizability.

The work in [27] employed TextCNN and BiLSTM models to extract word vector information and contextual features, noting that TextCNN did not inherently account for word order, which is crucial for semantic relationships. The study in [28] utilized an RNN combined with VADER sentiment analysis and TextBlob to detect emotional tones, highlighting challenges in managing ambiguous sentences where context heavily influenced sentiment interpretation. The EmoDNN framework introduced in [29] addressed cognitive ability prediction but stressed the importance of aligning predictions from multiple models, especially with variable-length inputs, as misalignment could lead to inconsistencies and impact ensemble performance.

2.2 Related works on multi-task learning

The work in [30] explored adversarial MTL with neural networks to balance task-specific and invariant feature extraction, a concept also present in [31], which focused on non-redundant training for emotion identification and cause analysis. Both approaches highlighted the need to isolate shared and task-specific features but faced challenges in preventing negative transfer and maintaining task-specific accuracy. Integrating multiple emotion categories into a cohesive model remained a significant challenge. Although multitasking frameworks improved overall accuracy, unifying diverse emotional expressions within the model proved difficult. This issue was exacerbated by the complexity of balancing task-specific and shared features, leading to potential inconsistencies in capturing the full range of emotional states. Additionally, [32] addressed sentiment and emotion detection through a multitasking system with a deep ensemble approach, emphasizing the benefits of multitasking frameworks but not fully resolving the challenge of integrating multiple emotion categories into a unified model.

This work [33] utilized bidirectional language models in MTL for text classification, emphasizing the role of language modeling in capturing nuanced features-a limitation also noted in [34], where integrating features from auxiliary tasks into the primary sentiment analysis task remained complex. The model introduced in [8] was designed for simultaneous sentiment and emotion detection. However, it struggled with capturing contextual relationships between words, which is crucial for understanding nuanced emotions. Similarly, [35] explored multimodal emotion recognition using images, texts, and tags. Although it demonstrated potential for multimodal integration, its applicability was limited when focusing solely on textual data, underscoring the challenges of accurately detecting emotions in text alone. The fusion of pre-trained models with MTL, as shown in [36] and [37], aimed to enhance performance across tasks through domain-specific fine-tuning but often faced issues with generalizability beyond specialized domains.

The work in [38] enhanced emotion recognition in conversations by incorporating speaker identification, emphasizing the importance of contextual and speaker-related information. However, this approach struggled with conversations lacking clear speaker cues. Similarly, [39] explored deep neural networks and hybrid representations for MTL, focusing on shared and task-specific features but facing challenges with feature selection and redundancy. In [9], features were selected based on embedding representations and hidden layer values from a transformer's encoder. However, the study emphasized auxiliary tasks, overlooking the need for capturing bidirectional context within text sequences. The research in [40] examined sentiment analysis within MTL using BERT methods, particularly for aspect-based opinion mining, but required enhancements for handling nuanced and context-dependent sentiment expressions. The model suggested in [7] focused on identifying emotions as the primary task but had limited applicability in detecting nuanced dependences in longer text sequences, similar to issues faced by earlier BERT methods. Likewise, [41] developed an MTL framework for personality traits and emotion detection, also struggling with long text data, a recurring issue in handling extended sequences. Further models, such as MT-Text GCN [42] and a multi-task hierarchical approach [43], addressed complex tasks like hate speech detection and topic analysis but encountered difficulties in maintaining relevant contextual dependencies.

The work in [44] focused on multi-label emotion classification, leveraging emotion descriptors to capture correlations among different emotions. A similar concept was explored in the category-aware CatVRNN model by [45], which generated diverse text outputs. In [46], the MTL-based ESD-ERC model was introduced for emotion recognition in conversations, including an auxiliary task of emotion shift detection. This model utilized context-based attention features, but performance could have been improved by partially sharing these features across tasks to better leverage shared contextual insights. The objective of maintaining loose coupling in MTL frameworks was crucial, as it allowed for concurrent execution of interconnected tasks without excessive interdependence. The work in [47] addressed hate speech and offensive language detection with a multi-head MTL model based on BERT. While it focused on classification tasks, including emotion detection, it did not fully explore the nuances of long text data, revealing a gap in the field. Similarly, [48] explored an MTL approach for identifying emotions and sarcasm detection but placed less emphasis on extracting semantic information vital for both auxiliary and main tasks. These models faced challenges with category coherence and context-dependent tasks. In [49], inaccurate sentiment analysis due to sarcasm was addressed through deep learning methods, suggesting an MTL framework for simultaneous sentiment analysis and sarcasm detection. However, this approach struggled with accurately detecting subtle sarcasm that heavily relies on contextual understanding.

The Text Guided MTL Network aimed to enhance semantic information in non-text modalities [50]. Studies like [51] and [52] explored dialogue act recognition and sentiment classification, showing significant advancements. Integrating non-textual modalities posed challenges due to modality alignment issues. The study in [53] employed an end-to-end MTL framework to identify politeness and emotions in dialogues, particularly in mental health and legal aid contexts. This approach, while comprehensive, struggled with domain-specific language nuances. The research in [54] introduced a multiple-attention technique for precise examination of individual word contributions to different emotions. Accurate emotion detection often required appropriate contexts, which were challenging to obtain. In work [55] used a multi-task support vector machine in a semi-supervised learning framework, and in work [56] proposed the M2Seq2Seq model with attention mechanisms for multimodal MTL. These models highlighted MTL's potential to handle diverse tasks but faced issues with maintaining consistency across tasks. The study in [57] explored MTL by integrating AdapterFusion with language adapters in multilingual contexts, addressing various classification tasks but encountering challenges in balancing cross-lingual features. The work in [58] demonstrated the challenges and advancements in comic emotion analysis within an MTL framework. Collectively, these studies highlighted the need for improved cross-task feature alignment and consistency in MTL, illustrating interconnected challenges and advancements across various MTL applications and refining feature extraction and task-specific adaptability.

2.3 Approaches to address research gaps

The proposed methodology addresses the limitations of existing MTL approaches by leveraging SPS-based techniques to better capture and utilize shared and task-specific features. Unlike prior models that struggled with negative transfer and maintaining task-specific accuracy, our approach uses SPS to balance the shared and independent layers, enhancing adaptability and reducing interference between tasks. By incorporating auto-regressive-based permutation and enhanced permutation language modeling, our models are adept at capturing nuanced linguistic expressions and long-range dependencies, which are often missed by conventional methods. The proposed shared auto-regressive-based transformer method facilitates high-level representation extraction, ensuring that crucial contextual and semantic features are effectively shared across tasks. This is particularly beneficial in tasks like emotion detection in diverse mental health contexts, where the models must discern subtle emotional cues and handle complex, context-dependent language. Additionally, our approach to partially sharing attention mechanisms allows for fine-tuning the focus on relevant features, thus overcoming limitations related to the fusion of features from auxiliary tasks and enhancing the models' generalizability and scalability.

In addition, the proposed approach focusing on relevant parts of the input texts, enabling the model to capture intricate patterns and relationships within the sequences. This is achieved through enhancements that specifically target features relevant to textual emotion recognition and incorporate novel models designed to capture contextual information and long-range dependencies. The methodology captures bidirectional context, allowing the model to understand words in relation to both preceding and succeeding words, which is crucial for a comprehensive understanding of disorders and emotions in sequential texts. Our method also emphasizes the active integration of auxiliary tasks to enhance the main task's accuracy rather than relying solely on these auxiliary tasks. By using BiLSTM and MHA, our approach processes sequences bidirectionally, facilitating the comprehension of contextual information across longer text segments. This strategy helps capture relationships between all words in a sequence, thus improving the model's capacity to handle longer texts and mitigate potential scalability and generalizability limitations. Additionally, the approach captures semantic dependencies and context within text sequences, aiding in emotion detection.

By simultaneously focusing on different parts of the text, our models efficiently capture intricate relationships and dependencies. The use of an auto-regressive process preserves bidirectional contexts and transformer-based deep learning methods capture global contextual features. Auxiliary tasks further enhance the model's performance on the main task. The proposed models offer several advantages over existing methodologies, including capturing context from both directions, which improves understanding of text dependencies and generates highly informative word representations. These models preserve sequential dependencies, which are crucial for nuanced feature comprehension, and adapt seamlessly to variable-length texts. They combine information from both forward and backward directions, providing a holistic context for superior feature extraction and task performance. The models' components are adept at capturing global context understanding, allowing for a focused analysis of specific text parts, leading to more informative text representations. Overall, the proposed models significantly enhance context understanding and nuanced emotion analysis compared to traditional methods. The overall strategies ensures robust performance of the SPS-based MTL across varied datasets and complex mental health-related tasks, outperforming state-of-the-art models in MTL contexts.

3 Suggested framework

Initially, the focus is on the problem statement, followed by a discussion of the proposed techniques for parameter sharing.

3.1 Problem statement

Given a set of tasks $T = \{T_1, T_2, T_3\}$, where X_{T_i} is the input dataset corresponding to each task $T_i \in T$. In the proposed MTL approach, SPS learns a set of functions $\{f_1, f_2, f_3\}$ such that $f_{T_i} : X_{T_i} \rightarrow Y_{T_i}$, where $X_{T_i} \in \{X_{T_1}, X_{T_2}, X_{T_3}\}$, and Y_{T_i} is the vector of class labels corresponding to task T_i. No two tasks share the same input training dataset, such that $X_{T_i} \neq X_{T_j}$ for all $i, j \in \{1, 2, 3\}$. Let X_{mt} represent the main task, and all other tasks X_r are termed auxiliary tasks.

In this paper, perform different tasks, such as disorder detection in patient–doctor conversation-related psychiatric texts. Then, categorize suicide or non-suicide-related texts using Reddit data. The above tasks are considered auxiliary tasks. However, the main task contains emotion mining from suicide notes. This paper suggests FS, PS, and IL subtypes of sharing to preserve SPS rules. If an entire layer L_{T_i} is shared by all tasks in T, then $L_{T_i} \in \{T_1, T_2, T_3\}$, which is known as the fully shared subtype. If a particular layer L_1 belongs to task T_1 and partially shares the input from task T_2 such that input is shared with both T_1 and T_2, this is called partial sharing. However, when $T_1 \neq T_2 \neq T_3$ without sharing inputs X_{T_i}, it is called an independent layer.

3.2 Proposed techniques for parameter sharing

In MTL, we aim to jointly optimize a set of tasks $T = \{T_1, T_2, T_3\}$ with their respective datasets X_{T_i} and labels Y_i. Let Θ represent the shared parameters of the model across all tasks. For a given task T_i, the task-specific parameters are denoted as Θ_i. The objective is to minimize the total loss $\mathcal{L}$ across all tasks, which is a combination of individual task losses. The total loss $\mathcal{L}$ can be expressed as follows:

$$\mathcal{L} = \sum_{i=1}^{N} L_i(\Theta, \Theta_i, X_{T_i}, Y_i) \tag{1}$$

where L_i represents the loss function for task T_i with parameters Θ and Θ_i, and N is the total number of tasks.

3.2.1 Dynamic parameter adaptation

In this approach, we dynamically adapt the shared parameters Θ based on the gradients of individual task losses. The update rule for the shared parameters Θ can be defined as follows:

$$\Theta^{(t+1)} = \Theta^{(t)} - \eta \sum_{i=1}^{N} \nabla_{\Theta} L_i(\Theta, \Theta_i, X_{T_i}, Y_i) \tag{2}$$

Here, $\Theta^{(t)}$ is the current value of the shared parameters at iteration t and $\Theta^{(t+1)}$ denotes the value of the shared parameters after the update at iteration $t + 1$. N is used to denote the total number of tasks over which the gradients are summed, ensuring that the update rule for the shared parameters reflects the combined information from all tasks. η is the learning rate, t is the iteration index, and $\nabla_{\Theta} L_i$ is the gradient of the loss function for task T_i with respect to the shared parameters Θ.

3.2.2 Task-specific adaptation

For task-specific adaptation, the parameters Θ_i are updated independently for each task. The update rule for task-specific parameters Θ_i can be expressed as follows:

$$\Theta_i^{(t+1)} = \Theta_i^{(t)} - \eta \nabla_{\Theta_i} \mathcal{L}_i(\Theta, \Theta_i, X_{T_i}, Y_i) \tag{3}$$

where $\nabla_{\Theta_i} \mathcal{L}_i$ is the gradient of the loss function for task T_i with respect to the task-specific parameters Θ_i.

3.3 Subtypes in soft-parameter sharing

In SPS, the subtypes we employ, namely fully shared, partial sharing, and individual layer adaptation, offer distinct advantages over existing methods in MTL. Fully shared parameter sharing allows maximum parameter sharing across tasks, promoting the learning of a unified representation while reducing the risk of overfitting and effectively leveraging the commonalities between tasks. Partial parameter sharing strikes a balance between shared and task-specific parameters, offering flexibility to adjust to different levels of task interdependence. This approach enables the model to capture both shared and task-specific information, leading to improved generalization performance across multiple tasks. In individual layer adaptation, each layer of the model remains as distinct as possible without sharing knowledge across tasks. This approach ensures that each layer learns task-specific features independently, thereby preserving the unique characteristics of each task. By maintaining the individuality of layers, the model can effectively capture task-specific nuances without being influenced by unrelated information from other tasks. Overall, these subtypes of SPS facilitate efficient knowledge transfer across tasks and maintain task-specific representations for enhanced performance and robustness in MTL scenarios.

3.3.1 Subtype 1: Fully shared parameters

In fully shared parameter sharing, all parameters of the model are shared across tasks. Let Θ represent the shared parameters. The total loss function $\mathcal{L}$ is optimized jointly for all tasks, and the update rule for shared parameters Θ encompasses gradients from all tasks. The optimization objective can be denoted as follows:

$$\min_{\Theta} \sum_{i=1}^{N} \mathcal{L}_i(\Theta, X_{T_i}, Y_i) \tag{4}$$

where min is used because the goal is to minimize the total loss function by adjusting the shared parameters Θ. This process aims to enhance the overall performance of the model across all tasks by reducing the combined loss.

3.3.2 Subtype 2: Partial shared parameters

Partial shared parameter sharing involves a combination of shared and task-specific parameters. Let Θ represent the shared parameters and Θ_i represent the task-specific parameters for task T_i. Some parameters are shared across tasks, while others are specific to each task. The optimization objective can be expressed as follows:

$$\min_{\Theta,\Theta_1,\ldots,\Theta_N} \sum_{i=1}^{N} \mathcal{L}_i(\Theta, \Theta_i, X_{T_i}, Y_i) \tag{5}$$

where $\mathcal{L}_i$ represents the loss function for task T_i with shared parameters Θ and task-specific parameters Θ_i.

3.3.3 Subtype 3: Individual layer adaptation

In this subtype, specific layers of the model are shared across tasks, while others remain task-specific. L is the total number of layers, and l is an index used to reference each layer individually within that total. Let Θ_l represent the parameters of layer l in the model, and $\Theta_l^{(i)}$ represent the task-specific parameters for layer l in task T_i. For example, lower layers responsible for capturing basic features may be shared, while higher layers responsible for task-specific representations may be task-specific. The optimization objective can be denoted as follows:

$$\min_{\Theta,\{\Theta_l^{(i)}|l=1,\ldots,L;\, i=1,\ldots,N\}} \sum_{i=1}^{N} \mathcal{L}_i(\Theta, \{\Theta_l^{(i)} \mid l = 1,\ldots,L\}, X_{T_i}, Y_i) \tag{6}$$

where $\mathcal{L}_i$ represents the loss function for task T_i with shared parameters Θ and task-specific parameters $\Theta_1^{(i)}, \ldots, \Theta_L^{(i)}$, and L is the total number of layers in the model.

$$\Theta_l^{(i)} = \begin{cases} \Theta_l & \text{if } l \text{ is shared} \\ \text{task-specific parameters} & \text{otherwise} \end{cases} \tag{7}$$

Equation 7 denotes how parameters are assigned across layers. It specifies that $\Theta_l^{(i)}$ is either the shared parameter Θ_l or task-specific parameters, depending on the layer's role. This ensures the model correctly applies shared parameters across tasks and uses task-specific parameters where appropriate.

