IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 February 2025, accepted 28 February 2025, date of publication 4 March 2025, date of current version 12 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3547813

RESEARCH ARTICLE

Pseudo-Labeling Domain Adaptation Using Multi-Model Learning

VICTOR AKIHITO KAMADA TOMITA AND RICARDO MARCONDES MARCACINI
Instituto de Ciências Matemáticas e de Computação (ICMC), University of São Paulo, São Carlos 13566-590, Brazil

Corresponding author: Victor Akihito Kamada Tomita (akihito012@usp.br)

This work was supported in part by São Paulo Research Foundation (FAPESP) under Grant 2018/15163-6 and Grant 2022/01793-3, in part by the National Council for Scientific and Technological Development (CNPq) under Grant 141010/2018-5 and Grant 316507/2023-7, in part by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior–Brasil (CAPES)–Finance Code 001, and in part by CAPES under Grant 00x0ma614.

ABSTRACT With the constant growth of state-of-the-art models, obtaining sufficient labeled data to train these models for specific domains has become increasingly costly. Domain adaptation methods offer a potential solution to enhance model performance in new, unseen domains while minimizing the need for manual annotation of target domain. Despite recent advances in using pseudo-labeling for domain adaptation, significant challenges remain in maximizing the effectiveness of pseudo-labeling, particularly when aiming to create informative and interpretable representations from pseudo-labels. To address these challenges, we introduce the method *Pseudo-labeling Domain Adaptation* (PDA), which leverages pseudo-labels generated by multiple models to create a robust cross-domain representation. Additionally, to further mitigate the domain-shift problem, we propose a novel method called *UMAP Domain Adaptation* (UMAP DA), a UMAP-based technique that allows for connections only between nodes from different domains. We use these representations to construct a heterogeneous bipartite graph, where a neural network is employed for final classification. Experiments on six different datasets show an average F1-score improvement of 8 points, measuring the harmonic mean of precision and recall, compared to existing methods in the literature. The proposed method enhances both performance and interpretability, offering a new direction for cross-domain learning with pseudo-labels.

INDEX TERMS Domain adaptation, graph neural networks, interpretability, pseudo-labeling.

I. INTRODUCTION

Cross-domain methods aim to facilitate domain adaptation by enabling models trained on data from one domain to perform effectively in a different target domain, thus reducing the costs associated with manual data labeling in the target domain [1], [2], [3]. These techniques have gained increasing importance given the rapid advancement and growing size of models that define the state of the art across various machine learning tasks [4], [5]. As models become more computationally expensive to train, the feasibility of repeatedly training them on large datasets diminishes, making efficient domain adaptation crucial.

One of the most promising strategies to address these challenges is the use of pseudo-labels, which facilitates the labeling of previously unlabeled data by utilizing predictions from pre-trained models [6], [7], [8]. However, maximizing the effectiveness of pseudo-labeling remains a challenge, particularly in obtaining data representations that are not only informative but also interpretable [9], [10], [11]. Achieving clarity in how data points relate across domains is crucial for improving model performance, which is where dimensionality reduction techniques can play a decisive role [12], [13], [14]. These techniques enable a more concise visualization of the data underlying structure, facilitating a deeper understanding of domain relationships and adaptations [15], [16], [17].

To address these challenges, we introduce a novel approach called *Pseudo-labeling Domain Adaptation (PDA)*, designed to generate robust representations based on pseudo-labels generated from multiple models. Furthermore, we propose a complementary technique, *UMAP Domain Adaptation*

The associate editor coordinating the review of this manuscript and approving it for publication was Michele Magno.

(UMAP DA), which applies dimensionality reduction tailored to domain-specific data. By projecting the data into a lower-dimensional space and forcing UMAP to connect *only nodes from different domains*, UMAP DA further reduces domain shift, enabling more precise cross-domain learning.

To leverage these improved representations, we construct a heterogeneous bipartite graph that integrates the pseudo-labels in this reduced space. This graph-based approach allows for more accurate classification by exploiting the relationships between data points across different domains in a low-dimensional space.

Through extensive experimentation, our proposed method demonstrates significant performance improvements in terms of *F1-score* across six distinct text classification datasets. In conclusion, the key contributions of this paper can be summarized as follows:

- We propose a novel method for generating cross-domain data representations based on pseudo-labels.
- We introduce a new domain-specific dimensionality reduction approach, UMAP DA, that effectively mitigates domain shift through a lower dimension projection.
- We present a novel heterogeneous bipartite graph that leverages pseudo-labels in a low-dimensional space for final classification.

The remainder of this paper is organized as follows. Section II provides the foundation and reviews related work on domain adaptation emphasizing their relevance to our approach. Section III presents the proposed method, detailing its improvements in few-shot domain adaptation scenarios. Section IV reports the experimental results, offering a comparative analysis against state-of-the-art domain adaptation methods on six benchmark datasets. Finally, Section V summarizes the key contributions of this work and discusses potential directions for future research.

II. BACKGROUND AND RELATED WORKS

Recent studies in cross-domain learning have demonstrated substantial progress through the application of domain adaptation techniques, which seek to mitigate the shift of data distribution between the source and target domains [1], [2], [18]. Several promising approaches have been developed to address this challenge, including pseudo-labeling methods, ensemble training, and the use of Graph Neural Networks (GNNs) [19], [20], [21], [22], [23]. These techniques have shown effectiveness in narrowing the domain gap and enhancing model generalization across diverse datasets [24], [25], [26], [27].

Building on these advancements, this work introduces a novel domain adaptation method that leverages pseudo-labeling to create an explainable vector space for more effective cross-domain learning. Our approach involves generating pseudo-labels through an ensemble, which enables us to produce robust and interpretable representations. These representations serve as inputs to a new UMAP DA method, designed to create lower-dimensional representations while effectively minimizing domain shift. Furthermore, we used these low-dimensional representations to train a GNN for cross-domain text classification.

Our method demonstrates significant performance gains, particularly in terms of the *F1-score*, for text classification tasks. By integrating pseudo-labels into the GNN architecture, we enhance both the interpretability of the representations and the accuracy of the model when applied to cross-domain scenarios.

A. DOMAIN ADAPTATION

Traditional machine learning methods operate under the assumption that training and testing data originate from the same distribution. However, this assumption often fails in real-world scenarios. For instance, a model trained to classify book reviews may encounter significant difficulties when tasked with classifying texts related to electronic products. Domain adaptation, a subset of transfer learning, addresses this issue by focusing on adapting a model trained in a source domain to function effectively in a different target domain [28], [29].

The primary objective of domain adaption is to align the discrepancies between these domains to generate a domain-invariant representation [7], [30], [31]. Among the key categories of domain adaption methods, we can identify feature-based and instance-based approaches [1]. Feature-based methods aim to align data distributions by learning a domain-invariant feature space, facilitating the transfer of knowledge from the source to the target domain, while instance-based methods prioritize instances from the source domain that exhibit greater similarity to the target domain, thus enhancing the model performance in the new context [32], [33], [34].

Among the traditional approaches in the field of domain adaptation, Feature Augmentation (FA) stands out. This method establishes three distinct representations for the same data: a generic representation applicable to both the source and target domains, a target domain-specific representation, and a source domain-specific representation. By leveraging the generic representation, FA enables the generation of a domain-independent representation while capturing the intrinsic characteristics of each domain through the specific representations [35].

Another notable technique is CORAL (Correlation Alignment), which addresses domain shift by aligning the covariance matrices of the two domains. This alignment reduces discrepancies in feature distributions, facilitating better performance on the target domain [36].

Subspace Alignment (SA) method employs techniques like Principal Component Analysis (PCA) to generate a lower-dimensional subspace that represents both domains. This approach involves applying a transformation matrix to align the two subspaces derived from the source and target

domains, thus enhancing the model's ability to generalize between differing distributions [37].

One of the main domain adaptation techniques within adversarial training was introduced by Ganin et al. This method Discriminative Adversarial Neural Network (DANN) consists of a three-part model [20]. The first component is an encoder that is responsible for generating embeddings from the input data. Subsequently, specific layers designed for the downstream task are applied. Lastly, domain classification layers are added. The objective of the model is to produce embeddings that are difficult to predict accurately by domain classification layers.

This is accomplished by applying a reverse gradient to the loss of domain classification, effectively forcing the encoder to learn domain-invariant features. When the domain classifier performs poorly, it indicates that the embeddings are similar across domains, thereby reducing the impact of data shift. This approach has demonstrated significant improvements across a range of tasks, including text classification [38], [39].

Building on the DANN, several methods have been developed, including Margin Disparity Discrepancy (MDD) and Wasserstein Distance Guided Representation Learning (WDGRL). WDGRL integrates a domain discriminator while also focusing on minimizing the Wasserstein distance between the source and target domains [40]. This approach enhances the alignment of distributions across domains, leading to improved generalization performance.