4 Our methodology

Suppose all tasks share the same structure, prone to optimization conflicts. Our research suggests three new models for three different tasks to keep individuality between the resources and follow the rule of MTL. SPS for MTL is categorized into IL, FS, and PS subtypes for maintaining the contextual features. We propose different optimization algorithms such as the SPS-based LSTM with attention mechanism (SPS-LSTM-AM) model for disorder detection in clinical texts, SPS-based bidirectional gated neural networks with self-head attention mechanism (SPS-BiGRU-SAM) model for suicide or non-suicide-related classification using social media data, and SPS-based bidirectional neural network with multi-head attention mechanism (SPS-BNN-MHAM) model for emotion detection in suicide notes. The SPS-BiGRU-SAM model is developed to understand intricate patterns in text, essential for identifying suicide-related features in textual content. The SPS-LSTM-AM model focuses on capturing sequential patterns, making it ideal for disorder detection that relies on contextual understanding. The SPS-BNN-MHAM model is proposed to interpret complex relationships and expressions found in emotionally charged text. The suggested models share common auto-regressive-based encoder representations. The above approach focuses on each input position and correlates with other inputs. Figure 3 shows the suggested approach architecture for SPS. In this study,

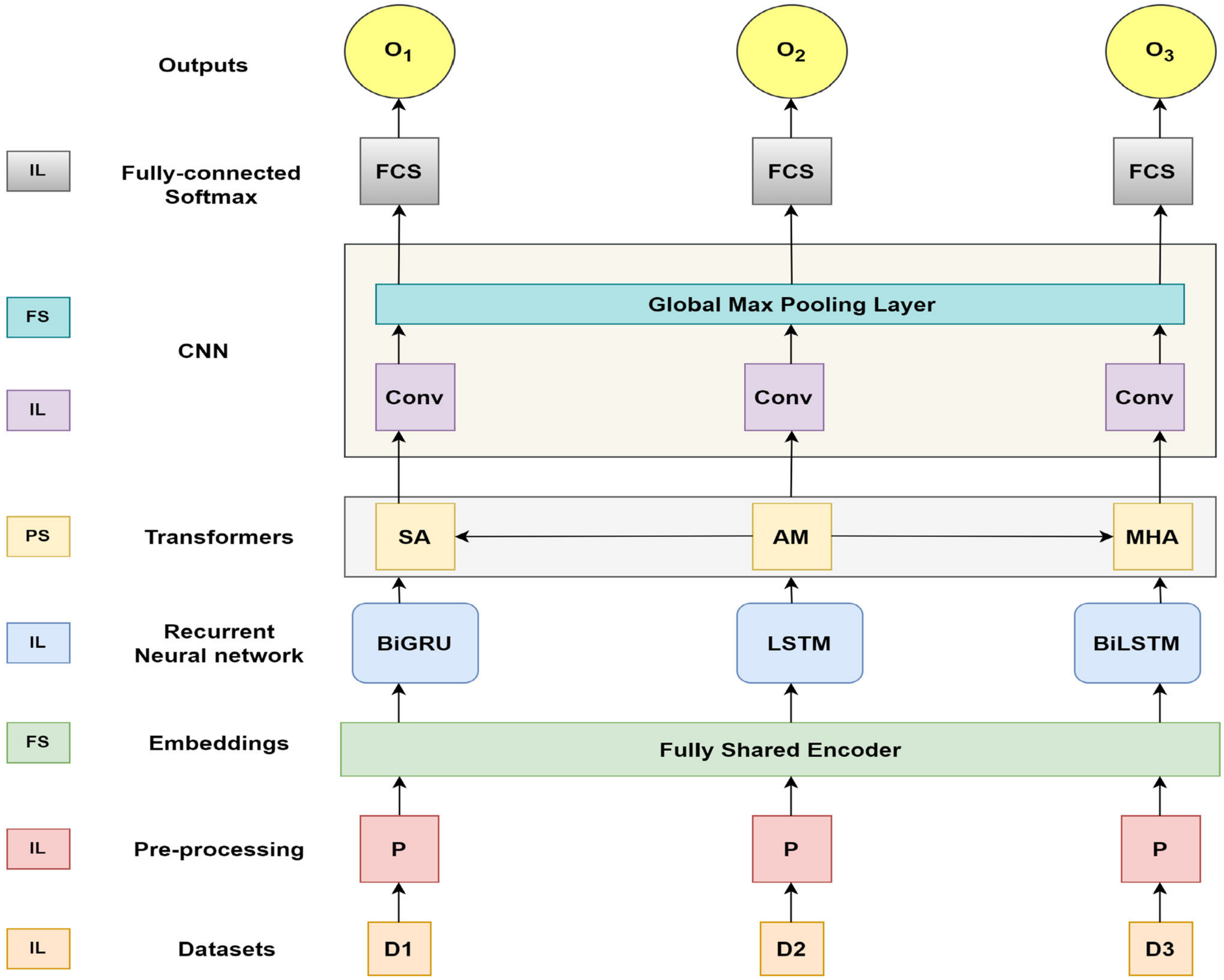


Fig. 3 The architecture of proposed models with SPS for MTL

each model is allocated to a specific task based on its superior performance in that particular task.

4.1 S-ARLM approach

This paper suggests the S-ARLM approach to extract the embedding representations. It is applied over autoencoding and context-free methods and considers the auto-regressive method to maintain the embedding values. In masked-language modeling (MLM)-based models, around 15% of tokens are masked. These models estimate only the masked tokens. Here, the auto-regressive (AR) method organizes the nearest token's subsets in its assessment, such that the expected tokens are not independently detected. Here, all tasks share the common embedding layer, which denotes the S-ARLM.

The AR method employs various techniques, including random permutations, to maintain the integrity of long text data. The AR embeddings E utilize the transformer-XL approach to accommodate inputs of varying lengths. The MLM can only execute the fixed length of inputs. Because of this limitation, the AR model is considered in classification chores. The specific input using the permutation method for each layer depends on the preceding input features from the previous hidden layer. However, auto-regressive-based permutation is applied to select the bidirectional tokens without changing the tokens' order and does not hide any token like MLM-based models.

Moreover, the AR approach is recommended over autoencoding, denoising processes, and bidirectional modeling contexts, demonstrating superior performance compared to MLM-based models. Here, denoising autoencoding and MLM processes rely on frequent masking tokens in the same position and ignore the position of the input tokens. The proposed S-ARLM component provides an attention mechanism on each input token. It not only compares the adjacent token but also correlates with other distant tokens without changing the order of the tokens. Nevertheless, the above process continues till the expected likelihood is maximized. Thus, a maximum number of permutations are extracted, and the factorization of the permutations is stored for reference. This process executes the transformer-XL, which is necessary to allocate the position for each token.

The S-ARLM is grounded in the concept of leveraging auto-regressive language modeling to capture contextual dependencies in mental health texts across multiple tasks simultaneously. It capitalizes on the shared contextual features among tasks to enhance the model's performance and efficiency. In ARLM, for the given input sequences $X = (X_1, X_2, ..., X_N)$, the probability of a sequence is given by the product of the conditional probabilities of each token:

$$P(X|\Theta_{\text{LM}}) = \prod_{t=1}^{T} P(x_t|x_{<t}, \Theta_{\text{LM}}) \tag{8}$$

Here, Θ_{LM} is the parameters of the language model, which include the weights and biases learned during training. The t-th token in the sequence X, for which the probability is being computed. $x_{<t}$ represents the tokens preceding x_t, while the subset of tokens that precede x_t in the sequence, including all tokens from x_1 to x_{t-1}, where N is the length of the input sequence, and $P(x_t \mid x_{<t}, \Theta_{\text{LM}})$ is the conditional probability of the t-th token x_t given the preceding tokens $x_{<t}$ and the model parameters Θ_{LM}. $\prod_{t=1}^{T}$ represents the product of the conditional probabilities over all tokens in the sequence.

Algorithm 1 S-ARLM for MTL

```
Input: $X_1, X_2, X_3$: Input sequences for tasks $T_1, T_2, T_3$
Output: $H_1, H_2, H_3$: Task-specific feature matrices
Function TrainModel($S$-$ARLM$):
    Initialize $\Theta_{\text{shared}}$ and $\Theta_1, \Theta_2, \Theta_3$ randomly // Shared model parameters
    Initialize loss function $\mathcal{L}$ to 0
    Train the shared auto-regressive language model
    Compute shared representation $H_{\text{shared}}$ using the shared model:
    $H_{\text{shared}} = f_{\text{shared}}(X_1, X_2, X_3;\ \Theta_{\text{shared}})$
    for $i = 1$ to $3$ do
        Update $\Theta_i$ by minimizing the loss function $\mathcal{L}_i$ using $H_{\text{shared}}$:
        $\min_{\Theta_i} \mathcal{L}_i(\Theta_{\text{shared}}, \Theta_i, X_{T_i}, Y_i)$
        Compute task-specific feature matrix $H_i$ using $\Theta_i$:
        $H_i = g_i(H_{\text{shared}};\ \Theta_i)$
    end
return $H_1, H_2, H_3$
```

4.1.1 S-ARLM framework for handling different tasks

In MTL, we aim to jointly optimize multiple tasks by sharing knowledge across them. Let $T = \{T_1, T_2, T_3\}$ denote the set of tasks related to mental health text analysis. Each task T_i has its own dataset X_{T_i} and labels Y_i. S-ARLM leverages a shared auto-regressive language model f_{shared} to extract common contextual features from input sequences across tasks. Let H_{shared} represent the shared representation obtained from f_{shared}.

$$H_{\text{shared}} = f_{\text{shared}}(X_1, X_2, X_3) \tag{9}$$

Next, task-specific mappings g_i transform H_{shared} into task-specific feature matrices H_i for each task T_i.

$$H_i = g_i(H_{\text{shared}}), \quad i = 1, 2, 3 \tag{10}$$

We aim to minimize the total loss across all tasks, which is a combination of individual task losses:

$$\min_{\Theta_{\text{shared}}, \Theta_1, \Theta_2, \Theta_3} \sum_{i=1}^{3} \frac{1}{N_i} \sum_{j=1}^{N_i} \mathcal{L}_i(\Theta_{\text{shared}}, \Theta_i, X_{T_i}, Y_{i,j}) \tag{11}$$

Here, $\mathcal{L}_i$ represents the loss function for task T_i with shared parameters Θ_{shared} and task-specific parameters Θ_i. N_i represents the number of samples in the i-th dataset X_{T_i}. $D_{i,j}$ and $Y_{i,j}$ represent the j-th sample and its corresponding label in the i-th dataset X_{T_i}, respectively.

The S-ARLM algorithm 1 leverages a shared auto-regressive language model to extract common contextual features from input sequences across multiple tasks, facilitating MTL in mental health text analysis. By jointly optimizing the tasks with shared parameters, it efficiently captures complex dependencies and patterns inherent in mental health texts, addressing issues such as task interference and limited adaptability encountered by existing models. The algorithm works by initializing and training a shared model on the input sequences, computing a shared representation that encapsulates task-relevant information.

Subsequently, task-specific parameters are updated based on this shared representation, allowing for the generation of task-specific feature matrices. This approach enables effective parameter sharing and knowledge transfer between tasks, leading to enhanced model performance and generalization across diverse mental health text analysis tasks. Through its shared parameterization and multi-task optimization, the S-ARLM algorithm effectively mitigates task interference and improves adaptability, thus advancing mental health detection in texts. The outputs obtained from the S-ARLM algorithm include task-specific feature matrices H_1, H_2, H_3, where H_i represents the extracted features specific to each task T_i. These feature matrices encapsulate task-relevant information extracted from the shared representation, enabling downstream tasks such as classification or prediction in mental health text analysis.

4.1.2 Auto-regressive-based permutation encoder representations

Autoencoder language modeling and denoising autoencoding models perform regular fine-tuning procedures with the same input positions in each iteration such that the model learns only similar input values in the exact location, which creates a significant impact while training the model. Learning similar features using the special token [MASK] can lead to a decrease in the model's performance. In order to avoid such discrepancies, S-ARLM with an auto-regressive-based permutation component is suggested instead of autoencoder language modeling and denoising autoencoding models for model training.

The ARLM is a neural network-based technique used to predict the next token in a sequence based on a set of preceding tokens. This approach considers the interdependence of all previous tokens, involving the consideration of all possible permutations of tokens in a sequence. Furthermore, we emphasize the substantial advantages of incorporating the enhanced permutation language modeling approach alongside ARLM. This combined approach helps retain bidirectional contextual features and preserves the order of each token permutation. The ARLM method with XLNet shows improved performance of our model than earlier models for classification tasks. The enhanced permutation language modeling estimates the conditional probabilities to calculate p(S) from an input sequence $S = [I_1, I_2, .., I_t]$, and then ARLM representations are applied without losing the token contexts.

$$p(I_{u_t}) = (I \mid I_{u_{<t}}) \tag{12}$$

Considering all permutations of the length U_N, U is used for the set of all permutations and index sequence N is defined as $[I_1, I_2, .., I_N]$. u is particular permutation denoted from U_N and permutation for t^{th} element is u_t. $I_{u_{<t}}$ is used to denote the sequence of tokens or elements before the t-th position in the permutation u_t. Also, u_t means the $t-1$ elements. Meanwhile, query, key, and value matrices are Q, K, and V, respectively. The result is a m query sequence operator with a hidden layer h, such as conditional probability for u. The permutation-based representations are calculated through the softmax function. In each permutation, the input position is calculated along with u_t, and the encoder representations of h are to maintain u_t. Then, estimate the sequence of input values with h and handle these elements with Qo. Each S contains the content stream c at h_i position in the hidden layer, which is a hidden state of the transformer and computes bidirectional contexts and U_t original token. This can be further represented as follows:

$$c_{u_t}^{(m)} \leftarrow Attention(Q = c_{u_t}^{(m-1)}, K, V = c_{U_{\leq t}}^{(m-1)}) \tag{13}$$

Then, c contextual representations at the corresponding phase depend on the u_t position and I_{u_t} feature. Here, the permutation position is assumed with h rather than v because it is applied only to understand conditional probability. The query operator Qo is considered for each u_t, while Qo contains the attention query vector for permutation u_t at layer m. This vector is used to compute attention scores by querying the key vectors.

$$h_{u_t}^{(m)} = Attention(Q = h_{u_t}^{(m-1)}, K, V = c_{U_{\leq t}}^{(m-1)}) \tag{14}$$

The term $h_{u_t}^{(m)}$ represents the updated hidden state for the t-th permutation u_t at layer m of the model. However, the query stream refers to the set of vectors used to generate attention scores. In Eq. 14, $h_{u_t}^{(m-1)}$ is the query stream at layer $m-1$. It helps in evaluating the relevance or similarity of different parts of the input sequence in relation to the current context or query, while Qo for u_t can be calculated along with the weight parameter w. Auto-regressive-based features are preserved in both $h_{ut}^{(m)}$ and content stream $c_{ut}^{(m)}$ related attention mechanisms. Also, calculate the final hidden features of the auto-regressive method. Here, content stream and query stream values are stored in both hidden layers $c_{u_t}^{(m)}$ and $h_{u_t}^{(m)}$ using the projection matrices Q, K, and V. Subsequently, derive the permutation feature c_t^0 from the embedding context e^c using the given inputs I_t, while h_t^0 incorporates suitable weights. c_t^0 and h_t^0 represent specific components related to the permutation-based approach in the auto-regressive model. E_1, E_2, E_3 represent the individual embedding outputs for three distinct tasks, while $c_t^1, Qo_t^1, c_t^2, Qo_t^2, c_t^3$, and Qo_t^3 denote the embedding features extracted from both $c_{ut}^{(m)}$ and $Qo_{ut}^{(m)}$, respectively. Here, g_t^1, g_t^2, and g_t^3 are features or transformations corresponding to the specific permutations or tasks. They are used alongside c_t^1, c_t^2, and c_t^3 to represent the embedding outputs for different tasks.

$$E_1 = c_t^1, g_t^1 \tag{15}$$

$$E_2 = c_t^2, g_t^2 \tag{16}$$

$$E_3 = c_t^3, g_t^3 \tag{17}$$

Hence, all embedding features from various tasks are consolidated into a unified layer. Our S-ARLM technique offers significant advantages, including consolidating embedding representations into a single channel without replicating the entire layer and retaining both contextual data and the sequence of each input position within the repository.

4.2 SPS-LSTM-AM model

The contextual features from the auto-regressive method are passed to the deep neural networks. Embedding features E_2 are fed to the LSTM component that consists of different gates such as input i_t, forget f_t, and output o_t gates. The input vector is represented as $I_t \in \mathbb{R}^l$, where l indicates the dimension of the input vector, $\mathbb{R}$ used to denote vector spaces or a real-valued space, and $W \in \mathbb{R}^{\mathbf{v}} \times l$ and $b \in \mathbb{R}^{\mathbf{v}}$ represent learnable parameters associated with the vocabulary size denoted by $\mathbf{v}$. Within the LSTM, x_t represents the input vector, h_{t-1} denotes the previous hidden state, and c_{t-1} represents the previous cell state. The current hidden and cell states are denoted as h_t and c_t, respectively. The symbol × signifies point-wise vector multiplication, while b_i, b_f, b_o, and b_c are trainable parameters used during model training. The forget gate f_t determines whether to retain or discard input values from the earlier state c_{t-1}, potentially retaining features from x_t and h_{t-1}. The update gate determines which features from x_t and h_{t-1} should be used to update the cell state. The input values of c_t represent the resulting cell state, which is influenced by i_t, f_t, and c_{t-1}. Subsequently, o_t manages the flow of data from c_t to h_t.

A deep learning method incorporates an attention mechanism (AM) to tackle the challenge posed by long-term dependencies. Each input value corresponds to a hidden state, which is then inputted into the decoder segment. At each step of the decoder channel within the AM, the hidden states are considered. This mechanism enables attention generation to the preceding decoder phase, akin to the encoder's hidden states, where q query, k key, and v value can be amalgamated. The output is the weighted average w of the v, with weights represented by k and q reflecting their functional compatibility; k and q are distinct vectors. The AM consists of two components: SA and MHA, which are shared with the other two models, namely SPS-BiGRU-SAM and SPS-BNN-MHAM models.

The AM extracts q from the previous hidden layer' features, representing the pair of k and v, vectors linked with k, v, and q. The weighted amount of the input values is assumed from the above result. Simultaneously, the weight assigned to each value is denoted as q with the corresponding k. Nonetheless, the LSTM generates the sequence $(I_{h_1}, I_{h_2}, .., I_{hN})$ from the given input values. hN refers to the last element or feature in the sequence generated by the LSTM. The subscript N indicates that it is the N-th element or position in the sequence. Here, keys $(k_1, .., k_n)$, q, and values $(v_1, .., v_n)$ are further calculated below:

$$S_1 = (QK^{\mathbf{T}}/\sqrt{d_k})V \tag{18}$$

$$Mh_1(Q, K, V) = Concat(h_1, .., h_h)W^O \tag{19}$$

The term $K^{\mathbf{T}}$ is the transpose of the key matrix, and $\sqrt{d_k}$ is the square root of the dimension of the keys, used for scaling. The equation calculates the attention scores by performing the dot product between Q and $K^{\mathbf{T}}$, scaling by $\sqrt{d_k}$, and then using these scores to weight the values in V. Compute the self-attention (SA) scores and pass them into different linear layers organized by multi-heads. In the MHA element, Mh_1 denotes multi-head, and h_1, h_2 are different heads with orthogonal projection-based weight matrix W^O for task 1. Here, attention features AI_k with related C context value and hidden value h_k vectors are considered final attention features.

$$h_1 = (C, h_k, AI_k) \tag{20}$$

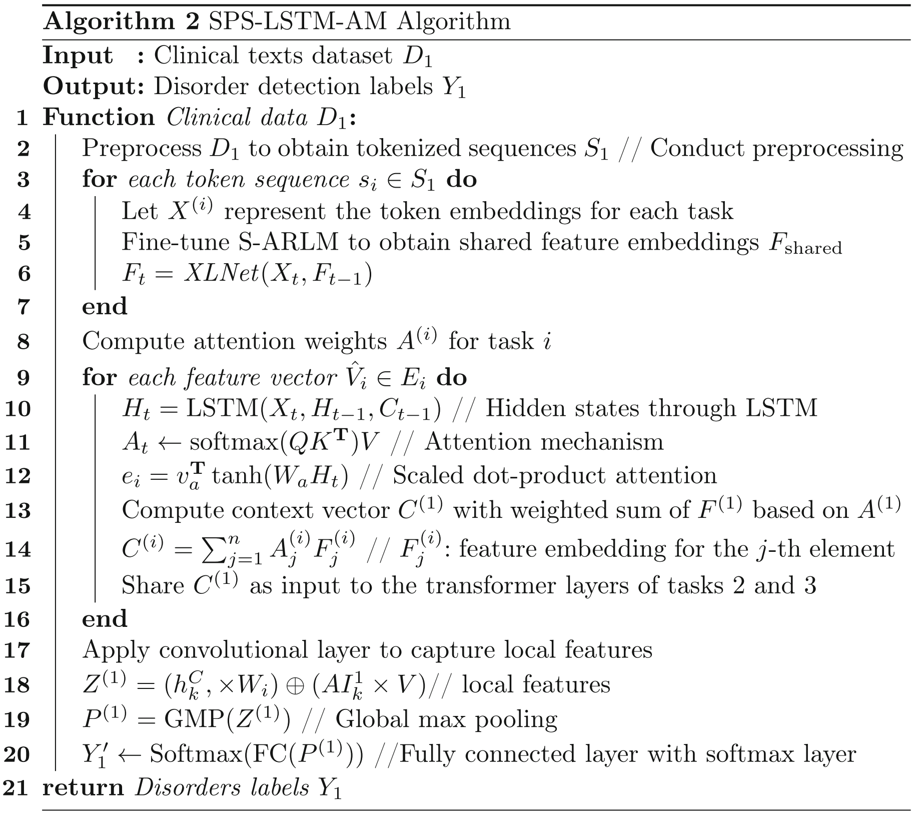
The concatenating MHA and convolution feature CZ_1 for task 1 using a splicing operator $\oplus$.

$$CZ_1 = (h_k^C, \times W_i) \oplus (AI_k^1 \times V) \tag{21}$$

The term h_k^C represents the hidden value vector for task 1 from the convolutional layer, while W_i denotes the weight matrix applied to h_k^C. The vector AI_k^1 corresponds to the attention weights for task 1, and V is the value matrix. The concatenated feature CZ_1 is derived by combining the weighted hidden values from the convolutional layer with the weighted attention values.

The SPS-LSTM-AM model for detecting disorders in clinical texts involves a detailed processing pipeline that begins with preprocessing the texts to clean and standardize the data (Fig. 2a). These preprocessed texts are then fed into a fully shared encoder layer, which serves to learn and extract combined contextual representations that are common across all three models. This shared layer ensures that the foundational semantic and syntactic features are uniformly captured, providing a robust basis for further analysis. Following this, the data is passed to the LSTM layer, which specializes in capturing sequential dependencies and long-range relationships within the text, which is crucial for understanding the temporal aspects of clinical narratives. The output from the LSTM layer is then processed by an attention mechanism, which selectively focuses on the most relevant parts of the input data, enhancing the model's ability to identify critical features indicative of specific disorders. Some of these attention-derived features are partially shared with other models, allowing for cross-task feature learning and transfer of important weightings. Subsequently, the refined features are passed through a CNN layer, which aids in detecting local patterns and features. The CNN output is fully shared with a global max pooling (GMP) layer across the three models, summarizing the most significant features while maintaining consistency in feature extraction. Finally, the processed data are input to a fully connected softmax layer, which classifies the data into specific disorders, providing the final diagnostic output. This comprehensive architecture leverages both shared and task-specific features, ensuring accurate and efficient disorder detection. Note that the descriptions between each component refer to outcomes, which are then fed into the subsequent component as depicted in Fig. 2a–c.

Algorithm 2 SPS-LSTM-AM algorithm



```
Algorithm 2 SPS-LSTM-AM Algorithm
Input   : Clinical texts dataset $D_1$
Output: Disorder detection labels $Y_1$
Function Clinical data $D_1$:
    Preprocess $D_1$ to obtain tokenized sequences $S_1$ // Conduct preprocessing
    for each token sequence $s_i \in S_1$ do
        Let $X^{(i)}$ represent the token embeddings for each task
        Fine-tune S-ARLM to obtain shared feature embeddings $F_{\text{shared}}$
        $F_t = XLNet(X_t, F_{t-1})$
    end
    Compute attention weights $A^{(i)}$ for task $i$
    for each feature vector $\hat{V}_i \in E_i$ do
        $H_t = \text{LSTM}(X_t, H_{t-1}, C_{t-1})$ // Hidden states through LSTM
        $A_t \leftarrow \text{softmax}(QK^{\mathbf{T}})V$ // Attention mechanism
        $e_i = v_a^{\mathbf{T}} \tanh(W_a H_t)$ // Scaled dot-product attention
        Compute context vector $C^{(1)}$ with weighted sum of $F^{(1)}$ based on $A^{(1)}$
        $C^{(i)} = \sum_{j=1}^{n} A_j^{(i)} F_j^{(i)}$ // $F_j^{(i)}$: feature embedding for the $j$-th element
        Share $C^{(1)}$ as input to the transformer layers of tasks 2 and 3
    end
    Apply convolutional layer to capture local features
    $Z^{(1)} = (h_k^C, \times W_i) \oplus (AI_k^1 \times V)$// local features
    $P^{(1)} = \text{GMP}(Z^{(1)})$ // Global max pooling
    $Y_1' \leftarrow \text{Softmax}(\text{FC}(P^{(1)}))$ //Fully connected layer with softmax layer
return Disorders labels $Y_1$
```

The SPS-LSTM-AM algorithm 2 aims to detect disorders in texts. First, it converts the given texts into token sequences and extracts token embeddings. It fine-tunes the S-ARLM to obtain shared feature embeddings and utilizes XLNet to process token embeddings, generating output feature vectors. Attention mechanisms compute attention weights for each task, facilitating task-specific feature identification. LSTM processing computes hidden and cell states, while attention mechanisms generate context vectors shared across tasks. Local feature extraction via convolutional layers captures task-specific patterns, and GMP consolidates features for classification. The algorithm leverages shared parameterization and feature sharing mechanisms to enhance disorder detection.

4.3 SPS-BiGRU-SAM model

The bidirectional hidden features are captured from the encoder layer using the BiGRU component. The model considers the BiGRU component to maintain the sequential dependencies. Then, information passes into two hidden layers, such as $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$. Here, w_t and v_t are corresponding weights for BiGRU.

$$h_k^2 = (\overrightarrow{h}_t + v_t)(\overleftarrow{h}_t + b_t) \tag{22}$$

The SA component is shared from the attention mechanism. Instead of sharing the entire transformer layer, one particular vector, such as SA, is considered for model building. It calculates the weighted average of input features in place of the weight computed for the similarity score, which combines input features. The input matrix $I \in \mathbb{R}^{n \times d}$ represents a sequence of n input vectors, each of dimension d. The matrices $W^V \in \mathbb{R}^{d \times d_v}$, $W^Q \in \mathbb{R}^{d \times d_q}$, and $W^K \in \mathbb{R}^{d \times d_k}$ are used to project the input vectors into value, query, and key vectors, with dimensions d_v, d_q, and d_k, respectively. To extract hidden values with Q, K, and V are denoted below:

$$S_2^C = (QK^{\mathbf{T}}/\sqrt{d_k})V \tag{23}$$

Here, S_2^C represents the contextual features for task 2. The dimension vector $\sqrt{d_k}$ is derived from the respective matrices, typically referred to as the row-wise normalization procedure. Consequently, each element in S_2^C is influenced by all other elements within the same sequence. The resulting SA features and convolutional layer for task 2 are represented as follows:

$$h_2 = (S_2^C, h_k^2, AI_k^2) \tag{24}$$

$$CZ_2 = (S_2^C, h_k^2 \times W_i) \oplus (AI_k^2 \times V) \tag{25}$$

In Eq. 24, h_2 denotes the result of combining the contextual features S_2^C, the hidden value vector h_k^2, and the attention weight vector AI_k^2. In Eq. 25, CZ_2 is defined as the concatenation of S_2^C with the weighted hidden value vector $h_k^2 \times W_i$, where W_i is the weight matrix, and the weighted attention values $AI_k^2 \times V$. The splicing operator $\oplus$ is used to concatenate these components.

To classify social media texts as suicide or non-suicide-related, the SPS-BiGRU-SAM model starts with preprocessing the texts, which is uniform across all three models (Fig. 2b). These preprocessed texts are input into a fully shared encoder that learns combined contextual representations across three models. The encoder generates a comprehensive representation of the text, capturing nuanced information pertinent to suicide detection. This representation is then passed to a BiGRU layer, which processes the sequence in both forward and backward directions to enhance understanding of context. The output of the BiGRU is fed into a self-attention mechanism where some features are partially shared, allowing the model to focus on significant parts of the input while leveraging shared weights from model 2. The refined features are then processed through a CNN to capture hierarchical patterns and through a GMP layer that aggregates features across the models. The aggregated features are passed through a fully connected softmax layer that classifies the texts into suicide or non-suicide-related categories, delivering the final outcome based on the model's comprehensive analysis.

Algorithm 3 SPS-BiGRU-SAM Algorithm

```
Input  : Social media dataset $D_2$, partial sharing context feature vector $C^{(1)}$
Output: Suicide and non-suicide labels $Y_2$
Function Social data $D_2$:
    $X' \leftarrow \text{Preprocess}(D_2)$ // Conduct preprocessing
    for each token sequence $w_i \in X'$ do
        $E_i^{(2)} \leftarrow \text{XLNet}[\hat{V}_1^{(2)}, \hat{V}_2^{(2)}, \ldots, \hat{V}_n^{(2)}]$ // Get feature embeddings
    end
    Feature vectors are passed to the subsequent layer
    for each feature vector $\hat{V}_i^{(2)} \in E_i^{(2)}$ do
        $h_k^2 = (\overrightarrow{h}_t + v_t)(\overleftarrow{h}_t + b_t))$ // Hidden states through BiGRU for Task 2
        Use $C^{(1)}$ as input for self-attention mechanism in Task 2
        $S_2^C = (QK^{\mathbf{T}}/\sqrt{d_k})V$ // Self attention mechanism
        $A_t^{(2)} \leftarrow \text{softmax}(Q^{(2)}K^{\mathbf{T}^{(2)}}V^{(2)} + Q^{(1)}K^{\mathbf{T}^{(1)}}V^{(1)}$ // Partial sharing
        Compute context vector $C^{(2)}$ as weighted sum of $F^{(2)}$ based on $A^{(2)}$
        $C^{(2)} = (S_2^C, h_k^2, AI_k^2)$// Gets final feature representations
    end
    Apply convolutional layer to capture local features
    $Z_{\text{local}}^{(2)} = \sigma\left(\mathbf{W}_{\text{local}}^{(2)} \times C^{(2)} + \mathbf{b}_{\text{local}}^{(2)}\right)$ // Local features
    $P_{\text{global}}^{(2)} = \max\left(Z_{\text{local}}^{(2)}\right)$ // Global max pooling
    $Y_{\text{softmax}}^{(2)} \leftarrow \text{Softmax}\left(\mathbf{W}_{\text{softmax}}^{(2)} \times \mathbf{P}_{\text{global}}^{(2)} + \mathbf{b}_{\text{softmax}}^{(2)}\right)$
return Suicide and non-suicide labels $Y_2$
```

The SPS-BiGRU-SAM algorithm is designed for the classification of suicide and non-suicide labels within social media texts. It leverages a combination of fully shared and partially shared components to enable efficient knowledge and feature sharing across different tasks. The algorithm begins by preprocessing the social media text dataset and converting it into tokenized sequences. These sequences are then fed into an XLNet model, which is fully shared across all tasks, allowing for the extraction of contextualized feature embeddings. Simultaneously, a BiGRU layer is employed to capture task-specific features. The algorithm utilizes a self-attention mechanism, where contextual features from Task 1 are partially shared with Task 2, enhancing the model's ability to capture relevant information. The self-attention mechanism computes attention weights based on shared context vectors and task-specific features, refining the representation of the input data. Following this, the algorithm applies a convolutional layer to capture local features, followed by GMP to aggregate features across the entire dataset. Finally, separate fully connected layers with softmax activation functions are employed for each task to generate the classification output.