MDD aims to identify a new representation that minimizes the margin disparity discrepancy between the source and target domains. Like DANN and WDGRL, MDD employ an encoder, a task network, and a discriminator [41]. By concentrating on the margin disparity, this method seeks to bridge the gap between domains more effectively, thereby facilitating better adaptation and performance in downstream tasks.

B. ENSAMBLES

Traditional ensemble training typically employs bagging, where multiple models are trained on random subsets of the source domain dataset, known as bootstrap samples [42], [43]. This method promotes the development of diverse models, each highlighting different characteristics of the data depending on the specific bootstrap sample it was trained on, resulting in a collection of expert models [44], [45].

Ensembles are commonly used to generate weak labels, which are instrumental in training models on weakly labeled data from the target domain [27], [46]. In this approach, a model is trained on the target domain data using pseudo labels produced by a pre-trained model. This process helps to reduce the discrepancy in the representations generated between the source and target domains, facilitating domain adaptation [47], [48], [49], [50].

The rapid development of Large Language Models (LLMs), which continually push the boundaries of state-of-the-art performance, often renders the training of such models impractical due to the significant computational costs involved. This issue becomes even more pronounced when considering ensemble techniques, such as bagging, where multiple models are combined to enhance performance. Additionally, several methods do not provide access to the embeddings generated by these models, as is the case with GPT-4 [51]. The absence of these embeddings makes it difficult to apply domain adaptation techniques that rely on diverse model representations [31]. To address these challenges, the proposed approach leverages the classifications produced by different models as inputs, creating a new vector space where each dimension corresponds to the output of a particular model. This strategy not only reduces the dependence on embeddings but also enhances the interpretability of the resulting representations.

C. PSEUDO-LABELING

Pseudo-labeling is a technique that assigns labels to unlabeled data, treating these pseudo-labels as if they were true labels during training [25], [52]. This method can enhance domain discrimination and improve category feature alignment, leading to more effective intra-domain adaptation [6], [53]. Many works, such as those by Zhao and Wang and Choi et al., incorporate adversarial training to learn domain-invariant representations. These approaches are often coupled with pseudo-labeling, which aids in classifying data from the target domain by helping the model generalize across domains [54], [55].

While the goal of domain discrimination is to create domain-agnostic representations, a common drawback is the potential loss of important features specific to the target domain [56]. To mitigate this, several approaches have adopted category feature alignment, which aims to reduce the distance between data points of the same class in the feature space. For example, Ma et al. employs pseudo-labels from the target domain and optimizes a loss function designed to ensure that data from the same class have similar representations in the embedding space [57]. Additionally, methods like [56] use supervised clustering techniques to further promote this alignment, ensuring that data within the same class are grouped closely together in the learned feature space.

Although many existing works use pseudo-labels to enhance domain adaptation, this method is, to the best of our knowledge, the first to use pseudo-labels to generate features in a low-dimensional projection for domain adaptation, rather than solely for target prediction.

D. GRAPH NEURAL NETWORKS

GNNs have emerged as the state-of-the-art approach for various tasks involving graph-structured data. By operating on data in a node-edge format, GNNs can model the complex relationships between entities, enabling the extraction of both global and local information [58], [59]. This structure

allows GNNs to effectively represent non-Euclidean problems, making them particularly suitable for domains such as social network analysis, molecular chemistry, and text classification [60], [61], [62]. Graphs used in GNNs can be classified into two types: homogeneous and heterogeneous. Homogeneous graphs consist of a single type of node and edge, offering a simpler representation of relational data. In contrast, heterogeneous graphs incorporate multiple types of entities and relationships, which allows for a more detailed and nuanced representation of complex systems, this flexibility makes heterogeneous graphs a powerful tool for representing more intricate real-world scenarios, by allowing richer representations [63], [64], [65].

Among deep learning methods for graphs, one widely used approach is GraphSAGE, which learns node embeddings by sampling and aggregating a fixed number of neighboring nodes. This method is particularly effective for large and dynamic graphs due to its inductive capability [66], [67]. Another fundamental technique in GNNs is Graph Convolutional Networks (GCNs), which generalize Convolutional Neural Networks (CNNs) to graph-structured data by performing convolutional aggregation over neighboring nodes [68], [69], [70]. A key variant of GCNs that has advanced the state-of-the-art is Graph Attention Networks (GATs). GATs integrate the attention mechanism to dynamically weigh the importance of neighboring nodes during aggregation, allowing them to capture more complex and context-dependent relationships in the graph [71], [72], [73].

Works such as [22] and [74] propose methods that leverage GNNs for domain adaptation, showing that learning representations in a unified space significantly improves performance in these tasks. Specifically, they suggest integrating GNNs with adversarial training to align features across domains more effectively. A key aspect of this approach is the use of heterogeneous graphs, which offer a more flexible and expressive framework for representing different types of nodes and edges. This flexibility can lead to better results in cases with complex data structures. For instance, Yang et al. utilizes a heterogeneous graph to model nodes from multiple domains, followed by a GAT that merges the samples into a unified latent space, enhancing domain adaptation performance [75].

In this work, we construct a bipartite heterogeneous network to facilitate domain adaptation by leveraging pseudo-labels in a structured manner. A key innovation in our approach is the proposed method UMAP DA, a novel dimensionality reduction technique specifically designed to mitigate domain shift. Unlike traditional UMAP, which preserves local structures without explicit domain alignment, UMAP DA enforces inter-domain connections by restricting the similarity computation to pairs of points from different domains. This ensures that the learned low-dimensional representations emphasize cross-domain relationships rather than intra-domain clusters, effectively aligning the source and target domains in a shared latent space.

The bipartite graph is constructed using these UMAP DA reduced representations. By embedding these elements into a heterogeneous graph structure, our method enhances both interpretability and accuracy, enabling a more structured propagation of information across domains. The use of UMAP DA plays a crucial role in reducing domain discrepancy while maintaining class separation, ensuring that the final learned representation is both compact and domain-invariant. This refined embedding space allows a GNN to further exploit relational patterns, ultimately leading to improved cross-domain classification performance.

III. METHOD

In this paper, we introduce a novel approach for few-shot domain adaptation by leveraging pseudo-labels. Our method utilizes pseudo-labels generated through various models to create an embedding space, which is then employed by a heterogeneous network responsible for producing the final predictions. To facilitate easy visualization of the learned representations, we propose two distinct approaches. In the first approach, we directly utilize the generated embeddings for classification. In the second approach, we introduce a new method, UMAP DA, which projects the embeddings into a two-dimensional space to create a domain-agnostic representation, thereby reducing the domain-shift.

Figure 1 presents the proposed method, which is divided into three main components. The first part involves the feature extraction process, where we employ three distinct types of models: task-specific, zero-shot, and embedding models. From each of these models, we extract the predicted class probabilities and the associated predicted classes. Given that many models may produce irrelevant information for the specific task, it is essential to implement a mechanism to filter out unreliable pseudo-labels. To address this, we apply feature selection techniques, including wrapper methods, correlation analysis, and embedding methods.

Once this initial filtering is completed, UMAP DA is applied to project the data into a lower-dimensional space. Alternatively, the class probabilities and pseudo-labels can be used directly without dimensionality reduction. In the third step, we construct a heterogeneous bipartite network. In this network, one set of nodes represents the concatenated class probabilities, while the other set represents the pseudo-labels. This structure facilitates more robust classification and better domain adaptation by incorporating diverse sources of information from different types of models, and also allows for an interpretable representation.

A. FEATURE EXTRACTION

The first step of the proposed method involves feature extraction through various classification techniques. Given an initial text, we aim to utilize multiple models to classify the same text across different aspects. From these classifications, we generate a comprehensive feature representation of the text. This approach enables us to leverage the predictions of

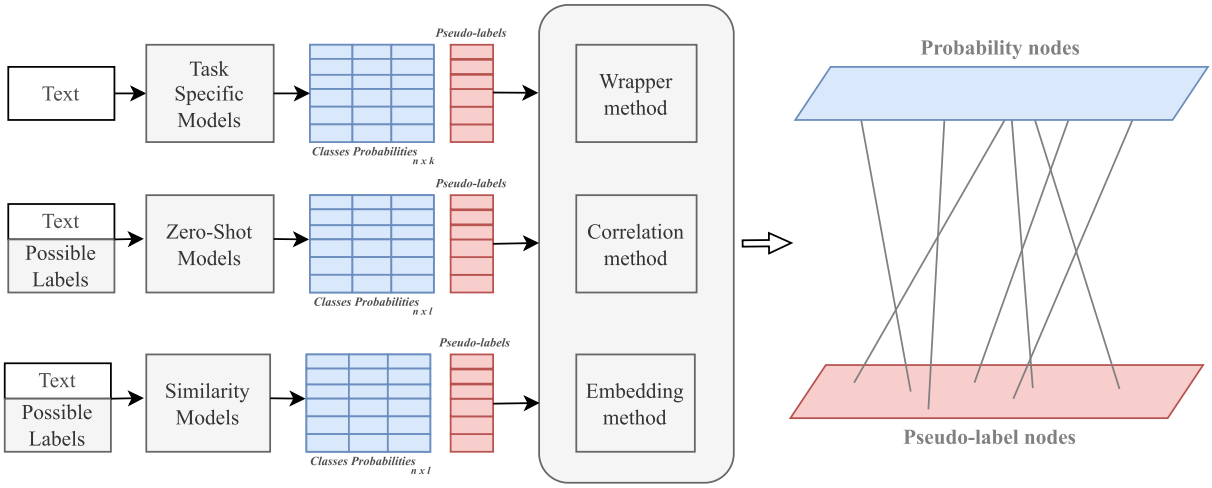

FIGURE 1. Illustration of the PDA method.

multiple models without the need to train them specifically for the target task.