4.4 SPS-BNN-MHAM model

Extract the embedding values from E_3 to feed into BiLSTM. Here, hidden representations are maintained in each forward $\overrightarrow{h_t}$ and backward $\overleftarrow{h_t}$ hidden layers, respectively. bh_t indicates the bidirectional hidden layers.

$$bh_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{26}$$

The MHA module consolidates insights from the attention mechanism. It can gather attention scores from both preceding and succeeding contexts of a given input, ensuring the preservation of attention scores with semantic-driven long-range dependency. Multiple linear layers are incorporated to preserve the text sequence, signifying that the MHA transformer links input values across various positions within the sequence. In this context, Mh_3 represents multi-head with I input features for task 3.

$$Mh^3 = Attention(IW_Q, IW_K, IW_V) \tag{27}$$

Here, h_3 is the ultimate hidden state layer, which is denoted as follows:

$$h_3 = (C^3, bh_t, Mh^3) \tag{28}$$

C^3 denotes the context vector specific to task 3. This vector encapsulates the contextual information relevant to task 3, which is combined with other components like bh_t and Mh^3 to form the final representation h_3.

Next, the process of extracting convolution features for task 3 is detailed below:

$$CZ_3 = (C^3, bh_t \times W_i) \oplus (Mh^3 \times V) \tag{29}$$

Here, fully sharing the pooling layer, thus considering the common pooling layer for all tasks. Considering the x_i pooling operator over the extracted hidden features CZ_1, CZ_2 and CZ_3. Apply a fully connected (FC) softmax layer for all tasks individually. Equation 30 describes how the feature vector CZ_n is computed in an SPS-based MTL framework. While r max-pooling operation applied to the concatenated feature maps CZ_1, CZ_2, and CZ_3. This pooled result is then combined with the weight matrix W_c and the bias term b_{ic}. Finally, the function h_k processes this combined input to produce CZ_n.

$$CZ_n = h_k(r(CZ_1, CZ_2, CZ_3) + W_c, b_{ic}) \tag{30}$$

$$Y_n = \text{Softmax}\left(W_y \times \text{FC}(CZ_n) + b_y\right) \tag{31}$$

In this Eq. 31, Y_n describes the final prediction layer. W_y denotes the weight matrix applied to the feature vector FC(CZ_n), which is derived from the concatenated and pooled features CZ_n. The bias vector b_y is added to the linear transformation before applying the softmax function. This process converts the logits into class probabilities Y_n, enabling the model to perform multi-class classification by predicting the likelihood of each class based on the input features.

The SPS-BiLSTM-MHA model for detecting emotions in suicide notes includes several key steps (Fig. 2c). First, text data are preprocessed through tokenization, normalization, and embedding, a procedure used across all models. This preprocessed text is input into a fully shared encoder, which learns combined contextual representations for different tasks. The encoder's output is passed to a BiLSTM network that captures both forward and backward dependencies, adding temporal context. Next, the data are processed through an MHA mechanism, where features are partially shared with weightings from model 2, integrating task-specific insights while maintaining shared representations. The resulting features from MHA are handled by a CNN layer, followed by a GMP layer to extract the most relevant features. Finally, these features are fed into a fully connected softmax layer to classify the emotions in suicide notes. The entire approach combines both shared and partially shared components to improve the model's effectiveness in detecting emotions.

Algorithm 4 SPS-BiLSTM-MHA model for task 3

```
Input  : Suicide notes dataset $D_3$, partial sharing context feature vector $C^{(1)}$
Output: Negative emotion labels $Y_3$
Function Suicide notes $D_3$:
    $X' \leftarrow$ Preprocess($D_3$) // Conduct preprocessing
    for each token sequence $s_i \in X'$ do
        Fine-tune S-ARLM to obtain shared feature embeddings $F_t$
        $F_t = XLNet(X_t^{(3)}, F_{t-1}^{(3)})$ // $F_t$ is the output feature vector
    end
    for each feature vector $\hat{V}_i^{(3)} \in E_i^{(3)}$ do
        $\overrightarrow{h}_t \leftarrow \text{LSTM}(\hat{v}_i, \overrightarrow{h}_{t-1})$ // Forward LSTM
        $\overleftarrow{h}_t \leftarrow \text{LSTM}(\hat{v}_i, \overleftarrow{h}_{t-1})$ // Backward LSTM
        $h_k^3 \leftarrow [\overrightarrow{h}_{t-1}, \overleftarrow{h}_{t-1}]$ // Hidden states through BiLSTM for Task 3
    end
    while attention heads $< n$ do
        Consider $C^{(1)}$ as input for MHA mechanism in Task 3, where $n = 12$
        $Mh^3 = Attention(IW_Q, IW_K, IW_V)$ // Projection vectors
          $\text{MH}(Q, K, V) = Concat(head_1, ., head_n)W^O$ //MH: Multi attention
        $A_t^{(3)} \leftarrow \text{softmax}(Q^{(3)}K^{(3)\mathbf{T}} + Q^{(1)}K^{(1)\mathbf{T}}V^{(1)})$ // Partial sharing
        Compute context vector $C^{(3)}$ as weighted sum of $F^{(3)}$ based on $A^{(3)}$
    end
    $h_c \leftarrow \text{CNN}(A_t^{(3)})$ // Apply CNN layer
    $P^{(3)} = max(Z^{(3)})$ // Global max pooling
    $Y^{(3)} \leftarrow Softmax(FC^{(3)}(P^{(3)}))$ // Compute softmax probabilities
return Negative emotion labels $Y_3$
```

The SPS-BiLSTM-MHA algorithm 4 for Task 3 is designed to detect negative emotions in suicide notes from a dataset D_3. This model leverages the shared knowledge and features obtained from XLNet embeddings, ensuring a comprehensive understanding of the textual data. By utilizing a BiLSTM layer, the model captures the nuanced temporal dependencies within the sequences. The MHA mechanism enhances the model's capability to focus on relevant parts of the input while considering shared context features from Task 1, contributing to a deeper contextual understanding. Moreover, the model incorporates both fully shared and partially shared components for feature extraction and parameter updates, allowing for effective knowledge transfer across tasks.

Each task has a different loss function (L_i), where $i \in \{1, 2, 3\}$. Here, γ_1, γ_2, and γ_3 are different tasks.

$$Loss = \gamma_1 L_1 + \gamma_2 L_2 + \gamma_3 L_3 \tag{32}$$

The cross-entropy function for each task is denoted as follows:

$$L^n_{CE} = -1/n\ \Sigma^n_{i=0}\ y_i log \hat{y}_i + (1 - y_i) log(1 - \hat{y}_i) \tag{33}$$

We select the predicted label $\hat{y}$ from the class y_i to compute the loss function. To mitigate data overfitting, we employ the Adam optimizer. Additionally, a dropout layer is incorporated to handle variations in the data. Figure 3 shows the proposed approach for SPS. Here, each layer is explained in Sects. 3 and 4, respectively. The architectural design of our proposed models incorporates SPS to facilitate MTL. This framework allows tasks to share certain parameters, enhancing efficiency and information exchange while minimizing redundancy. In addition, time and space complexity of the proposed models is as follows: Suppose N be the number of data samples, D be the dimension of word embeddings, T be the maximum sequence length, H_shared be a dimension of hidden states for shared features in the XLNET layer, and K be the number of tasks (in this case, $K = 3$ for disorder detection, suicide or non-suicide-related classification, and emotion detection). Additionally, time complexity for S-ARLM is training time $T_{shared} : O(N \times T_{shared} \times H^2_{shared})$, inference time $(T_{inference_{shared}}) : O(N_{shared} \times H^2_{shared})$, BISTM with attention mechanism is $O(N \times T \times D)$, LSTM with MHA mechanism is $O(K \times N \times T^2 \times D)$, and BIGRU with SA mechanism is $O(N \times T \times D)$. The overall time complexity is $O(K \times N \times T \times D)$, which depends on how tasks share features, the number of shared parameters, and the optimization process, making it task-specific. Space complexity for S-ARLM is $O(N \times T \times D \times H_{shared})$, BISTM with attention mechanism is $N \times D^2$, LSTM with MHA is $O(K \times N \times T^2 \times D)$, and BiGRU with SA mechanism is $O(N \times T \times D)$. The overall space complexity is $O(N \times T \times D)$, which depends on the sharing mechanism and is based on our architecture used for MTL. The proposed algorithm for MTL is shown in Algorithm 5.

However, in the proposed architecture (Fig. 3), layer sharing is implemented when tasks benefit from leveraging common intermediate representations learned at specific layers of the model. In mental health analysis, shared layers process diverse types of textual data, such as clinical mental health records, Reddit posts about suicide, and suicidal notes. Despite differences in content, these texts often exhibit overlapping emotional and psychological patterns. For example, in the S-ARLM model, shared layers extract generalized features relevant to mental health, such as emotional distress and common language patterns. This approach allows the model to generalize from common patterns, enhancing tasks like disorder detection, suicide classification, and emotion analysis, and avoids redundant learning by building a unified understanding of mental health indicators.

Feature sharing is exemplified by the use of a partial sharing transformer mechanism. For example, in the SPS-LSTM-AM model, foundational text embeddings generated by initial layers are shared across tasks. These embeddings capture essential contextual and semantic features. Shared embeddings, such as those encoding psychological disorders and emotional tone, are used across layers and are responsible for tasks like emotion detection and suicidal text classification. The shared layers retain weights fine-tuned during training to adapt embeddings to task-specific nuances. This mechanism allows leveraging common features while providing flexibility for task-specific adaptations, ensuring consistent feature representation and enhancing the model's ability to generalize and tailor processing for different mental health-related tasks.

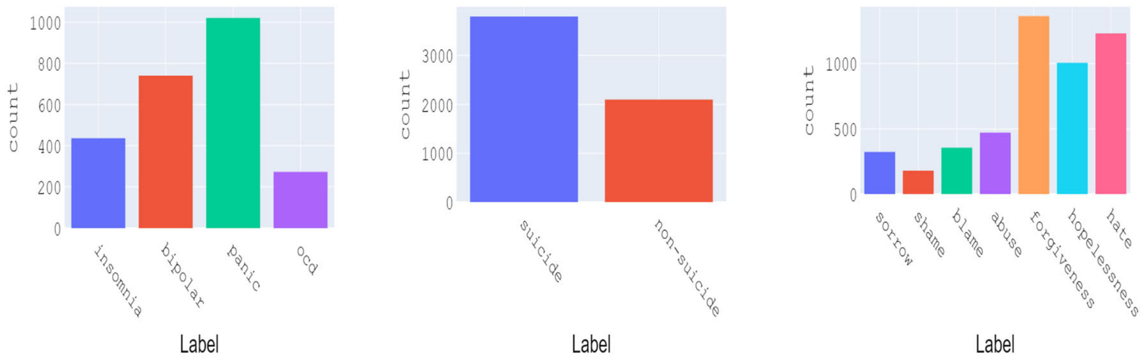
Algorithm 5 Soft-parameter sharing-based proposed methodology

```
   Input  : Mental health datasets $D_1, D_2, D_3$
   Output: Multi task outcomes $Y_1, Y_2, Y_3$
1  Function $D_1, D_2, D_3$:
2      $S'_1, S'_2, S'_3 \leftarrow$ Preprocess($X_{T_i}$) //Conduct preprocessing
3      for all word $\in S_i$ do
4          $XZ = e(x_t)$ and $gt^0$ // S-ARLM
5      end
6      Contextual feature vectors are passed to the subsequent layer
7      for each feature vector $\hat{V}_i \in XZ$ do
8          Computing the AR embeddings with deep neural networks
9          $BGh_t = (\overrightarrow{h}_t + v_t \times \overleftarrow{h}_t + b_t)$ // BiGRU component
10         $h_t = LSTM(x_t, h_{t-1})$ // LSTM component
11         $Bh_t = LSTM(\overrightarrow{h}_t, \overleftarrow{h}_t)$ // BiLSTM component
12         $H_i = \Sigma_{j=1}^{n} X_j \times v_j$ // Attention features
13         To extract features representations Q, K, and V
14         $Q = IW_Q, K = IW_K, V = IW_V$ // Projected matrices
15         The SA output for each input feature in a sequence
16         $(QK^{\mathbf{T}}/\sqrt{d_k}) = v_a^{\mathbf{T}} \tanh(W_a H_t)$ // Scaled-dot product scores
17         Combining input features $X_i$ and hidden state $h_n$ with the matrix $W^O$
18         $h_k = X_i h_n \times W^O$ // The attention output
19         The final hidden layer states with $\overset{\sim}{h}_k$
20         $\overset{\sim}{h}_k = Concat(head_1, head_2, .., head_n)$ // Final hidden features of MHA
21     end
22     Applying convolution with max-pooling
23     $Z = X_i \times r + \overset{\sim}{h}_k$ // Partial sharing CNN
24     Computing the $Z$ values with fully-connected softmax
25     $y_1 = \text{softmax}(w_i \times Z + b)$ // Compute individually
26 return $Y_1, Y_2, Y_3$ //Output for each task
```

In the proposed SPS-based MTL algorithm 5, we consider various datasets and conduct individual preprocessing tasks (line 2). In line 4, share the common auto-regressive-based model pre-training. Apply the deep neural networks with BiGRU (line 9 for model 1), LSTM (line 10 for model 2), and BiLSTM (line 11 for model 3). Extract the hidden

Table 1 Examples of MTL texts with class labels

Text	Dataset	Label
"My medications aren't working for mood stability, but my doctor doesn't want to change anything. What should I do? I have been on and off psych medications for almost eight years now I started to be medicated for major depression when I was 13, then they started treating me for bipolar. Along with the constant medication shifts, until my current psychiatrist refused to change my medication, I was hospitalized in a mental facility and underwent ect treatments and psychotherapy. I don't know what to do. Please help."	Clinical	Bipolar
"Hey, so right now, I probably have had the worst day in my entire life. I really don't know what to do, so for some reason, I just felt like sharing it somewhere, and this got to be it."	Reddit	Non-suicide
"I've struggled with suicidal ideation most of my life. I've attempted it before and am thinking about it again with more certainty. But I'm not sure what to do. I am stuck in my life, with being homeless coming up soon. I'm a month behind on my rent and getting laid off, so eviction isn't far. I have been fighting for so long. My friends have been there for me but can barely support themselves now."	Reddit	Suicide
"I am going to commit suicide, and nobody is responsible for my death."	CEASE-v2.0	Hopelessness



(a)

(b)

(c)

Fig. 4 Overall class labels **a** Clinical, **b** Reddit, **c**) CEASE-v2.0

features from the LSTM (line 12 for model 2) and then perform a weighted average summation over these hidden features (line 12 in model 2) using an attention mechanism. Define attention matrices based on the collected vectors (line 13 in model 2) and compute the attention-scaled dot product from these values (line 14 in model 2). This results in trainable weight matrices and calculates the attention score (line 15 in both model 1 and model 2) as the outcome of the SA mechanism for each input feature. Apply the attention heads to the resulting features (line 16 in models 2 and 3) and consider the multi-head values over these projected values (line 17 in models 2 and 3). Obtain the final MHA values. Pass the attention features through convolution with the pooling layer (line 22 in models 1, 2, and 3), and apply the fully connected softmax layer (lines 23 to 24 in models 1, 2, and 3) over the convolution features to perform the ultimate classification (line 25 in models 1, 2, and 3).