Among the feature extraction techniques employed, we use *task-specific models*, which are pre-trained for specific classification tasks, such as sentiment analysis and topic classification. These models are designed to perform tasks that are closely related to the target domain task. As a result, the classifications from these models offer different perspectives on the text, which may vary significantly from the target task, necessitating a filtering process to retain only the relevant information.

Another method used for feature extraction is the *zero-shot model*. Given a set of possible classes, this model predicts the most likely class for a given text. Zero-Shot models allow us to establish pseudo-labels, which not only serve as input to the proposed method but also function as weak labels for training the final neural network. When considering zero-Shot models based on text entailment, we extract the probability distribution across the possible labels. For models such as LLMs that predict only the most likely token, we treat the prediction as a one-hot encoded representation of the predicted class.

Lastly, we employ similarity-based models for classification. These models compute the similarity between the input text and the target classes by extracting text embeddings. Instead of directly using these embeddings in the proposed method, we use them to generate classifications based on similarity. In one approach, we compute the similarity between the embedding of each class and the full text, followed by applying a softmax function over the resulting similarity scores to obtain class probabilities. In another representation, we divide the text into sentences and identify the sentence that is most similar to each class. We then apply softmax over these similarities to obtain the final probabilities. Considering the first approach, given a text t and a set of classes $C = \{c_1, c_2, \cdots, , c_n\}$, we have a model M that generates the embeddings. We define the probability that the text t belongs to the class c_i as:

$$P(c_i \mid t) = \frac{\exp(\text{sim}(\text{emb}_M(t), \text{emb}_M(c_i)))}{\sum_{j=1}^{n} \exp(\text{sim}(\text{emb}_M(t), \text{emb}_M(c_j)))} \quad (1)$$

where $\text{emb}_M(x)$ is the embedding generated by the model M for a given text x. and sim(x,y): a similarity function between the embeddings x and y.

For the second case, considering the text t we divide it into sentences $S = \{s_1, s_2, \cdots, s_K\}$, then the similarity is considered based on the sentence s_i most similar to the class c_i, and can be represented as:

$$P(c_i \mid t) = \frac{\exp\left(\max_{1 \le i \le K} \text{sim}(\text{emb}_M(s_i), \text{emb}_M(c_i))\right)}{\sum_{j=1}^{n} \exp\left(\max_{1 \le k \le K} \text{sim}(\text{emb}_M(s_k), \text{emb}_M(c_j))\right)} \quad (2)$$

We concatenate the probabilities generated by each model into a single feature vector, which serves as the input to the proposed method. This feature vector is then used to create one of the node types in the heterogeneous network, enabling the integration of diverse classification perspectives into a unified representation for downstream tasks.

B. FEATURE SELECTION

When extracting multiple models that generate pseudo-labels, which may differ significantly from the target domain, it becomes crucial to establish methods for determining the relevance of these pseudo-labels to the final classification. To address this challenge, we propose three distinct approaches for assessing the relevance of each pseudo-label: a wrapper, a correlation, and an embedding method. These methods allow us to rank the pseudo-labels based on their relevance, enabling us to select those with the highest average rankings.

Consider the dataset $\mathbf{X} = [x_1, x_2, \cdots, x_n]$ of size n, where each $x_i \in \mathbb{R}^d$ is a feature vector of dimension d, representing the i-th data point. The goal is to classify a set of target

variables y. To facilitate this classification, we introduce a binary feature selection vector $s \in 0, 1^d$, where each element s_i indicates whether the corresponding feature is selected: if $s_i = 1$, the i-th feature is included; otherwise, $s_i = 0$.

In this setting, our goal is to identify the vector s that minimizes the classification loss function $\mathcal{L}(F(\mathbf{X}), y)$, where $F(\mathbf{X})$ represents the predictive model applied to the selected features. Specifically, $F(\mathbf{X})$ serves as the classifier tasked with predicting the values of y. The challenge lies in selecting the most relevant features through s to optimize the classifier's performance while reducing the dimensionality of the input space.

In this study, as the wrapper method, we employed Recursive Feature Elimination (RFE) in conjunction with Support Vector Machines (SVM) for feature selection [76], [77], [78]. RFE systematically selects features through a process of recursive elimination. Initially, an SVM is trained using all available features. In each iteration, the algorithm ranks the features based on their importance and removes the least relevant ones. This elimination process continues until the desired number of features is reached. For our analysis, we set a limit of one feature to derive a rank for each individual feature, facilitating a comprehensive understanding of their relevance in the classification task.

Another possible approach for feature selection is to evaluate the correlation between the features and the target classes, ranking the features based on the absolute value of their correlation [79], [80], [81]. To achieve this, we utilize Pearson correlation, computed between the pseudo-labels and the true labels. The Pearson correlation coefficient quantifies the linear relationship between two variables and is defined as:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3}$$

where x_i is the pseudo-label associated with data i and y_i is the label associated with data i, $\bar{x}$ and $\bar{y}$ are the average of the pseudo-labels and labels, and n is the amount of data.

For embedding-based methods, we apply a logistic regression model to assign coefficients to each feature, allowing to rank these features according to the absolute values of their coefficients [82], [83]. To enhance the reliability of this ranking, we combine the results from three distinct approaches, calculating the average rank for each feature. By doing so, we ensure that only the most relevant pseudo-labels are retained, while those deemed less significant are filtered out.

Despite filtering the pseudo-labels, we maintain the original probability vectors, ensuring that the associated probabilities remain intact. This approach is grounded in our experimental findings, which indicate that GNNs are relatively robust to noise in probability nodes. However, the pseudo-label nodes are more sensitive to irrelevant connections, necessitating a filtering mechanism to preserve the integrity of the model and improve classification performance.

C. UMAP DOMAIN ADAPTATION

To enhance data visualization and mitigate domain shift, we propose a novel method called UMAP DA, which aims to reduce high-dimensional representations to a two-dimensional space. This transformation allows for a more intuitive visualization of the data used by the heterogeneous network. To achieve domain adaptation with UMAP, we impose a constraint that nodes from the same domain should not be considered when calculating similarities. This forces UMAP to establish connections only between data points from different domains, thus promoting cross-domain connections. As a result, the final two-dimensional representation demonstrates reduced variance between domains, effectively aligning them more closely in the reduced space. This reduction in inter-domain variance not only improves the visualization but also supports a better understanding of domain adaptation, as the data from distinct domains appears more homogeneous in this lower-dimensional representation.

UMAP DA can be described as a function that maps data $x_i \in \mathbb{R}^n$ to a two-dimensional space, i.e. $f : \mathbb{R}^n \rightarrow \mathbb{R}^2$ generates a new two-dimensional representation for x_i, $f(\mathbf{x}_i) = \mathbf{y}_i \in \mathbb{R}^2$. Since the similarity between points in the same domain should not be considered, let D_1 and D_2 be two distinct domains of data. The domain characteristic function can be described as:

$$\chi(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 1, & \text{if } \mathbf{x}_i \in D_1 \text{ and } \mathbf{x}_j \in D_2 \text{ or vice versa,} \\ 0, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \in D_1 \text{ or both } D_2. \end{cases} \tag{4}$$

The distance function $S(\mathbf{x}_i, \mathbf{x}_j)$ can be described as follows:

$$S(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} d(\mathbf{x}_i, \mathbf{x}_j), & \text{if } \chi(\mathbf{x}_i, \mathbf{x}_j) = 1, \\ inf, & \text{if } \chi(\mathbf{x}_i, \mathbf{x}_j) = 0. \end{cases} \tag{5}$$

Aside from this modification over the distance function, we keep the traditional UMAP approach as it is. The process begins with the construction of a fuzzy simplicial set, which represents the likelihood that two points are connected in the high-dimensional space. UMAP then creates a two-dimensional projection that aims to preserve these neighborhood relationships from the original space. This approach enables a faithful representation of the local structure of the data in the reduced space while also reducing the domain-shift.

D. GRAPH NEURAL NETWORK

Using UMAP on probability vectors and pseudo-labels, we construct a bipartite heterogeneous graph. This graph is composed of two types of nodes: one is generated through the representations generated by UMAP DA on probabilities, while the other is the result of using UMAP on pseudo-labels. In the graph, probability nodes establish connections

exclusively with nodes that correspond to pseudo-labels associated with data classification.