5 Experimental results

To demonstrate the ability of the proposed models, we first discuss the dataset collection, followed by a detailed discussion of the experiments conducted on these datasets and a comparison with state-of-the-art methods.

5.1 Dataset collection and preprocessing

We collected the data from different sources, such as 4932 texts from suicide notes, i.e., Corpus of Emotion Annotated Suicide notes in English (CEASE), version 2.0 dataset [59], and 5896 posts from the subreddit mental health, anxiety, AskDoc, psychiatry, and suicide watch using Reddit [5], and 2471 from clinical data, respectively. Clinical data are collected from online medical-related websites such as HealthTap and WebMD [60]. However, all datasets contain posts between the years 2015 and 2019. In the clinical dataset, each patient’s text contains manual labelings assigned by psychiatrists, while suicide and non-suicide-related classification is performed using the search terms like “worried, it’s clipped, to be killed, dead, self-injury, self-harm, take my life, cutting myself, wants to die, step off, shade, want to die, no longer, at rest, and self-slaughter” in suicide-related texts. Also, non-suicide terms are considered like “suicide attack, bomb, car attack, suicide attacks, and suicide hotline” [5]. However, the CEASE dataset contains manual labelings, which are annotated by three independent annotators. We divide our datasets for model training as 70%, testing and validation purposes about 30%. Consider the preprocessing procedure for collected texts from three datasets. For each dataset, we adhere to specific data preprocessing stages, such as identifying and removing duplicate records for data integrity. We eliminate hyperlinks and special characters, ensuring uniformity by converting text to lowercase. Examples of mental health-related texts with class labels are shown in Table 1.

5.2 Settings for proposed models

The suggested model settings are as follows: LSTM, BiLSTM, and BiGRU hidden layers are configured with 128 units each, and XLNet embedding dimensions are set to 768 with approximately 110 million parameters [61]. The model uses 12 MHA layers and a convolutional kernel size of 3. Optimization is performed with the Adam optimizer for regularization. Following this, GMP is applied, and a fully connected layer with a softmax function is used. The learning rate is set to 0.01, the loss function is cross-entropy, and dropout is applied with a rate of 0.1. Models are trained for various epochs, with the best performance achieved in 10 epochs and a batch size of 50. We experimented with several pre-training models, including

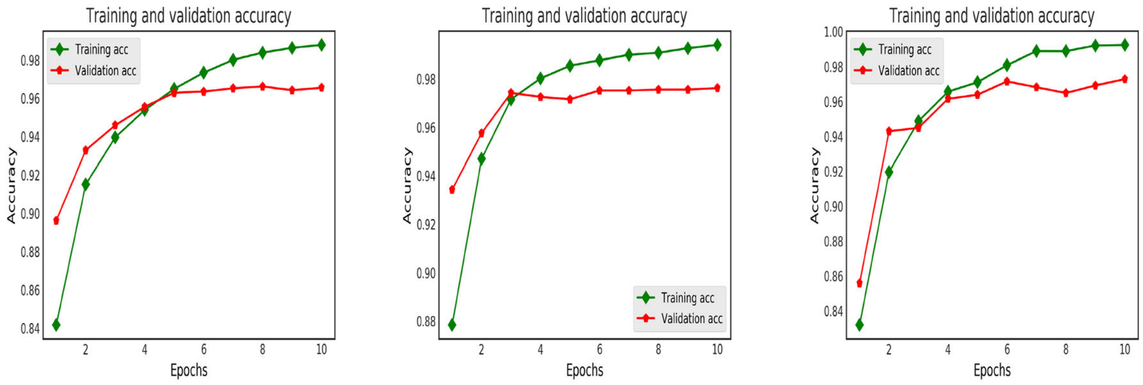
Word2vec with 300 dimensions [62], BERT with 768 dimensions [23], and XLNet. Additionally, hyperparameter tuning is crucial for optimizing model performance. In our study, we used the random search algorithm to identify the best parameters for improved efficiency. Figure 4 displays the overall class labels for the total number of texts.

5.3 Evaluation measures

Figures 5, 6, 7, 8 and 9 show the performance metrics such as accuracy, loss function, AUC-ROC, mean absolute error (MAE), and MSE for assessing the proposed models' performance. Figure 5 presents the accuracy performance of our proposed methodology across three datasets. The figure displays both validation and training accuracies, showing that the model achieved validation accuracies of more than 95% on all three datasets. The improvement in accuracy across epochs can be attributed to the iterative optimization process where the model continuously learns from the data, fine-tuning its parameters to enhance performance. As training progresses, the model becomes better at generalizing from the training data, which is reflected in the increasing accuracy rates.

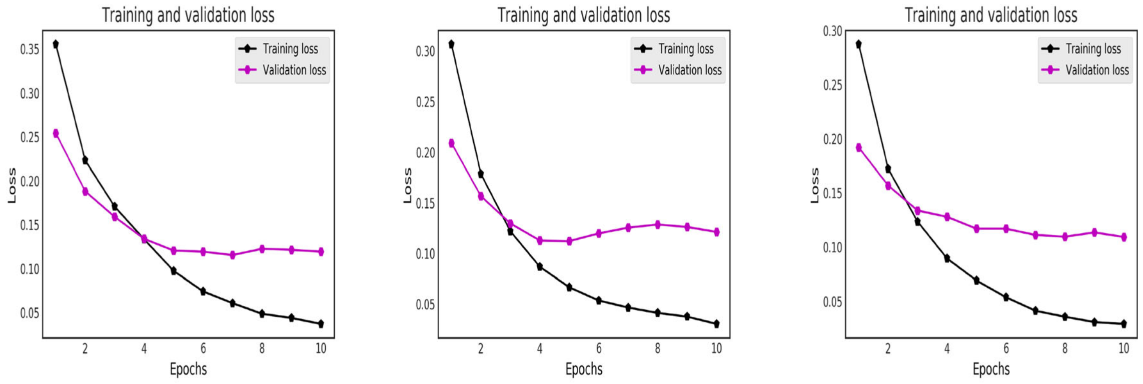
Figure 6 illustrates the loss function curves, providing a visual representation of the models' performance throughout the training process. The curves show a decrease in loss function values, with validation scores of approximately 0.11, 0.14, and 0.13 across the datasets. This decrease indicates that the models' predictions are becoming increasingly accurate over time. The regularization techniques implemented play a crucial role in stabilizing the loss function and preventing overfitting, thereby contributing to the models' improved overall performance. The loss function curves also provide a valuable guide for evaluating the impact of hyperparameters on training, allowing for further adjustments to optimize model dynamics.

Figures 7, 8, and 9 collectively provide a comprehensive evaluation of the MTL models' discriminative capacities, precision, and adaptability to diverse prediction types. Figure 7 illustrates the AUC-ROC curves, which evaluate the models' effectiveness in distinguishing between



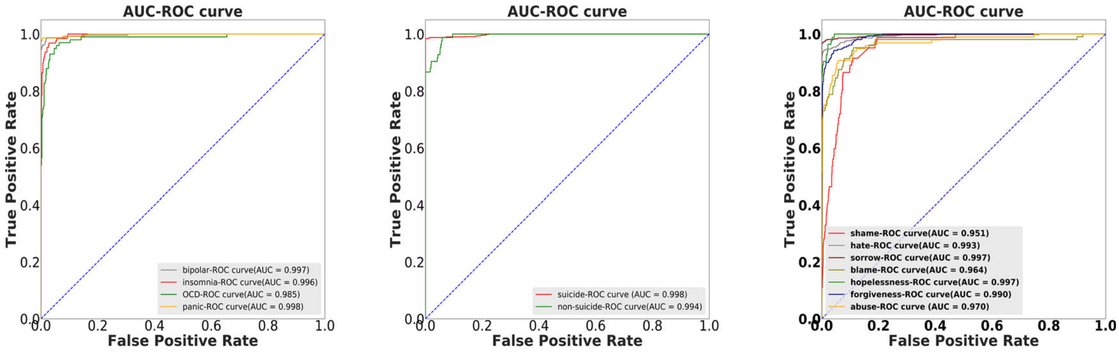
(a) (b) (c)

Fig. 5 Accuracy scores **a** Clinical, **b**, Reddit, **c** CEASE-v2.0



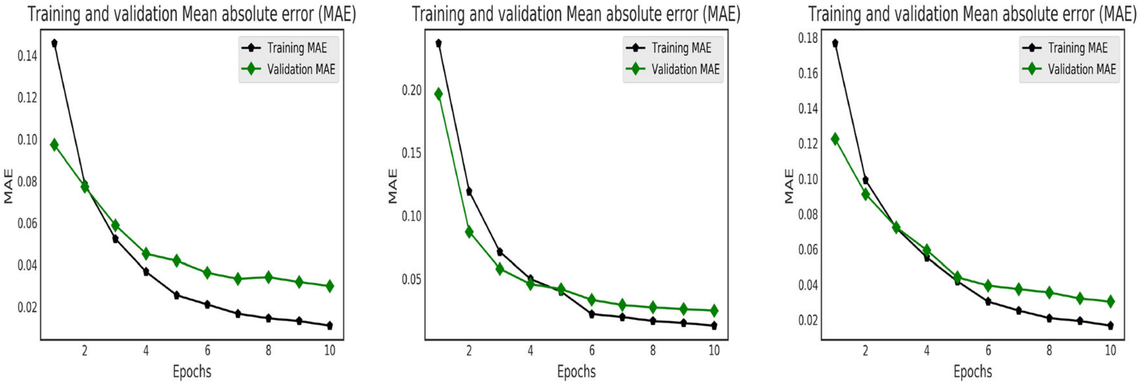
(a) (b) (c)

Fig. 6 Loss function **a** Clinical, **b** Reddit, **c** CEASE-v2.0



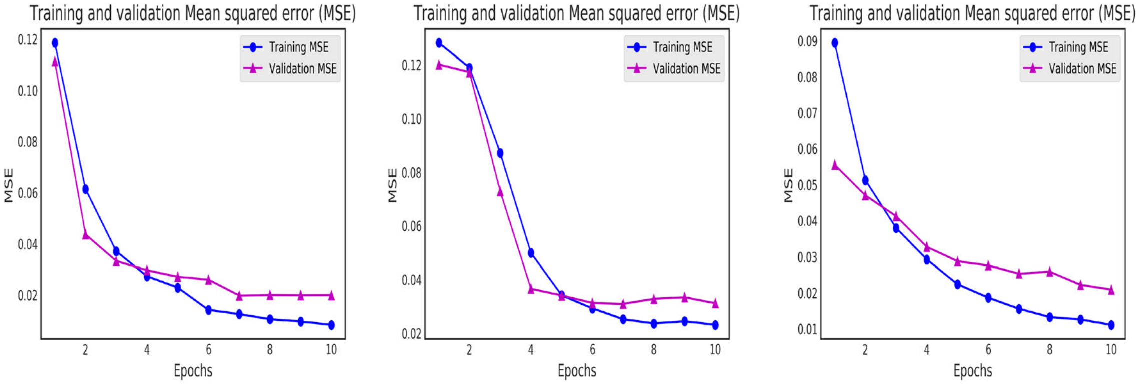
(a) (b) (c)

Fig. 7 AUC-ROC curves **a** Clinical, **b** Reddit, **c** CEASE-v2.0



(a) (b) (c)

Fig. 8 MAE scores **a** Clinical, **b** Reddit, **c**) CEASE-v2.0



(a) (b) (c)

Fig. 9 MSE scores **a** Clinical, **b** Reddit, **c** CEASE-v2.0

different classes across multiple tasks. We achieved AUC scores ranging from 0.985 to 0.998 for disorder labels, 0.998 for suicide classification, 0.994 for non-suicide-related classification, and 0.951 to 0.997 for emotion detection. Notably, the best AUC scores were observed for panic disorder, suicide classification, sorrow, and hopelessness emotion. The improved ROC curves reflect these advancements, demonstrating that the model maintains a high true positive rate while keeping the false-positive rate low across various classification tasks.

Figure 8 illustrates the MAE for evaluating the precision and accuracy of predictions within the MTL framework. Enhanced performance is attributed to the use of cross-entropy loss combined with the Adam optimizer. Our MAE scores are 0.30, 0.24, and 0.22 for the three datasets. The improvement in MAE with each epoch can be attributed to the effective optimization and regularization strategies employed. The Adam optimizer adjusts the learning rates dynamically, enabling the model to converge more quickly and accurately. Additionally, cross-entropy loss, being well-suited for classification tasks, facilitates precise error correction by penalizing incorrect predictions more heavily. As training progresses, the model fine-tunes its parameters, reducing prediction errors and resulting in progressively lower MAE values.

Figure 9 shows the MSE, which measures the models' ability to capture subtle variations in numerical values and evaluate prediction accuracy across tasks. The MSE scores are 0.37, 0.33, and 0.24 for the three datasets. Improved performance with each epoch is due to the models refining their weights and biases over time. Effective regularization and precise parameter tuning help prevent overfitting, allowing the models to generalize better and reduce prediction errors as training progresses. This iterative improvement leads to lower MSE scores and better overall performance.

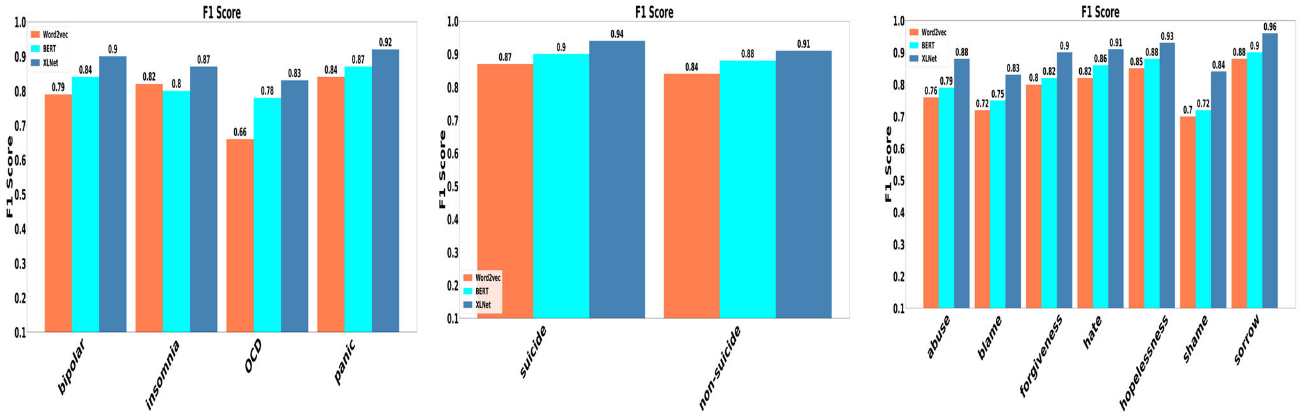
Figure 10 shows the F1 scores for each class label, demonstrating the model's capability to balance false positives and false negatives. We achieved F1 scores above 83% for all class labels. Figure 11 displays the precision for each class label, highlighting the model's accuracy in predicting positive instances across multiple tasks, with precision scores exceeding 86%. Figure 12 illustrates the recall scores for each class label, with recall scores above 81%, showing the model's effectiveness in identifying positive instances. In Tables 2, 5, 6, and 7, the "F," "A," "P," and "R" represent the test scores for "F1 measure, accuracy, precision, and recall," respectively. The best results are highlighted in Tables 2, 3, and 5, 6. The proposed models using XLNet demonstrated superior performance compared to those using Word2vec and BERT.