This graph structure allows edges to represent the classifications performed by models on the data; by doing so, the probability nodes with the same classification are indirectly linked via the pseudo-label nodes. In this way, the graph not only organizes information efficiently, but also facilitates the visualization and analysis of relationships between the different classifications performed by the models. Figure 1 presents in its third quadrant the structure of the heterogeneous graph, with the blue layer showing the nodes generated through probabilities and the red layer representing the nodes generated through pseudo-labels.

Regarding the final classification using heterogeneous graphs, we opted to utilize smaller models due to our adoption of a two-dimensional representation of the data. This strategic choice allowed us to create a lighter architecture, thereby avoiding excessive complexities associated with more intricate structures. For this task, we explored three distinct models. The first model is a simple dense neural network, which, unlike the other approaches, does not leverage the relationships established by the pseudo-labels and relies solely on the probabilities for its predictions. In addition to this dense network, we implemented two GNNs, which utilize the graph structure to capture interactions between the data more adeptly.

Figure 2 illustrates the generic architecture of these models, which consists of two convolutional layers, depicted in blue, followed by dense layers represented in green. At the end of this architecture, a classification layer, shown in gray, is responsible for producing the final output of the classification task. Within the convolutional layers, we employed GraphSAGE and GATv2 to enhance the model performance.

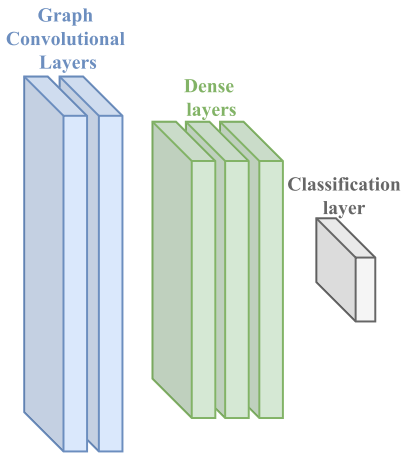

FIGURE 2. Illustration of the generic GNN method used for the final classification.

GraphSAGE is an architecture designed for efficient learning on large graphs, utilizing sampling and aggregation techniques to enhance both efficiency and scalability. The neighborhood sampling method effectively captures the local graph features, eliminating the need to process the entire graph. This aggregation process collects information from neighboring nodes in a manner that preserves the integrity of the final result, ensuring that feature propagation maintains the symmetry of local relationships [66].

While GraphSAGE employs various aggregation strategies, its primary objective is to generate robust representations of nodes by incorporating information from their immediate neighborhoods. This approach enables the model to learn valuable features without depending on the complete global graph structure, making it particularly effective for high-dimensional graphs and for representation learning in dynamic or distributed graph scenarios.

Formally given a graph $G = (V, E)$ where V represents the set of vertices and E the set of edges, and ${h_v}^{(k)}$ represents node v in layer k, the update of ${h_v}^{(k+1)}$ performed by GraphSAGE is described by an aggregation of neighbors of node v, as described in the equation:

$$\mathbf{h}_v^{(k+1)} = \sigma\left(\mathbf{W}^{(k)} \cdot AGGREGATE\left(\{\mathbf{h}_v^{(k)}\} \cup \{\mathbf{h}_u^{(k)} : u \in \mathcal{N}(v)\}\right)\right) \quad (6)$$

where $\mathcal{N}(v)$ is the set of neighbors of v in the graph, $AGGREGATE$ is the network aggregation function, W^k is the learned weight matrix, and σ is the network activation function.

GATv2 enhances the process of node aggregation in graph structures by replacing the uniform assignment of weights with a attention mechanism. This mechanism evaluates the relevance of each neighboring node during aggregation, allowing for the assignment of distinct weights based on their importance. The attention coefficients are learned during the training process, enabling the model to prioritize nodes that are more relevant to the task at hand [73]. This capability is particularly advantageous for modeling heterogeneous or large-scale graphs, as it facilitates targeted aggregation that takes into account the characteristics of the observed neighborhood. Moreover, GATv2 introduces a dynamic attention function that enhances the flexibility of modeling interactions between nodes. A key feature of this attention computation is its invariance to node order, which ensures that the aggregation of incoming messages remains unaffected by the sequence of connections among neighbors. This property significantly improves the robustness of the model by preventing bias from being introduced due to arbitrary ordering of connections within the graph.

Considering the graph $G = (V, E)$ where V is the set of vertices, E is the set of edges, and h_i is the representation of the $i - th$ node, GATv2 calculates the next representation h_i' via the following:

$$e_{ij} = \text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{W}\mathbf{h}_i \,||\, \mathbf{W}\mathbf{h}_j\right]\right) \quad (7)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})} \quad (8)$$

$$\mathbf{h}_i' = \Big\|_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j\right) \tag{9}$$

W represents the learnable weight, || denotes concatenation, a is the attention weights, e_{ij} is the not normalized attention coefficient between the nodes i and j, *LeakyRelu* the activation function used. From the e_{ij} coefficient we need to apply the normalization, which is calculated via the *softmax* function over all the neighbors of the node i. This is represented on the second equation, where α_{ij} represents the normalized attention coefficient between i and j, and $\mathcal{N}(i)$ is the set of neighbors of i. The last equation is the sum of the attention weights over the forward pass of each neighbor followed by an activation function σ, $\|_{k=1}^{K}$ represents the K multi attention heads.

Algorithm 1 Feature Extraction

for each input sample $x \in X$ **do**
  for each model $M \in \mathcal{M}$ **do**
    if M is similarity-based **then**
      $P_M(c_i \mid x) = \text{softmax}(\text{sim}(\text{emb}_M(x), \text{emb}_M(c_i)))$
      ▷ For similarity-based models
    else
      $P_M(c_i \mid x) = \text{model\_out}(x)$
      ▷ For task-specific and zero-shot models
    end if
  end for
  Compute aggregated probability:
  $P_i(c_i \mid x) = \|_{M=1}^{\mathcal{M}} P_M(c_i \mid x)$
  Compute pseudo-label:
  $PL_i(c_i \mid x) = \|_{M=1}^{\mathcal{M}} \arg\max(P_M(c_i \mid x))$
end for

E. ALGORITHMS

Algorithm 1 details our feature extraction process, where for each input sample x we iterate over a set of models $\mathcal{M}$ to compute class probabilities and pseudo-labels, for similarity-based models, the probabilities $P_M(c_i \mid x)$ are derived by applying a softmax function to the similarity scores between the sample and class embeddings, while for task-specific and zero-shot models, the probabilities are obtained directly from the model outputs. These individual probabilities are then aggregated using a predefined operator $\|_{M=1}^{\mathcal{M}}$ to yield the final predicted distribution $P_i(c_i \mid x)$ and pseudo-labels $PL_i(c_i \mid x)$, the latter being computed by aggregating the arg max results from each model. Ultimately, the concatenation of these aggregated probabilities forms the final representation used in our method.

We employ three feature selection techniques to identify the most relevant features for classification. Algorithm 2 utilizes a SVM with an RBF kernel to iteratively eliminate the least important features based on ranking scores derived from the optimization problem and the gradient of the kernel function. Algorithm 3 computes the Pearson correlation coefficient between each feature and the labels to rank features according to their statistical association with the target variable. Finally, Algorithm 4 adopts an embedded approach by training a logistic regression model with regularization,

Algorithm 2 Feature Selection: RFE With SVM

Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with d features
Initialize feature set $\mathcal{F} \leftarrow \{1, \ldots, d\}$
while $|\mathcal{F}| > 1$ **do**
  Train SVM with RBF kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$$

  Solve the optimization problem:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

  Compute feature ranking scores:

$$\frac{\partial K(\mathbf{x}_i, \mathbf{x}_k)}{\partial x_j} = 2\gamma(\mathbf{x}_{i,j} - \mathbf{x}_{k,j}) K(\mathbf{x}_i, \mathbf{x}_k)$$

$$R_j = \sum_{i=1}^{n} \sum_{k=1}^{n} \alpha_i \alpha_k y_i y_k \frac{\partial K(\mathbf{x}_i, \mathbf{x}_k)}{\partial x_j}, \quad \forall j \in \mathcal{F}$$

  Remove the least important feature:

$$\mathcal{F} \leftarrow \mathcal{F} \setminus \{j^*\}, \quad j^* = \arg\min_{j\in\mathcal{F}} R_j$$

end while
Rank features in descending order of correlation.
Select the top k features.

Algorithm 3 Feature Selection: Pearson Correlation

Compute Pearson correlation coefficient:

$$r_j = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Rank features in descending order of correlation.
Select the top k features.

Algorithm 4 Feature Selection: Logistic Regression Coefficients

Train a logistic regression model with L1 regularization by solving:

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + \exp(-y_i \mathbf{w}^\top \mathbf{x}_i)\right) + \lambda \|\mathbf{w}\|_1$$

Extract the learned coefficient vector:

$$\mathbf{w} = [w_1, w_2, \ldots, w_d]^\top$$

Rank features in descending order of importance I_j.
Select the top k features.

thereby extracting and ranking features according to the magnitude of their learned coefficients.

Algorithm 5 UMAP DA: Domain Adaptation via Modified UMAP

Define the domain characteristic function:

$$\chi(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 1, & \text{if } \mathbf{x}_i \in D_1 \text{ and } \mathbf{x}_j \in D_2 \text{ or vice versa,} \\ 0, & \text{if both } \mathbf{x}_i, \mathbf{x}_j \in D_1 \text{ or both in } D_2. \end{cases}$$

Compute pairwise distances for all pairs (i, j):

$$d(\mathbf{x}_i, \mathbf{x}_j) \quad \text{for } i, j = 1, \ldots, N$$

for all $i = 1, \ldots, N$ **do**
  for all $j = 1, \ldots, N$ **do**
    if $\chi(\mathbf{x}_i, \mathbf{x}_j) = 1$ **then**
      $S(\mathbf{x}_i, \mathbf{x}_j) \leftarrow d(\mathbf{x}_i, \mathbf{x}_j)$
    else
      $S(\mathbf{x}_i, \mathbf{x}_j) \leftarrow \infty$
    end if
  end for
end for
Construct a fuzzy simplicial set $\mathcal{F}$ using the modified distance matrix S.
(This step is identical to traditional UMAP, but now the local connectivity is based on cross-domain similarities only.)
Optimize the low-dimensional mapping $f : \mathbb{R}^n \to \mathbb{R}^2$ by minimizing the cross-entropy loss between the high-dimensional fuzzy simplicial set $\mathcal{F}$ and its two-dimensional counterpart.

$$\min_f \ \mathcal{L}(\mathcal{F}, f(\mathcal{F}))$$

Return the embeddings $\{\mathbf{y}_i = f(\mathbf{x}_i)\}_{i=1}^N$

Algorithm 5 presents the UMAP DA method for domain adaptation. The main idea is to modify the calculation of distances so that, when constructing the fuzzy simplicial set, the connections between data from the same domain are disregarded, defining their distances as infinite. In this way, only the similarities between samples from different domains are used, bringing these representations in latent space closer together. This alignment helps to align the distributions of the two domains, while preserving the separation between the classes. The implications and details of this approach are discussed further in subsection IV-C3.

Based on the representation established by UMAP DA, we construct a bipartite heterogeneous graph composed of two types of nodes, feature nodes and class nodes, with edges connecting only nodes of different types. We then train a GNN consisting of GATv2 convolutional layers. The forward pass of this process is detailed in Algorithm 6. For every node i and each of its neighboring nodes j, the algorithm first computes linear embeddings. These embeddings are concatenated and passed through a *LeakyReLU* activation to obtain an unnormalized attention score e_{ij}. Next, these scores are normalized using a *softmax* function, resulting in attention coefficients α_{ij} that quantify the relative importance of the characteristics of each neighbor. The weighted sum of the neighbor features, using the computed attention coefficients as weights, is then passed through an activation function to

Algorithm 6 GATv2 Forward Pass

for $k = 1$ to K **do**
  for each node $i \in V$ **do**
    for each neighbor $j \in \mathcal{N}(i)$ **do**
      Compute linear embeddings:

$$\mathbf{h}_i^k = \mathbf{W}^k \mathbf{h}_i, \quad \mathbf{h}_j^k = \mathbf{W}^k \mathbf{h}_j$$

      Compute unnormalized attention coefficient:

$$e_{ij}^k = \text{LeakyReLU}\left(\mathbf{a}^T \left[\mathbf{h}_i^k \parallel \mathbf{h}_j^k\right]\right)$$

    end for
    Normalize attention coefficients:

$$\alpha_{ij}^k = \frac{\exp(e_{ij}^k)}{\sum_{l \in \mathcal{N}(i)} \exp(e_{il}^k)}$$

    Compute head-specific updated representation:

$$\mathbf{z}_i^k = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^k \mathbf{h}_j^k\right)$$

  end for
end for
Concatenate the outputs of all heads:

$$\mathbf{h}_i' = \|_{k=1}^K \mathbf{z}_i^k \quad \forall i \in V$$

Return $\{\mathbf{h}_i'\}_{i \in V}$

yield a head-specific updated representation for each node. Finally, the outputs from all attention heads are concatenated to form the final node representation, effectively aggregating diverse information from each node's local neighborhood. These representations are then used for the downstream classification task.

Algorithm 7 outlines the proposed PDA method, detailing its key stages and the algorithms employed at each step. The process begins with feature extraction, where Algorithm 1 is used to generate both feature nodes and pseudo-label nodes. Subsequently, feature selection is performed using a combination of embedded methods (4), Pearson correlation analysis (3), and recursive feature elimination with SVM (2) to retain the most informative features. To address domain shift, the UMAP DA Algorithm 5 projects the selected features into a low-dimensional space while reinforcing cross-domain relationships. The adapted features and pseudo-labels are then structured into a heterogeneous bipartite graph, where edges encode pseudo-label dependencies to capture the underlying domain structure. Finally, a Graph Neural Network (GNN) leveraging the GATv2 architecture (6) is trained on this graph, learning rich node representations that facilitate accurate classification.

IV. EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of the proposed method across six distinct datasets using a ten-fold cross-validation approach. This evaluation compares our method

Algorithm 7 PDA: Pseudo-Label Domain Adaptation

1: **Feature Extraction**
2: Use Algorithm 1 to create the feature nodes and the pseudo-label nodes.
3: **Feature Selection**
4: Use Algorithms 2, 3, and 4 to select the most relevant features for the given task.
5: **Dimensionality Reduction and Domain Adaptation with UMAP DA**
6: Use algorithm 5 apply UMAP DA to project feature representations into a low-dimensional space enforce cross-domain connections to mitigate domain shift.
7: **Construct Heterogeneous Bipartite Graph**
8: Initialize graph with two node types: feature nodes and pseudo-label nodes connect nodes based on pseudo-label relationships.
9: **Classification Using Graph Neural Networks**
10: Train a GNN using the GATv2 forward pass, as outlined in Algorithm 6, to learn meaningful representations from the constructed heterogeneous graph. These learned embeddings will then be utilized for classification.

against several established domain adaptation techniques from the literature. Specifically, we include the following methods in our comparison: TCA [84], fMMD [85], DANN [20], DeepCORAL [86], MCD [87], MDD [88], WDGRL [40], CCSA [89], PRED [35], SA [37], FA [35], and CORAL [36].

To provide a baseline for comparison, we include a naive approach, where domain adaptation is not applied. In this method, we perform only a fine-tuning of a pre-trained language model, followed by fine-tuning on the source domain data.

By including both established domain adaptation techniques and a baseline approach without domain adaptation, our comparisons allow for a thorough evaluation of the effectiveness of the proposed method. This enables us to assess how well our approach performs relative to various methods used for domain adaptation.

A. DATASETS

To analyze the proposed method, we used a compilation of datasets provided by [90], which aim to classify texts into different classes related to mental health analysis. This set of datasets is composed of six datasets *3k Conversations Dataset for Chatbot*, *Depression Reddit Dataset*, *Human Stress Prediction Dataset*, *Reddit Mental Health Data*, *Students Anxiety and Depression Dataset*, *Suicidal Tweet Detection Dataset*.

In each fold of our cross-validation process, we select one dataset as the target domain, while the remaining datasets serve as source domains. However, it is important to note that some of these datasets contain labels that are not present in the others. This discrepancy necessitates the inclusion of labeled data from the target domain, as it allows the models to be exposed to the new labels and creates a few-shot domain adaptation scenario.

To address this challenge, we incorporate 1% of the labeled data from the target domain into the training data. This inclusion ensures that the models have the opportunity to learn about these additional labels. For each fold, we perform a new selection of this 1% partition, and this process is repeated across the 10 folds for each dataset.

The *3k Conversations Dataset for Chatbot* is a collection of transcriptions derived from various interactions among people. These interactions encompass a range of contexts, including formal discussions, interviews, customer service exchanges, and social media conversations. The primary task associated with this dataset is to classify sentences into several distinct categories: *Normal*, *Depression*, *Anxiety*, *Bipolar*, *Personality Disorder*, *Stress*, and *Suicidal* [91].

Another dataset utilized in this study is the *Depression Reddit Dataset*, which comprises posts extracted from the social network *Reddit*. These posts are categorized into two groups: *Normal* and *Depression* [92]. Similarly, the *Human Stress Prediction Dataset* was also derived from texts collected on this social platform, focusing on the classification of texts into *Normal* or *Stress* [93].