5.4 Comparison of proposed models with existing models

We conducted various experiments to evaluate the performance of the proposed models compared to existing methods. Tables 2 and 5 present comparisons between baseline models and state-of-the-art methods. Table 2 includes results from baseline deep learning and transformer models tested on various pre-training models across three datasets. To evaluate the advantages of MTL over single-task learning, we also included single-task learning experiments in both the baseline and state-of-the-art evaluations.

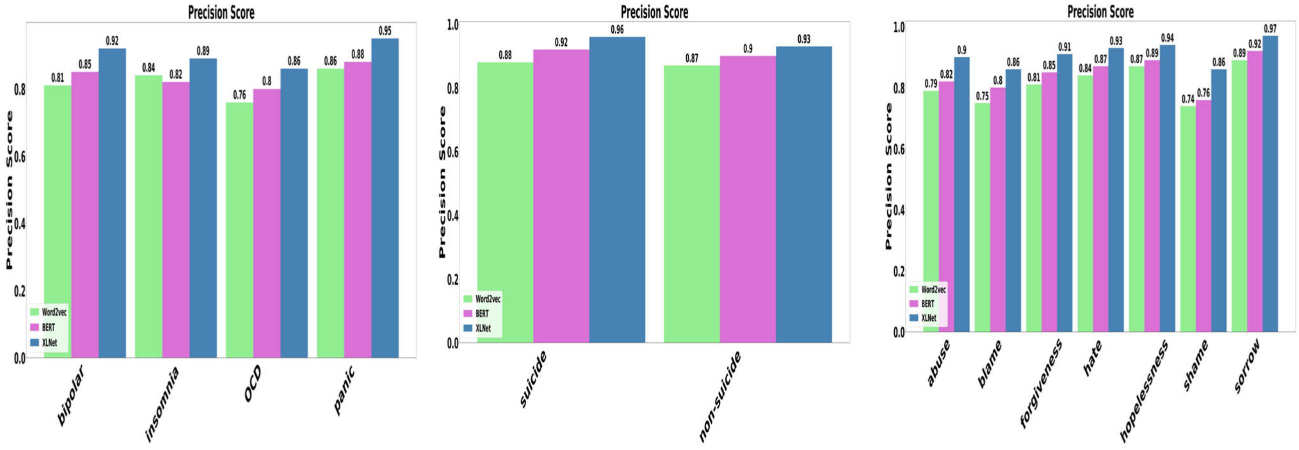
Our proposed models, which utilize the MTL framework, outperform existing models significantly. Specifically, they achieve over 96% accuracy, more than 93.1% F1 scores, over 94% precision, and more than 93% recall. The state-of-the-art models compared in this study are detailed as follows:

- *MT-DNN* The MTL with the deep neural network-based method was suggested in [36] to perform multiple tasks together. They used the task-specific and shared layer for MTL.
- *Fine-tuning MT-BERT* The work in [37] suggested the fine-tuning MT-BERT model for biomedical and clinical tasks. The task-specific fine-tuning approaches were suggested for MTL.
- *SMNL* To identify sentiments as the primary task, [63] proposed a shared MTL network model incorporating an attention fusion mechanism.



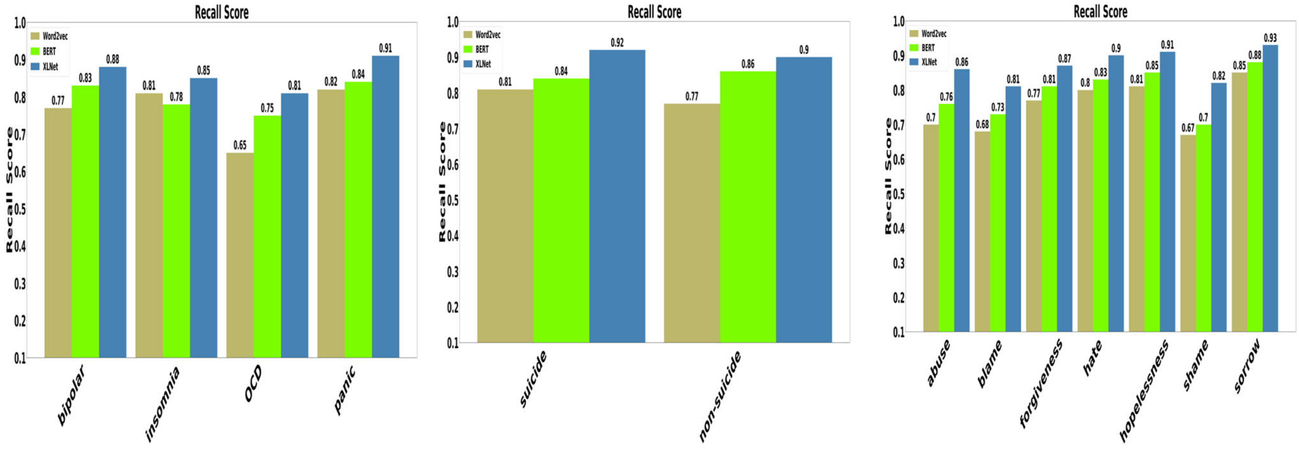
(a) (b) (c)

Fig. 10 Class label-wise F1 scores **a** Clinical, **b** Reddit, **c** CEASE-v2.0



(a) (b) (c)

Fig. 11 Class label-wise precision scores **a** Clinical, **b** Reddit, **c** CEASE-v2.0



(a) (b) (c)

Fig. 12 Class label-wise recall scores **a** Clinical, **b** Reddit, **c** CEASE-v2.0

- *Multi-head MTL* The work in [47] conducts MTL to focus on hate-related context detection. They suggested the MHA with the BERT model to perform MTL.
- *TransformerRNN* In [1], the suggested TransformerRNN model was introduced to extract contextual and long-textual information. This model was explicitly employed for the crucial task of detecting emotions in suicide-related texts.
- *MHA-BCNN* To extract emotions from mental health-related texts, [60] introduced the MHA-BCNN model, which leverages deep learning and attention mechanisms to preserve semantic information.
- *LSTM-Attention-CNN* The LSTM-Attention-CNN model was recommended to perform emotion classification on suicide-related texts [2].
- *C-BiGRU-MHA-CNN and L-BiLSTM-MHA-CNN* In [5], various models were introduced to capture both lexical and contextual data in text sequences. These models were specifically designed for the task of identifying behaviors in suicide-related texts.
- *Single-layered BiLSTM* For emotion detection in textual data, [64] introduced a model single-layered BiLSTM.

In addition, the superior performance is attributed to MTL's ability to leverage shared representations and learn complementary features across multiple tasks, improving generalization and robustness. Additionally, our models effectively capture bidirectional context and manage long-range dependencies, which further boosts their performance compared to traditional methods.

Table 2 Performance comparison of our proposed models with existing models

Model	Pre-training	Clinical	Reddit	CEASE-v2.0	Performance
CNN	Word2vec	0.836	0.838	0.847	F
		0.866	0.874	0.883	A
		0.842	0.845	0.852	P
		0.826	0.838	0.847	R
	BERT	0.842	0.850	0.853	F
		0.870	0.879	0.887	A
		0.844	0.849	0.856	P
		0.831	0.840	0.850	R
	XLNet	0.842	0.849	0.860	F
		0.876	0.881	0.890	A
		0.854	0.857	0.864	P
		0.833	0.852	0.857	R
RNN	Word2vec	0.830	0.834	0.841	F
		0.862	0.872	0.880	A
		0.838	0.841	0.847	P
		0.822	0.834	0.840	R
	BERT	0.835	0.844	0.849	F
		0.866	0.874	0.882	A
		0.840	0.843	0.852	P
		0.825	0.837	0.848	R
	XLNet	0.832	0.844	0.858	F
		0.871	0.876	0.885	A
		0.850	0.854	0.860	P
		0.830	0.848	0.855	R
LSTM	Word2vec	0.857	0.861	0.863	F
		0.877	0.880	0.892	A
		0.855	0.859	0.864	P
		0.835	0.848	0.857	R
	BERT	0.849	0.858	0.865	F
		0.881	0.888	0.896	A
		0.857	0.859	0.866	P
		0.837	0.852	0.860	R
	XLNet	0.850	0.862	0.870	F
		0.885	0.890	0.898	A
		0.861	0.864	0.870	P
		0.841	0.855	0.864	R
GRU	Word2vec	0.857	0.869	0.874	F
		0.890	0.894	0.902	A
		0.867	0.871	0.875	P
		0.847	0.861	0.870	R
	BERT	0.859	0.870	0.876	F
		0.893	0.898	0.904	A
		0.869	0.873	0.879	P
		0.850	0.863	0.871	R
	XLNet	0.861	0.872	0.878	F
		0.895	0.900	0.906	A
		0.870	0.875	0.880	P
		0.851	0.865	0.872	R

Table 2 (continued)

Model	Pre-training	Clinical	Reddit	CEASE-v2.0	Performance
RNN-CNN	Word2vec	0.842	0.854	0.857	F
		0.871	0.876	0.888	A
		0.850	0.851	0.858	P
		0.831	0.844	0.854	R
	BERT	0.845	0.856	0.860	F
		0.875	0.884	0.891	A
		0.851	0.853	0.860	P
		0.837	0.844	0.855	R
	XLNet	0.847	0.855	0.865	F
		0.880	0.884	0.894	A
		0.857	0.861	0.867	P
		0.839	0.854	0.860	R
GRU-CNN	Word2vec	0.864	0.875	0.881	F
		0.898	0.902	0.910	A
		0.872	0.878	0.882	P
		0.854	0.867	0.874	R
	BERT	0.867	0.878	0.884	F
		0.900	0.907	0.914	A
		0.875	0.880	0.885	P
		0.859	0.868	0.875	R
	XLNet	0.870	0.880	0.887	F
		0.902	0.911	0.918	A
		0.879	0.882	0.886	P
		0.861	0.870	0.878	R
LSTM-CNN	Word2vec	0.849	0.860	0.864	F
		0.881	0.886	0.895	A
		0.858	0.861	0.867	P
		0.837	0.851	0.860	R
	BERT	0.851	0.863	0.867	F
		0.884	0.890	0.898	A
		0.861	0.863	0.870	P
		0.840	0.854	0.862	R
	XLNet	0.853	0.865	0.871	F
		0.887	0.892	0.900	A
		0.864	0.869	0.873	P
		0.844	0.858	0.867	R
BiGRU	Word2vec	0.890	0.897	0.902	F
		0.922	0.930	0.935	A
		0.900	0.901	0.903	P
		0.885	0.892	0.898	R
	BERT	0.901	0.900	0.904	F
		0.928	0.933	0.937	A
		0.902	0.906	0.905	P
		0.890	0.899	0.900	R
	XLNet	0.902	0.904	0.905	F
		0.930	0.936	0.940	A
		0.904	0.907	0.908	P
		0.894	0.900	0.902	R

Table 2 (continued)

Model	Pre-training	Clinical	Reddit	CEASE-v2.0	Performance
BiLSTM-CNN	Word2vec	0.881	0.890	0.895	F
		0.912	0.921	0.928	A
		0.891	0.894	0.895	P
		0.874	0.883	0.890	R
	BERT	0.885	0.893	0.897	F
		0.915	0.922	0.930	A
		0.894	0.898	0.900	P
		0.876	0.885	0.892	R
	XLNet	0.886	0.895	0.899	F
		0.919	0.925	0.932	A
		0.896	0.900	0.902	P
		0.880	0.888	0.894	R
BiGRU-CNN	Word2vec	0.906	0.908	0.910	F
		0.936	0.940	0.943	A
		0.910	0.915	0.912	P
		0.898	0.903	0.907	R
	BERT	0.909	0.910	0.915	F
		0.942	0.945	0.950	A
		0.914	0.917	0.920	P
		0.902	0.907	0.912	R
	XLNet	0.910	0.913	0.917	F
		0.944	0.949	0.953	A
		0.918	0.920	0.922	P
		0.904	0.910	0.915	R
Attention mechanism	Word2vec	0.912	0.915	0.919	F
		0.947	0.952	0.955	A
		0.920	0.922	0.924	P
		0.907	0.912	0.918	R
	BERT	0.914	0.918	0.921	F
		0.950	0.953	0.957	A
		0.922	0.925	0.928	P
		0.910	0.918	0.921	R
	XLNet	0.917	0.921	0.923	F
		0.951	0.956	0.959	A
		0.925	0.928	0.930	P
		0.912	0.920	0.925	R
BiLSTM	Word2vec	0.872	0.882	0.890	F
		0.905	0.915	0.920	A
		0.881	0.885	0.888	P
		0.864	0.872	0.880	R
	BERT	0.874	0.885	0.891	F
		0.909	0.917	0.922	A
		0.886	0.888	0.890	P
		0.868	0.875	0.883	R
	XLNet	0.878	0.888	0.893	F
		0.910	0.919	0.925	A
		0.889	0.890	0.892	P
		0.870	0.879	0.886	R

Table 2 (continued)

Model	Pre-training	Clinical	Reddit	CEASE-v2.0	Performance
Attention-CNN	Word2vec	0.919	0.922	0.925	F
		0.952	0.957	0.960	A
		0.927	0.930	0.932	P
		0.916	0.924	0.928	R
	BERT	0.924	0.926	0.928	F
		0.955	0.960	0.962	A
		0.928	0.932	0.935	P
		0.919	0.926	0.929	R
	XLNet	0.926	0.929	0.930	F
		0.958	0.961	0.965	A
		0.931	0.933	0.937	P
		0.920	0.928	0.931	R
SPS-based MTL	Word2vec	0.928	0.931	0.934	F
		0.960	0.964	0.969	A
		0.933	0.935	0.940	P
		0.922	0.930	0.932	R
	BERT	0.931	0.933	0.936	F
		0.963	0.968	0.971	A
		0.937	0.939	0.942	P
		0.925	0.931	0.934	R
	XLNet	**0.938**	**0.940**	**0.946**	F
		0.969	**0.974**	**0.980**	A
		0.942	**0.945**	**0.951**	P
		0.933	**0.937**	**0.940**	R

Bold values represent the best results

5.5 Macro-F1 score analysis and statistical significance testing

The macro-F1 measure and statistical tests are conducted to evaluate the overall performance of our model across all classes and to assess the statistical significance of our results. The macro-F1 measure provides a comprehensive evaluation by averaging the F1 scores for each class, ensuring that the performance is balanced and not skewed by class imbalances. Table 3 shows 84.1%, 84.6%, and 85.8% of macro-F1 scores using clinical, Reddit, CEASE-v2.0 datasets, respectively. Furthermore, a statistical test is performed to assess the statistical significance of our models. The t test is conducted for the proposed models. A statistical method employed to compare the means of two distinct groups has been applied to determine if a notable difference exists between the means of these independent data samples and to elucidate their association. The corresponding p values are provided in Table 4. It is important to note that the t test has been conducted with unequal sample sizes, specifically varying training and testing sample sizes.

5.6 Additional experimental evaluations

We expanded our comparison studies to evaluate the proposed model's performance by comparing it with recent models (*Table IV*, Supplementary material). Moreover, we performed additional MTL experiments to evaluate the proposed models' performance when introduced to new datasets and compare them to state-of-the-art models (*Table V*, Supplementary material). The results demonstrate that the suggested models consistently outperform state-of-the-art models on these new datasets.