Furthermore, the *Students Anxiety and Depression Dataset* is sourced from the social network *Facebook* and comprises posts and comments from college students. In this dataset, the texts are classified into two categories: *Normal* and *Anxiety*. Notably, the classes *Depression* and *Anxiety* are combined into a single category, which adds a layer of complexity to the domain adaptation process [94].

The *Reddit Mental Health Dataset* consists of data extracted from Reddit, featuring posts classified into several categories: *Stress*, *Depression*, *Bipolar*, *Personality Disorder*, and *Anxiety* [95]. This dataset provides a diverse array of labels related to mental health, thereby enhancing the capability to analyze emotional states across various contexts.

Finally, the *Suicidal Tweet Detection Dataset* is derived from the social network 𝕏 and consists of *tweets* classified into two categories: *Suicidal* and *Normal* [96]. This dataset is particularly significant for studying risk behaviors and detecting suicidal intentions within social networks.

B. BASE COMPARISONS

To evaluate the proposed method, we conducted a comparative analysis using several feature-based domain adaptation techniques from the existing literature. This approach allowed for a more comprehensive and meaningful assessment of the proposed method. Specifically, we compared our method against the following approaches:

- Naive: This approach involves fine-tuning a language model on the source domain data without applying any domain adaptation technique. Once fine-tuned, the model is used for classification on the target domain data. This serves as a baseline method and can be

effective when the source domain is vastly different from the target domain, presenting high discrepancies in concepts between the two.

- PRED: *Predictive Domain Adaptation* is a method that begins by training an initial model solely on data from the source domain. The predictions generated by this model for the target domain are then used to guide the training of a second model, which is fine-tuned on the target domain data. By leveraging the representations learned by the first model, the second model aims to reduce the distributional discrepancy between the source and target domains. This process creates an enriched feature space that helps align the two domains more effectively. Such an approach is especially advantageous in situations where labeled data from the target domain is limited, as it enables the transfer of knowledge from the source domain to compensate for the scarcity of labeled examples in the target domain [35].
- FA: Feature Augmentation introduces three distinct representations for the same data: a generic representation shared by both domains, a domain-specific representation for the target domain, and another for the source domain. This structure enables the generation of a domain-independent representation from the generic features while simultaneously capturing the unique characteristics of each domain through the specific representations. As a result, this approach facilitates a more robust adaptation by preserving domain-invariant information while accounting for the intrinsic differences between the source and target domains [35].
- CORAL: *Corelation Aligment* aims to minimize the domain shift by aligning the second-order statistics of the distributions from both domains. Specifically, it focuses on the covariance matrices of these distributions. To achieve this alignment, linear transformation methods are applied to the features of the source domain, with the objective of minimizing the *frobenius norm* between the covariance matrices of the source and target domains [36].
- SA: *Subspace Alignment* focuses on aligning the subspaces generated by the feature representations of both the source and target domain data. This is done by first generating subspaces through the principal components, eigenvectors of the distributions using *PCA*. Afterward, the method learns a mapping function to align these subspaces, thereby reducing the domain shift between the two domains [37].
- TCA: *Transfer Component Analysis* aims to identify *transfer components*, which are the features that exhibit shared characteristics across different domains. To achieve this, data from both the source and target domains are projected into the space defined by these transfer components. By performing this projection into a common space [84].
- fMMD: *feature Selection with MMD* aims to identify the most crucial features for domain adaptation by determining which features remain invariant across domains. To achieve this, the method selects the subset of features that minimizes the Maximum Mean Discrepancy (MMD) between the source and target domains. By focusing on these invariant features, the approach enhances the model performance in transferring knowledge across different domains [85], [97].
- DeepCORAL: *DeepCORAL: Deep CORrelation ALignment* is an method that builds upon the original CORAL technique by enabling the learning of non-linear transformations [36]. This approach aligns the correlations of the activation layers within deep neural networks, facilitating the alignment of second-order statistics between the activations of these layers across the source and target domains [86].
- DANN: *Domain-Adversarial Neural Networks* employs adversarial training to align the distributions of the source and target domains. This method consists of three primary components: the *Feature Extractor*, which extracts features from the data; the *Label Predictor*, which predicts the label associated with the data; and the *Domain Discriminator*, which aims to distinguish between the domains. The model goal is to generate domain-invariant embeddings via the *Feature Extractor*, ensuring that the *Domain Discriminator* is unable to effectively differentiate between the domains while maintaining accurate classification performance from the *Label Predictor* [20].
- WDGRL: *Wasserstein Distance Guided Representation Learning*, similar to DANN, also employs a domain discriminator but introduces the novel approach of minimizing the *Wasserstein distance*. This technique aims to align the distributions of the target and source domains more closely, thereby enhancing the model ability to generalize across different domains. By effectively reducing the discrepancy between these distributions, the method improves the performance of the model in various domain-specific tasks.
- MCD: *Maximum Classifier Discrepancy* aims to discover a new representation that minimizes the discrepancy between the source and target domains. To achieve this, the method employs adversarial training involving three neural networks: an encoder and two classifiers, each initialized differently, trained on the source domain for the same task. The primary objective is to maximize the discrepancy between the classifications generated by the two classifiers while ensuring accurate classification performance for both. This approach enables the identification of target samples that lie far from the support of the source domain. Subsequently, a feature generator is trained to minimize this discrepancy, with the goal of producing target features that are closer to the support of the source domain [41].
- MDD: *Margin Disparity Discrepancy* focuses on discovering a new representation that minimizes the margin disparity discrepancy between the source and target

domains. Similar to DANN, MDD employs an *encoder*, a *label predictor*, and a *discriminator*. The *encoder* transforms the input data into a feature space, while the task network utilizes these features to perform the primary task. Concurrently, the *discriminator* distinguishes between the features of the source and target domains. Through adversarial training, the method estimates the discrepancy between these domains, and by minimizing this difference, MDD significantly enhances the model's ability to generalize across various domains [88].

- CCSA: *Classification and Contrastive Semantic Alignment* employs a Siamese network to learn embeddings that minimize the distance between source and target pairs sharing the same label while maximizing the distance between pairs with different labels. This objective is achieved through a contrastive loss function, which reduces the distance between elements of the same class and increases the distance between elements of different classes. As a result, the method creates an embedding space where the domains are effectively aligned, ensuring that similar items are positioned closer together, while dissimilar items are placed further apart [89].

C. RESULTS

Table 1 summarizes the results obtained from various domain adaptation methods. The first column lists the specific methods employed, while the subsequent columns present the mean and standard deviation of the *F1-Score* achieved by each method across different datasets. The methods highlighted in bold denote the novel approaches proposed in this work, specifically utilizing UMAP DA to enhance domain adaptation in a lower-dimensional feature space. The *F1-Scores* that are in bold are the best-performing for each dataset.

For the base-comparisons, we utilize the MiniLM model to extract embeddings, as described in [98]. These embeddings serve as inputs to the domain adaptation methods, which generate new representations based on both the source and target domain data. Through these representations, we employ a dense neural network to produce the final classification.

For the Naive model, we employ BERT and fine-tune it only on the source domain [99]. All the parameters of the model are used during the training, after which we utilize the embedding generated by the token *CLS* for classification. This embedding is then processed through a classification layer to produce the final predictions on the target domain.

For the proposed method, we generate representations from the probabilities obtained by various models, employing approaches such as zero-shot learning, similarity measures, and task-specific classification, as discussed in the Section III. Based on these representations, we introduce three architectures: a neural network that mirrors the one used in the adaptation methods, along with GATv2 and GraphSAGE. These architectures are trained on a heterogeneous network constructed according to our proposed method. In the experiments presented in the table, we apply UMAP DA to the representations derived from the probabilities. The proposed methods are highlighted in bold, starting with the acronym *PDA*.

We trained the *naive* method for 10 epochs, whereas the other methods were trained for 200 epochs. This discrepancy in training duration arises from the computational demands of the *naive* method, which necessitates training the entire language model. In contrast, the alternative methods only require training a neural network on the generated representations, making them more efficient. To optimize performance, we experimented with various combinations of learning rates for each method and selected those that yielded the best results. Specifically, we employed a learning rate of $1e^{-5}$ for the *naive* method. For the adaptation methods, we used a learning rate of $1e^{-3}$, while our proposed method utilized a learning rate of $2e^{-3}$. Additionally, we set the dropout rate for the layers at 0.15 to enhance regularization and prevent overfitting. We employ *early stopping*, selecting the optimal epoch based on the validation dataset to generate predictions on the target dataset.

The Table 1 demonstrates the effectiveness of the proposed method compared to existing domain adaptation techniques. Overall, the PDA method utilizing GATv2 consistently yields superior results. However, it is important to highlight the performance on the dataset *Students Anxiety and Depression*. In this case, the proposed method performed poorly because, as noted in IV-A, this dataset combines the labels *Anxiety* and *Depression* into a single class. In contrast, the other datasets treat these labels as distinct classes. This discrepancy leads to challenges in reducing domain shift, as it complicates the conceptual relationships associated with the classes.

Figure 3 displays the box plot of the $F1$-score along the y-axis for each method represented along the x-axis, based on all the datasets. Since we are considering all datasets, the distribution of the $F1$-scores becomes quite dispersed. From this figure, it is evident that the proposed method using GATv2 achieved, on average, the best results.

Table 2 presents the results of the ablation studies discussed in III. The methods assigned with the suffix *EMB* indicate the outcomes achieved without utilizing *UMAP DA* on the representations generated by the classification models. In contrast, the methods labeled with *UMAP DA* reflect the results obtained through the usage of *UMAP DA*. Generally, we expect the ablation studies to yield better results without *UMAP DA* because these representations incorporate diverse forms of data representation, without losing any information. However, it is important to note that these representations are high-dimensional, making direct visualization challenging. That said, employing *UMAP DA* facilitates the generation of representations that are not only low-dimensional but also easier to visualize. Ultimately, it is noteworthy that when examining the ablation results concerning *UMAP DA*,

TABLE 1. Compression of the methods over different and datasets.

Method	Target Dataset					
	3K Conversations Dataset for ChatBot	Depression Reddit Dataset	Human Stress Prediction Dataset	Reddit Mental Health Dataset	Students Anxiety And Depression	Suicidal Tweet Detection Dataset
Naive	0.1618 ± 0.050	0.8825 ± 0.016	0.5485 ± 0.089	0.2235 ± 0.033	0.5980 ± 0.055	0.6323 ± 0.025
CCSA	0.1724 ± 0.053	0.7389 ± 0.025	0.5542 ± 0.039	0.2691 ± 0.034	0.6004 ± 0.053	0.5007 ± 0.020
CORAL	0.3295 ± 0.074	0.9388 ± 0.005	0.6346 ± 0.028	0.5626 ± 0.038	0.8582 ± 0.031	0.7037 ± 0.004
DANN	0.1675 ± 0.043	0.7295 ± 0.025	0.5461 ± 0.034	0.2697 ± 0.030	0.5646 ± 0.040	0.5016 ± 0.018
DeepCORAL	0.1596 ± 0.036	0.7115 ± 0.008	0.5369 ± 0.064	0.2496 ± 0.033	0.5697 ± 0.029	0.5609 ± 0.015
FA	0.3207 ± 0.084	0.9382 ± 0.006	0.6376 ± 0.033	0.5550 ± 0.054	0.8527 ± 0.030	0.7036 ± 0.004
MCD	0.1683 ± 0.044	0.7302 ± 0.025	0.5478 ± 0.036	0.2693 ± 0.026	0.5639 ± 0.041	0.5003 ± 0.019
MDD	0.1683 ± 0.044	0.7303 ± 0.025	0.5481 ± 0.033	0.2692 ± 0.030	0.5654 ± 0.040	0.4884 ± 0.046
PRED	0.3122 ± 0.075	0.9352 ± 0.005	0.6123 ± 0.022	0.5319 ± 0.049	0.8611 ± 0.043	0.6261 ± 0.009
SA	0.3097 ± 0.086	0.9267 ± 0.010	0.6549 ± 0.035	0.5275 ± 0.066	$\mathbf{0.8646 \pm 0.036}$	0.6614 ± 0.012
TCA	0.0089 ± 0.001	0.3333 ± 0.003	0.3331 ± 0.011	0.0653 ± 0.003	0.2447 ± 0.186	0.3333 ± 0.003
WDGRL	0.1691 ± 0.044	0.7303 ± 0.025	0.5462 ± 0.035	0.2710 ± 0.036	0.5668 ± 0.038	0.5026 ± 0.017
fMMD	0.0515 ± 0.041	0.4270 ± 0.066	0.3925 ± 0.062	0.2112 ± 0.024	0.3775 ± 0.124	0.4137 ± 0.083
PDA GATv2	$\mathbf{0.4173 \pm 0.080}$	0.9198 ± 0.051	$\mathbf{0.7320 \pm 0.027}$	$\mathbf{0.6141 \pm 0.139}$	0.8316 ± 0.045	$\mathbf{0.7358 \pm 0.006}$
PDA GraphSAGE	0.3843 ± 0.093	$\mathbf{0.9606 \pm 0.007}$	0.6779 ± 0.067	0.5807 ± 0.110	0.8087 ± 0.095	0.6654 ± 0.044
PDA NN	0.2948 ± 0.082	0.9577 ± 0.007	0.6551 ± 0.060	0.5676 ± 0.101	0.6786 ± 0.116	0.5361 ± 0.033

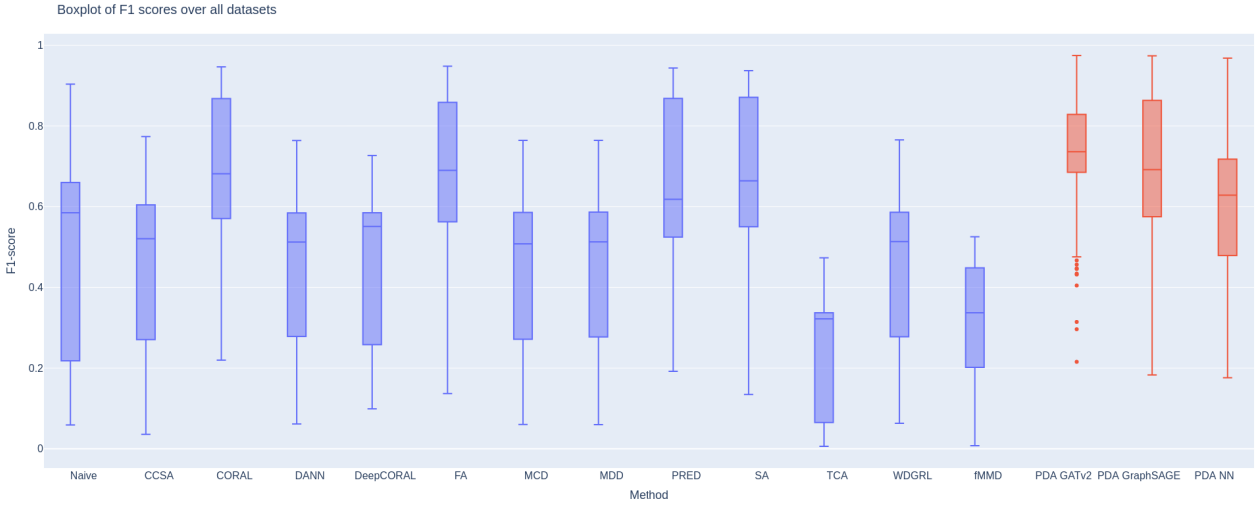

FIGURE 3. Box plot of the F1-score of the methods over all the datasets.

TABLE 2. Compression of abblation over the proposed method.

Method	Target Dataset					
	3K Conversations Dataset for ChatBot	Depression Reddit Dataset	Human Stress Prediction	Reddit Mental Health	Students Anxiety And Depression	Suicidal Tweet Detection Dataset
PDA GATv2 EMB	0.3975 ± 0.093	0.9176 ± 0.048	0.7230 ± 0.033	0.6150 ± 0.092	0.8469 ± 0.044	0.7379 ± 0.005
PDA GATv2 UMAP DA	0.4173 ± 0.080	0.9198 ± 0.051	0.7320 ± 0.027	0.6141 ± 0.139	0.8316 ± 0.045	0.7358 ± 0.006
PDA GraphSAGE EMB	0.4289 ± 0.042	0.9583 ± 0.008	$\mathbf{0.7621 \pm 0.028}$	0.6593 ± 0.055	0.8780 ± 0.036	$\mathbf{0.7383 \pm 0.004}$
PDA GraphSAGE UMAP DA	0.3843 ± 0.093	0.9606 ± 0.007	0.6779 ± 0.067	0.5807 ± 0.110	0.8087 ± 0.095	0.6654 ± 0.044
PDA NN EMB	$\mathbf{0.4736 \pm 0.003}$	$\mathbf{0.9612 \pm 0.006}$	0.7599 ± 0.031	$\mathbf{0.7093 \pm 0.010}$	$\mathbf{0.8969 \pm 0.013}$	0.7374 ± 0.006
PDA NN UMAP DA	0.2948 ± 0.082	0.9577 ± 0.007	0.6551 ± 0.060	0.5676 ± 0.101	0.6786 ± 0.116	0.5361 ± 0.033

the proposed method consistently outperforms the others base-comparisons domain adaptation methods across all datasets.

1) CD-DIAGRAM

The Figure 4 illustrates the critical difference diagram (CD-diagram) generated using the *Wilcoxon-Holm* method,

which is employed to assess pairwise significance among the various methods. The position of each classifier on the diagram reflects its mean **rank** across all datasets, with models on the leftmost side indicating superior performance. Thick lines connecting two or more methods signify that these methods do not exhibit statistically significant differences in their outcomes.