5.7 Discussions

In this study, we focused on emotion detection as the primary task, along with auxiliary tasks, systematically evaluating our proposed models against deep learning and transformer-based models. Our models outperformed state-of-the-art methods. Figures 5 and 6 highlight how parameter adjustments enabled capturing bidirectional contextual features and long-term dependencies. Consequently, our models exhibited improved performance metrics. Baseline models struggled with optimizing features for auxiliary

Table 3 Comparison of macro-F1 scores

Model	Pre-training	Clinical	Reddit	CEASE-v2.0
BiLSTM	Word2vec	0.770	0.802	0.811
	BERT	0.794	0.800	0.818
	XLNet	0.789	0.803	0.820
BiGRU	Word2vec	0.794	0.802	0.811
	BERT	0.806	0.821	0.820
	XLNet	0.810	0.824	0.827
BiLSTM-CNN	Word2vec	0.819	0.830	0.836
	BERT	0.827	0.833	0.841
	XLNet	0.830	0.832	0.844
Attention-CNN	Word2vec	0.821	0.825	0.837
	BERT	0.825	0.832	0.848
	XLNet	0.836	0.840	0.854
Multi-head attention	Word2vec	0.818	0.822	0.835
	BERT	0.827	0.830	0.844
	XLNet	0.838	0.837	0.850
Attention BiLSTM	Word2vec	0.819	0.825	0.833
	BERT	0.824	0.836	0.840
	XLNet	0.836	0.837	0.849
GRU-CNN	Word2vec	0.790	0.810	0.815
	BERT	0.812	0.810	0.818
	XLNet	0.808	0.820	0.829
BiGRU-CNN	Word2vec	0.800	0.805	0.816
	BERT	0.809	0.822	0.824
	XLNet	0.812	0.826	0.830
Attention LSTM	Word2vec	0.816	0.823	0.830
	BERT	0.825	0.832	0.842
	XLNet	0.833	0.836	0.842
Self-attention	Word2vec	0.824	0.826	0.832
	BERT	0.825	0.828	0.846
	XLNet	0.835	0.833	0.852
SPS-based MTL	Word2vec	0.828	0.830	0.841
	BERT	0.832	0.838	0.852
	XLNet	**0.841**	**0.846**	**0.858**

Bold values represent the best results

Table 4 Statistical tests using sample size

Test/validation size	Clinical	Reddit	CEASE-v2.0
25 %, 30%	**0.016**	**0.012**	**0.010**
25 %, 40%	0.020	0.026	0.024
25 %, 50%	0.030	0.034	0.028
30 %, 40%	0.023	0.020	0.017
30 %, 50%	0.032	0.030	0.033
40 %, 50%	0.044	0.047	0.042

Bold values represent the best results

tasks. Our models, with shared encoders and attention mechanisms, leveraged pre-trained contextual embeddings for better feature extraction. The SPS-BiGRU-SAM processed input sequences bidirectionally, while SPS-LSTM-AM focused on key sequence parts, and SPS-BNN-MHAM used multi-head attention for parallel processing. Test F1 scores were 93.82%, 94.40%, and 94.65%, with corresponding precision and recall for SPS-LSTM-AM, SPS-BiGRU-SAM, and SPS-BNN-MHAM across clinical, Reddit, and CEASE-v2.0 datasets. Figures 7 to 12 show that most patients' texts contain panic disorder and more suicide-related content on Reddit. However, sorrow and hopelessness are the most dominant emotions in suicide notes. The term “SPS-based MTL” in Tables 2, 3, 5, and 6 refers to our proposed models.

The primary innovation of this work is the extensive experimental validation of proposed models for mental health analysis. Our experiments include comparative analyses with baseline and state-of-the-art models across tasks like disorder detection, suicide classification, and emotion detection (Tables 2 and 5). The models achieved accuracies of 96.9%, 97.4%, and 98%, with F1 scores, precision, and recall all above 93%, demonstrating their ability to capture emotional and psychological patterns in various texts. A macro-F1 score comparison yielded 84.1%, 84.6%, and 85.8%, confirming the models' superior performance (Table 5). Ablation tests validated the importance of each component in the MTL framework. In addition, we developed models based on SPS architecture with subtypes to manage mental health texts' complexity. These models allow selective parameter sharing, balancing shared representations, and task-specific fine-tuning. Using auto-regressive-based permutation techniques and shared embedding representations, our models capture semantic nuances and maintain long-term dependencies with transformers. The architecture supports both shared and task-specific layers, enhancing learning efficiency and the models' ability to analyze diverse mental health data.

5.8 Ablation experiments

We performed ablation tests to identify the essential components contributing to the enhancement of the proposed models' performance (Tables 6 and 7). First, the S-ARLM component is suggested for sharing the common encoder layer. Embedding representations from all texts are stored in the above component. Then, the encoder training is passed to I-BiGRU, I-LSTM, and I-BiLSTM components. Here, “I” stands for independent layer. The above components maintain contextual features individually. Here, the attention mechanism (AM) contains input vectors from both self-attention (SA) and MHA. It shares both SA and MHA for other tasks. If a particular input vector is

Table 5 Comparing the suggested models' performance with state-of-the-art models

Model	Clinical	Reddit	CEASE-v2.0	Performance
MT-DNN [36]	0.913	0.920	0.922	A
	0.895	0.893	0.896	F
	0.898	0.901	0.900	P
	0.890	0.891	0.895	R
L-BiLSTM-MHA-CNN [5]	0.958	0.963	0.972	A
	0.920	0.930	0.931	F
	0.927	0.935	0.933	P
	0.920	0.928	0.927	R
Fine-tuning MT-BERT [37]	0.906	0.914	0.935	A
	0.887	0.892	0.909	F
	0.893	0.905	0.912	P
	0.885	0.887	0.907	R
MHA-BCNN [60]	0.959	0.960	0.968	A
	0.917	0.925	0.933	F
	0.922	0.931	0.936	P
	0.915	0.920	0.924	R
Shared MTL network [63]	0.950	0.960	0.962	A
	0.925	0.922	0.932	F
	0.929	0.927	0.935	P
	0.923	0.920	0.930	R
Multi-head MTL [47]	0.941	0.955	0.948	A
	0.920	0.923	0.930	F
	0.925	0.927	0.934	P
	0.918	0.919	0.922	R
C-BiGRU-MHA-CNN [5]	0.960	0.965	0.970	A
	0.926	0.932	0.935	F
	0.936	0.934	0.938	P
	0.924	0.930	0.931	R
Single-layered BiLSTM [64]	0.955	0.961	0.966	A
	0.919	0.927	0.928	F
	0.922	0.929	0.935	P
	0.917	0.925	0.927	R
LSTM-Attention-CNN [2]	0.957	0.959	0.963	A
	0.920	0.919	0.932	F
	0.926	0.920	0.937	P
	0.920	0.916	0.928	R
TransformerRNN [1]	0.938	0.950	0.954	A
	0.918	0.920	0.932	F
	0.920	0.923	0.936	P
	0.912	0.915	0.920	R
SPS-based MTL	**0.969**	**0.974**	**0.980**	A
	0.938	**0.940**	**0.946**	F
	0.942	**0.945**	**0.951**	P
	0.933	**0.937**	**0.940**	R

Bold values represent the best results

essential, it will be shared with the remaining tasks. Attention hidden values are further passed to the I-Conv components. In this context, we reduce higher-dimensional values using I-Conv and the fully shared (FS) layer. This

Table 6 Ablation study of specific components in proposed models

Component/model	Clinical	Reddit	CEASE-v2.0	Performance
S-ARLM	0.905	0.908	0.922	F
	0.938	0.942	0.945	A
	0.912	0.916	0.928	P
	0.905	0.910	0.919	R
I-BiGRU	0.915	0.917	0.930	F
	0.944	0.951	0.955	A
	0.918	0.920	0.931	P
	0.912	0.915	0.923	R
I-LSTM	0.909	0.911	0.926	F
	0.941	0.946	0.950	A
	0.915	0.918	0.930	P
	0.908	0.912	0.921	R
I-BiLSTM	0.918	0.920	0.932	F
	0.946	0.953	0.957	A
	0.923	0.925	0.934	P
	0.916	0.918	0.925	R
PS-SA	0.921	0.924	0.935	F
	0.950	0.958	0.960	A
	0.927	0.929	0.937	P
	0.919	0.922	0.928	R
Complete-AM	0.928	0.931	0.940	F
	0.955	0.964	0.969	A
	0.935	0.938	0.944	P
	0.924	0.930	0.932	R
PS-MHA	0.925	0.929	0.937	F
	0.952	0.960	0.964	A
	0.932	0.933	0.941	P
	0.921	0.925	0.930	R
I-Conv	0.904	0.907	0.918	F
	0.926	0.934	0.940	A
	0.910	0.914	0.925	P
	0.903	0.906	0.914	R
I-GMP	0.884	0.893	0.898	F
	0.912	0.918	0.922	A
	0.895	0.900	0.904	P
	0.871	0.890	0.895	R
SPS-based MTL	**0.938**	**0.940**	**0.946**	F
	0.969	**0.974**	**0.980**	A
	0.942	**0.945**	**0.951**	P
	0.933	**0.937**	**0.940**	R

Bold values represent the best results

process incorporates convolution features and is denoted as FS-GMP. Fully connected softmax (FCS) is maintained individually to get multiple outcomes.

We perform a component-wise ablation test to comprehend the roles of various components within each model for a specific task, as depicted in Table 6. Subsequently, we aggregate all components regardless of the model and systematically eliminate each component to check whether the overall performance improves when comparing with the proposed models (see Table 7). Additional ablation tests, including task ablation, model ablation, and feature ablation, are performed and detailed in the supplementary material (Tables I, II, and III). In task-specific ablation, we selectively remove one particular task for each model to assess the impact on performance. For instance, in the SPS-LSTM-AM model, the disorder detection task is removed, and the performance is evaluated for the remaining tasks. Here, "-" denotes the removal of a specific task for a particular model. In model-specific ablation, we eliminate one model (e.g., SPS-LSTM-AM) and reassess the performance of the remaining models (SPS-BiGRU-SAM and SPS-BNN-MHAM). In this context, "−" signifies the removal of a specific model for a particular dataset. For feature-specific ablation, we exclude a particular feature or component for a given model to evaluate its performance across all tasks. These diverse ablation studies underscore the significance of each component within our proposed models. Notably, removing any of these components results in a discernible decrease in the models' performance. However, it is crucial to acknowledge that individual components cannot match the overall performance of the complete model.

While the proposed models perform well on different datasets, there are some limitations that could be addressed in future work. One limitation is the need to enhance model explainability by identifying and understanding the features most crucial for the tasks, which would improve interpretability and decision-making clarity. Also, optimizing the integration of auxiliary tasks is to ensure a balanced contribution between primary and secondary objectives, thereby minimizing potential task interference. Additionally, expanding the models' adaptability to new or emerging mental health conditions presents a valuable opportunity for future research, ensuring that the models remain effective and relevant as new challenges arise. Addressing these aspects could further advance the models' capabilities and their application in diverse mental health contexts.

6 Conclusion and future scope

This research introduces multiple optimization algorithms to minimize discrepancies in handling various tasks. Our main objective is to gain a comprehensive understanding of the interrelated outcomes and facilitate the sharing of contextual features. Diverse MTL techniques are devised to gain insights into individuals' mental health. These initiatives focus on proactively identifying mental health

Table 7 Ablation study for analyzing key components in our models

Component	Clinical	Reddit	CEASE-v2.0	Performance
I-BiGRU+ I-LSTM+ I-BiLSTM+ PS-SA+ Complete-AM+ PS-MHA+ I-Conv+ I-GMP ablates S-ARLM	0.924	0.928	0.930	F
	0.959	0.962	0.968	A
	0.928	0.931	0.943	P
	0.921	0.924	0.928	R
S-ARLM+ I-LSTM+ I-BiLSTM+ PS-SA+ Complete-AM+ PS-MHA+ I-Conv+ I-GMP ablates I-BiGRU	0.918	0.923	0.925	F
	0.954	0.958	0.964	A
	0.922	0.927	0.938	P
	0.914	0.918	0.920	R
S-ARLM+ I-BiGRU + I-BiLSTM+ PS-SA+ Complete-AM+ PS-MHA+ I-Conv+ I-GMP ablates I-LSTM	0.921	0.926	0.928	F
	0.957	0.960	0.965	A
	0.924	0.929	0.940	P
	0.917	0.922	0.925	R
S-ARLM+ I-BiGRU+ I-LSTM + PS-SA+ Complete-AM+ PS-MHA+ I-Conv+ I-GMP ablates I-BiLSTM	0.915	0.920	0.923	F
	0.950	0.955	0.962	A
	0.921	0.924	0.936	P
	0.912	0.915	0.917	R
S-ARLM+ I-BiGRU+ I-LSTM+ I-BiLSTM+ Complete-AM+ PS-MHA+ I-Conv+ I-GMP ablates PS-SA	0.911	0.916	0.920	F
	0.948	0.952	0.960	A
	0.917	0.921	0.933	P
	0.910	0.912	0.914	R
S-ARLM+ I-BiGRU+ I-LSTM+ I-BiLSTM+ PS-SA+ PS-MHA+ I-Conv+ I-GMP ablates Complete-AM	0.900	0.904	0.906	F
	0.921	0.929	0.940	A
	0.908	0.911	0.916	P
	0.897	0.902	0.904	R
S-ARLM+ I-BiGRU+ I-LSTM+ I-BiLSTM+ PS-SA+ Complete-AM+ I-Conv+ I-GMP ablates PS-MHA	0.906	0.912	0.915	F
	0.943	0.950	0.957	A
	0.914	0.917	0.930	P
	0.905	0.907	0.911	R
S-ARLM+ I-BiGRU+ I-LSTM+ I-BiLSTM+ PS-SA+ Complete-AM+ PS-MHA+ I-GMP ablates I-Conv	0.929	0.930	0.933	F
	0.961	0.967	0.971	A
	0.936	0.934	0.945	P
	0.926	0.928	0.933	R
S-ARLM+ I-BiGRU+ I-LSTM+ I-BiLSTM+ PS-SA+ Complete-AM+ PS-MHA+ I-Conv ablates I-GMP	0.932	0.935	0.939	F
	0.965	0.969	0.973	A
	0.939	0.936	0.948	P
	0.929	0.930	0.936	R

behaviors through transformer-based deep learning. Employing a loose coupling approach preserves the uniqueness of each task, facilitates the sharing of essential features, and eliminates rigid constraints. Notably, the primary task exhibits superior performance compared to the auxiliary tasks. We introduce three new models for MTL, following the SPS rules. SPS emphasizes maintaining task individuality while sharing necessary input features, thus mitigating issues and enhancing model performance. The SPS-LSTM-AM model is suggested to detect disorders like bipolar, insomnia, OCD, and panic in clinical data related to psychiatric texts. The SPS-BiGRU-SAM model is suggested for suicide or non-suicide-related classification using Reddit data. The SPS-BNN-MHAM model is proposed for emotion detection, such as abuse, blame, forgiveness, hate, hopelessness, shame, and sorrow in suicide notes. The suggested models achieved 96.94%, 97.40%, and 98.08% of test accuracies over existing models.

Furthermore, the initial model incorporates LSTM-based transformers featuring a full attention mechanism. The following approach utilizes a bidirectional gated neural network with a partially shared attention mechanism resembling a self-attention component. The third method employs an MHA component in conjunction with a bidirectional neural network. In all cases, a fully shared embedding representation is assumed for constructing the encoder. The methodology follows an auto-regressive procedure to ensure attention is maintained at every input position. The overarching goal is to preserve long-term dependencies using transformer-based techniques. The objective is to share crucial semantic, nuance-dependent, and contextual features in variable-length text data using transformer-based deep learning. Through this use of distinct models with SPS, the strategy ensures seamless knowledge transfer between tasks, allowing for comprehensive capture of textual intricacies and context appropriateness. This collaborative learning approach optimizes the effectiveness of each task while ensuring the preservation of critical features, resulting in a nuanced and comprehensive understanding of textual data. For future work directions, we investigate additional tasks or dimensions related to mental health, such as resilience detection or changes in behavior over time. Explore the integration of other modalities, such as voice or image analysis, for a more thorough understanding of a person's mental health. Integrate explainability mechanisms into MTL models to highlight important features and factors contributing to mental health predictions.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00521-024-10753-7.

Authors' contributions Dheeraj Kodati is the principal contributor of this paper. Ramakrishnudu Tene cleared some of the issues emerged during this manuscript preparation

Funding None.

Data availability All the datasets used in this research are publicly available, and we have included the corresponding references.

Declarations

Conflict of interest The authors declare that there are no conflict of interest in this work.

Ethical approval None.

References

1. Zhang T, Schoene AM, Ananiadou S (2021) Automatic identification of suicide notes with a transformer-based deep learning model. Internet Interview 25:100422. https://doi.org/10.1016/j.invent.2021.100422
2. Renjith S, Abraham A, Jyothi SB, Chandran L, Thomson J (2022) An ensemble deep learning technique for detecting suicidal ideation from posts in social media platforms. J King Saud Univ Comput Inf Sci 34:9564–9575. https://doi.org/10.1016/j.jksuci.2021.11.010
3. Cohan A, Desmet B, Yates A, Soldaini L, MacAvaney S, Goharian N (2018) SMHD: a large-scale resource for exploring online language usage for multiple mental health conditions. In: Proceedings of the 27th international conference on computational linguistics, pp 1485–1497. Association for Computational Linguistics. https://aclanthology.org/C18-1126
4. Fei Z, Yang E, Li D, Butler S, Ijomah W, Li X, Zhou H (2020) Deep convolution network based emotion analysis towards mental health care. Neurocomputing 388:212–227. https://doi.org/10.1016/j.neucom.2020.01.034
5. Kodati D, Ramakrishnudu T (2022) Identifying suicidal emotions on social media through transformer-based deep learning. Appl Intell 53:11885–11917. https://doi.org/10.1007/s10489-022-04060-8
6. Wu J, He Y, Yu L, Lai KR (2020) Identifying emotion labels from psychiatric social texts using a bi-directional LSTM-CNN model. IEEE Access 8:66638–66646. https://doi.org/10.1109/ACCESS.2020.2985228
7. Kumari R, Ashok N, Ghosal T, Ekbal A (2021) Misinformation detection using multitask learning with mutual learning for novelty detection and emotion recognition. Inf Process Manag 58:102631. https://doi.org/10.1016/j.ipm.2021.102631
8. Akhtar MS, Chauhan D, Ghosal D, Poria S, Ekbal A, Bhattacharyya P (2019) Multi-task learning for multi-modal emotion recognition and sentiment analysis. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, Volume 1, pp 370–379. https://doi.org/10.18653/v1/N19-1034
9. Khare A, Parthasarathy S, Sundaram S (2020) Multi-modal embeddings using multi-task learning for emotion recognition. In: Proc. Interspeech 2020, pp 384–388. https://doi.org/10.21437/Interspeech.2020-1827
10. Desmet B, Hoste V (2013) Emotion detection in suicide notes. Expert Syst Appl 40:6351–6358. https://doi.org/10.1016/j.eswa.2013.05.050
11. Larsen ME, Boonstra TW, Batterham PJ, O'Dea B, Paris C, Christensen H (2015) We feel: mapping emotion on twitter. IEEE J Biomed Health Inform 19:1246–1252. https://doi.org/10.1109/JBHI.2015.2403839
12. Ji S, Yu C, Fung S-F, Pan S, Long G (2018) Supervised learning for suicidal ideation detection in online user content. Complexity 2018:1–10. https://doi.org/10.1155/2018/6157249
13. Wu M, Shen C-Y, Wang ET, Chen A (2020) A deep architecture for depression detection using posting, behavior, and living environment data. J Intell Inf Syst 54:225–244. https://doi.org/10.1007/s10844-018-0533-4
14. Gao K, Xu H, Gao C, Hao H, Deng J, Sun X (2018) Attention-based bilstm network with lexical feature for emotion classification. In: 2018 international joint conference on neural networks (IJCNN), pp 1–2 . https://doi.org/10.1109/IJCNN.2018.8489577
15. Chatterjee A, Gupta U, Chinnakotla M, Srikanth R, Galley M, Agrawal P (2018) Understanding emotions in text using deep learning and big data. Comput Hum Behav 93:309–317. https://doi.org/10.1016/j.chb.2018.12.029

16. Bhat HS, Goldman-Mellor SJ (2017) Predicting adolescent suicide attempts with neural networks. arXiv . https://doi.org/10.48550/arXiv.1711.10057
17. Song H, You J, Chung J-W, Park JC (2018) Feature attention network: Interpretable depression detection from social media. In: Proceedings of the 32nd Pacific Asia conference on language, information and computation. Association for Computational Linguistics. https://aclanthology.org/Y18-1070
18. Xia C, Zhao D, Wang J, Liu J, Ma J (2018) Icsh 2018: Lstm based sentiment analysis for patient experience narratives in e-survey tools. In: International conference on smart health, pp 231–239. https://doi.org/10.1007/978-3-030-03649-2_23
19. Cong Q, Feng Z, Li F, Xiang Y, Rao G, Tao C (2018) X-a-bilstm: a deep learning approach for depression detection in imbalanced data. In: 2018 IEEE international conference on bioinformatics and biomedicine (BIBM), pp 1624–1627. https://doi.org/10.1109/BIBM.2018.8621230
20. Chakraborty K, Bhatia S, Bhattacharyya S, Platos J, Bag R, Hassanien AE (2020) Sentiment analysis of COVID-19 tweets by deep learning classifiers-a study to show how popularity is affecting accuracy in social media. Appl Soft Comput 97:106754. https://doi.org/10.1016/j.asoc.2020.106754
21. Qaqish E, Aranki A, Etaiwi W (2023) Sentiment analysis and emotion detection of post-COVID educational tweets: Jordan case. Soc Netw Anal Min 13:39. https://doi.org/10.1007/s13278-023-01041-8
22. Chen Q, Zhao G, Wu Y, Qian X (2023) Fine-grained semantic textual similarity measurement via a feature separation network. Appl Intell 53:18205–18218. https://doi.org/10.1007/s10489-022-04448-6
23. Devlin J, Chang M-W, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies, Volume 1, pp 4171–4186. https://doi.org/10.18653/v1/N19-1423
24. Mahto D, Yadav SC (2023) Emotion prediction for textual data using GloVe based HeBi-CuDNNLSTM model. Multimedi Tools Appl. https://doi.org/10.1007/s11042-023-16062-w
25. Yan H, Li H, Yi B (2023) Multi-channel convolutional neural network with sentiment information for sentiment classification. Arab J Sci Eng 48:10551–10561. https://doi.org/10.1007/s13369-023-07695-y
26. Bahgat M, Wilson S, Magdy W (2020) Towards using word embedding vector space for better cohort analysis. In: Proceedings of the international AAAI conference on web and social media 14:919–923. https://doi.org/10.1609/icwsm.v14i1.7358
27. He B, Zhang J (2023) An association rule mining method based on named entity recognition and text classification. Arab J Sci Eng 48:1503–1511. https://doi.org/10.1007/s13369-022-06870-x
28. Kanaparthi SD, Patle A, Naik KJ (2023) Prediction and detection of emotional tone in online social media mental disorder groups using regression and recurrent neural networks. Multimed Tools Appl 82:43819–43839. https://doi.org/10.1007/s11042-023-15316-x
29. Kamran S, Zall R, Hosseini S, Kangavari M, Rahmani S, Hua W (2023) Emodnn: understanding emotions from short texts through a deep neural network ensemble. Neural Comput Appl 35:13565–13582. https://doi.org/10.1007/s00521-023-08435-x
30. Liu P, Qiu X, Huang X (2017) Adversarial multi-task learning for text classification. In: Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 1), pp 1–10. Association for Computational Linguistics, . https://doi.org/10.18653/v1/P17-1001
31. Chen Y, Hou W, Cheng X, Li S (2018) Joint learning for emotion classification and emotion cause detection. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp 646–651. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1066
32. Akhtar S, Ghosal D, Ekbal A, Bhattacharyya P, Kurohashi S (2019) All-in-one: emotion, sentiment and intensity prediction using a multi-task ensemble framework. IEEE Trans Affect Comput 13:285–297. https://doi.org/10.1109/TAFFC.2019.2926724
33. Yang Q, Shang L (2019) Multi-task learning with bidirectional language models for text classification. In: 2019 international joint conference on neural networks (IJCNN), pp 1–8. https://doi.org/10.1109/IJCNN.2019.8852388
34. He R, Lee WS, Ng HT, Dahlmeier D (2019) An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In: Annual meeting of the association for computational linguistics, pp 504–515. https://doi.org/10.18653/v1/P19-1048
35. Pagé Fortin M, Chaib-draa B (2019) Multimodal multitask emotion recognition using images, texts and tags. In: Proceedings of the ACM workshop on crossmodal learning and application, pp 3–10. Association for Computing Machinery. https://doi.org/10.1145/3326459.3329165
36. Liu X, He P, Chen W, Gao J (2019) Multi-task deep neural networks for natural language understanding. arXiv. https://doi.org/10.48550/arXiv.1901.11504
37. Peng Y, Chen Q, Lu Z (2020) An empirical study of multi-task learning on BERT for biomedical text mining. arXiv. https://doi.org/10.48550/arXiv.2005.02799
38. Li J, Zhang M, Ji D, Liu Y (2020) Multi-task learning with auxiliary speaker identification for conversational emotion recognition. arXiv. https://doi.org/10.48550/arXiv.2003.01478
39. Lu G, Gan J, Yin J, Luo Z, Li B, Zhao X (2020) Multi-task learning using a hybrid representation for text classification. Neural Comput Appl 32:6467–6480. https://doi.org/10.1007/s00521-018-3934-y
40. Patel M, Ezeife CI (2021) Bert-based multi-task learning for aspect-based opinion mining. In: Database and expert systems applications, pp 192–204. Springer, Cham. https://doi.org/10.1007/978-3-030-86472-9_18
41. Li Y, Kazameini A, Mehta Y, Cambria E (2021) Multitask learning for emotion and personality detection. Neurocomputing. https://doi.org/10.1016/j.neucom.2022.04.049
42. Marreddy M, Oota SR, Vakada LS, Chinni VC, Mamidi R (2022) Multi-task text classification using graph convolutional networks for large-scale low resource language. In: 2022 international joint conference on neural networks (IJCNN), pp 1–8. https://doi.org/10.1109/IJCNN55064.2022.9892105
43. Li C, Braud C, Amblard M (2022) Multi-task learning for depression detection in dialogs. In: Proceedings of the 23rd annual meeting of the special interest group on discourse and dialogue, pp. 68–75. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.sigdial-1.7
44. Lin N, Fu S, Lin X, Wang L (2022) Multi-label emotion classification based on adversarial multi-task learning. Inf Process Manag 59:103097. https://doi.org/10.1016/j.ipm.2022.103097
45. Cheng P, Dai J, Liu J (2022) Catvrnn: generating category texts via multi-task learning. Knowl-Based Syst 244:108491. https://doi.org/10.1016/j.knosys.2022.108491
46. Gao Q, Cao B, Guan X, Gu T, Bao X, Wu J, Liu B, Cao J (2022) Emotion recognition in conversations with emotion shift detection based on multi-task learning. Knowl-Based Syst 248:108861. https://doi.org/10.1016/j.knosys.2022.108861
47. Plaza-del-Arco FM, Halat S, Padó S, Klinger R (2022) Multi-task learning with sentiment, emotion, and target detection to recognize hate speech and offensive language. https://doi.org/10.48550/arXiv.2109.10255

48. Plaza-Del-Arco FM, González MD, Ureña-López L, Martín-Valdivia M-T (2022) Integrating implicit and explicit linguistic phenomena via multi-task learning for offensive language detection. Knowl-Based Syst 258:109965. https://doi.org/10.1016/j.knosys.2022.109965
49. Tan Y, Chow CO, Kanesan J, Chuah JH, Lim Y (2023) Sentiment analysis and sarcasm detection using deep multi-task learning. Wirel Pers Commun 129:2213–2237. https://doi.org/10.1007/s11277-023-10235-4
50. Luo Y, Wu R, Liu J, Tang X (2023) A text guided multi-task learning network for multimodal sentiment analysis. Neurocomputing 560:126836. https://doi.org/10.1016/j.neucom.2023.126836
51. Cerisara C, Jafaritazehjani S, Oluokun A, Le HT (2018) Multi-task dialog act and sentiment recognition on mastodon. In: Proceedings of the 27th international conference on computational linguistics, pp 745–754. Association for Computational Linguistics. https://aclanthology.org/C18-1063
52. Xu Y, Yao E, Liu C, Liu Q, Xu M (2023) A novel ensemble model with two-stage learning for joint dialog act recognition and sentiment classification. Pattern Recogn Lett 165:77–83. https://doi.org/10.1016/j.patrec.2022.11.032
53. Priya P, Firdaus M, Ekbal A (2023) A multi-task learning framework for politeness and emotion detection in dialogues for mental health counseling and legal aid. Expert Syst Appl 224:120025. https://doi.org/10.1016/j.eswa.2023.120025
54. Ameer I, Bölücü N, Siddiqui MHF, Can B, Sidorov G, Gelbukh A (2023) Multi-label emotion classification in texts using transfer learning. Expert Syst Appl 213:118534. https://doi.org/10.1016/j.eswa.2022.118534
55. Liu B, Chen Q, Xiao Y, Wang K, Liu J, Huang R, Li L (2023) Semi-supervised multi-task learning with auxiliary data. Inf Sci 626:626–639. https://doi.org/10.1016/j.ins.2023.02.091
56. Zhang Y, Wang J, Liu Y, Rong L, Zheng Q, Song D, Tiwari P, Qin J (2023) A multitask learning model for multimodal sarcasm, sentiment and emotion recognition in conversations. Inf Fusion. https://doi.org/10.1016/j.inffus.2023.01.005
57. Rathnayake H, Sumanapala J, Rukshani R, Ranathunga S (2024) Adapterfusion-based multi-task learning for code-mixed and code-switched text classification. Eng Appl Artif Intell 127:107239. https://doi.org/10.1016/j.engappai.2023.107239
58. Dutta A, Biswas S, Das AK (2024) Emocomicnet:a multi-task model for comic emotion recognition. Pattern Recognit. https://doi.org/10.1016/j.patcog.2024.110261
59. Ghosh S, Ekbal A, Bhattacharyya P (2021) A multitask framework to detect depression, sentiment and multi-label emotion from suicide notes. Cogn Comput 14:110–129. https://doi.org/10.1007/s12559-021-09828-7
60. Kodati D, Ramakrishnudu T (2021) Negative emotions detection on online mental-health related patients texts using the deep learning with MHA-BCNN model. Expert Syst Appl 182:115265. https://doi.org/10.1016/j.eswa.2021.115265
61. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV (2019) Xlnet: generalized autoregressive pretraining for language understanding. NeurIPS 32:5753–5763
62. Mikolov T, Chen K, Corrado Gs, Dean J (2013) Efficient estimation of word representations in vector space. In: International conference on learning representations
63. Yao C, Song X, Zhang X, Zhao W, Feng A (2021) Multitask learning for aspect-based sentiment classification. Sci Program 2021:1–9. https://doi.org/10.1155/2021/2055555
64. Hameed Z, Garcia-Zapirain B (2020) Sentiment classification using a single-layered BiLSTM model. IEEE Access 8:73992–74001. https://doi.org/10.1109/ACCESS.2020.2988550

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.