Given the multitude of methods considered, it is challenging for any single method to consistently achieve a significantly higher average rank than the others. This complexity is further compounded by the possibility that different variations of the proposed method may influence each other's rankings. As a result, additional *folds* or *datasets* may be necessary to reliably determine a genuine statistical difference in ranking among the methods. Nevertheless Figure 4 allows us to conclude that the proposed method demonstrates better average performance when considering the ranks obtained across the various datasets.

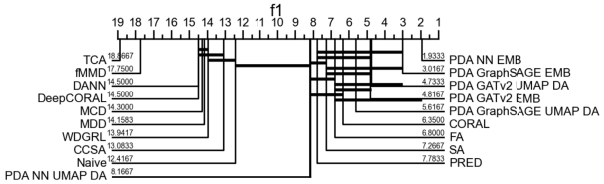

FIGURE 4. CD-diagram of the methods, considering every single fold as a different dataset.

Figure 5 illustrates the CD-diagram derived when considering only the best variation of the proposed method. This diagram reveals a statistically significant difference in rankings between the PDA and the other methods. Notably, this distinction was not apparent in Figure 4, where variations in the positions of the approaches within the *PDA* method in certain scenarios obscured the differences.

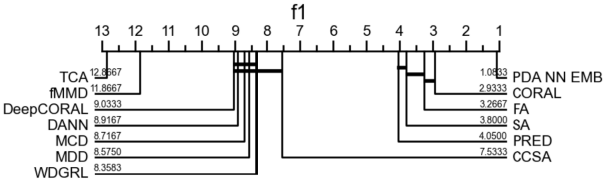

FIGURE 5. CD-diagram with only the best PDA method, considering every single fold as a different dataset.

2) EMBEDDING REPRESENTATIONS

The Figure 6 illustrates the distribution of embeddings generated by different methods applied to the *Reddit Mental Health Dataset*. For the *WDGRL*, *FA*, and *CORAL* methods, the original high-dimensional representations had to be reduced to two dimensions for visualization purposes. In order to do it, we employed UMAP for this dimensionality reduction. In contrast, the proposed method, *PDA*, does not require such a reduction, as it inherently produces a two-dimensional representation. This feature facilitates a direct comparison with the other methods. In the graphs, the colors and shapes of the points represent the different classes associated with the data, including *bipolar*, *depression*, *anxiety*, *stress*, and *personality disorders*.

Through Figure 6 clearly illustrates the relationship between the quality of the representations and the performance of each method. For instance, the *WDGRL* method exhibits lower performance, primarily due to the challenges in class separation, as evidenced by the significant overlap of points representing different classes. In contrast, the *FA* and *CORAL* methods show a more distinct separation of classes, which is directly reflected in their superior performance.

The proposed method, *PDA*, stands out for its ability to provide a clear and consistent separation between classes in the two-dimensional space, which directly contributes to its superior performance. Despite functioning in a lower-dimensional space, the representations generated by this method effectively distinguish between different classes when compared to the other methods. This characteristic of accurately differentiating among classes is a key factor in achieving better results than the alternatives.

3) UMAP DA

To demonstrate the capabilities of UMAP DA, a synthetic dataset was created consisting of 500 instances, each with ten dimensions. In this dataset, the class of each instance was defined based on the value of the first dimension, while the domain was determined from the value of the second dimension. This construction was planned so that the method could easily perform both the separation of classes and the separation of domains.

Figure 7 shows a visual comparison of the results obtained by conventional UMAP and UMAP DA. On the left, you can see the result of the traditional UMAP, which shows a clear separation of the classes in the synthetic dataset. On the right, you can see the performance of UMAP DA, which maintains a similar separation between the classes, showing that the adaptation of the algorithm does not compromise the ability to discriminate class.

On the other hand, Figure 8 highlights the main difference between the two methods when representing the distribution of domains. In the case of UMAP DA, the representations generated have a significant level of domain invariance, which is reflected in the difficulty of distinguishing the source and target domains. This characteristic is fundamental for reducing *domain-shift*. While the representations generated by the UMAP method are characterized by representations whose domain is more easily discernible.

An effective way of assessing the degree of difference between two domains is to measure the distance between their representations in the latent space. In this context, the metric based on the Euclidean distance between the representations of each domain stands out for its correlation with the visual characteristic of the representations in the latent space. To calculate this distance, the average of the distances of the N nearest neighbors that do not belong to

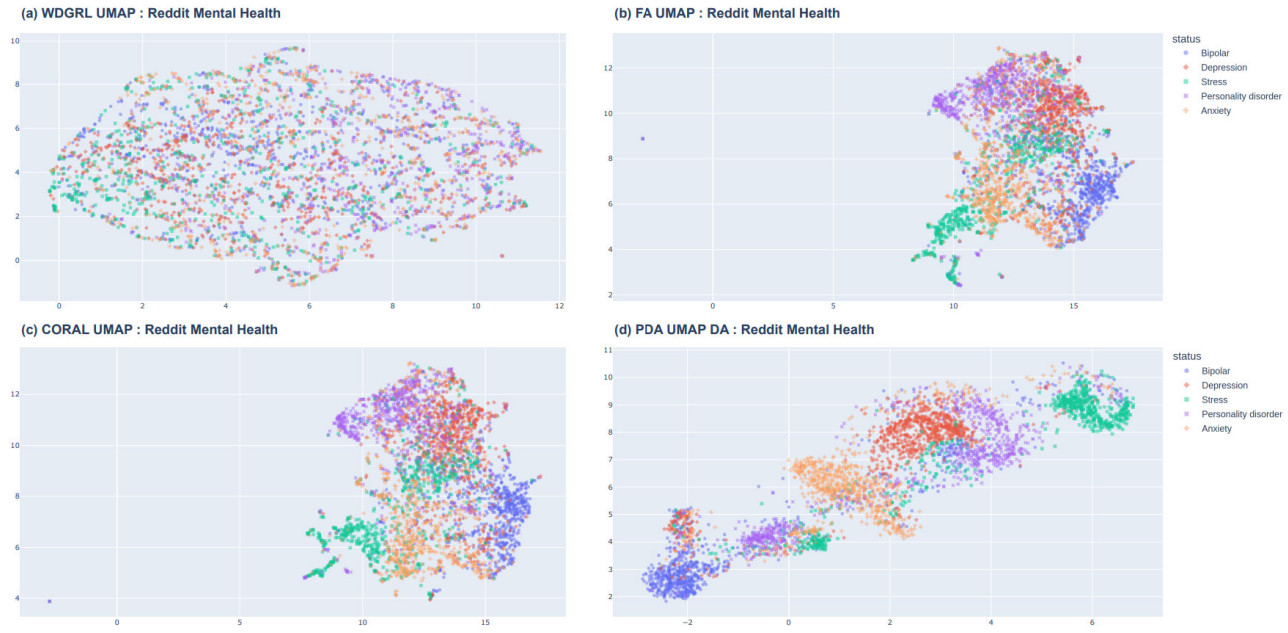

FIGURE 6. Embedding representation of the different methods.

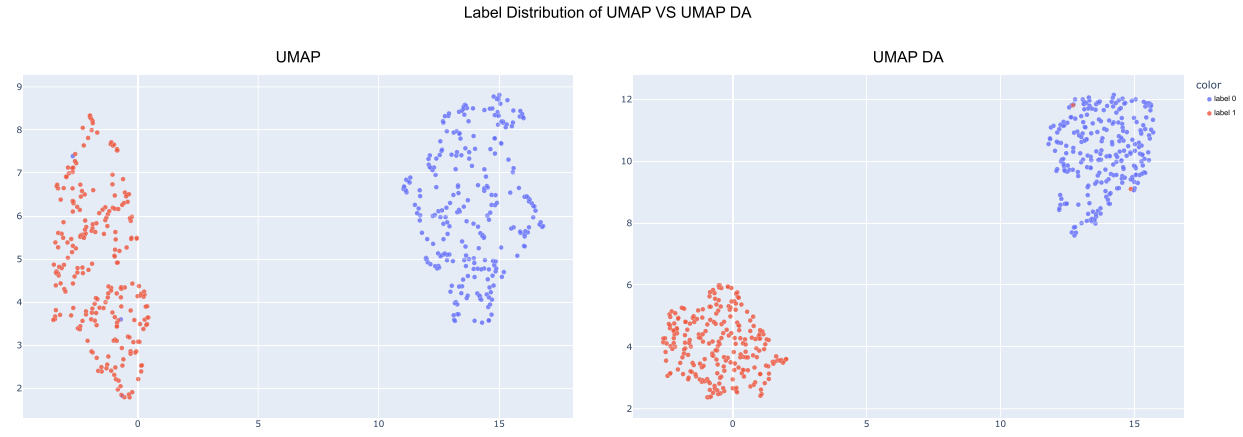

FIGURE 7. Label distribution of UMAP and UMAP DA applied to the syntactic dataset. The graph on the left illustrates the results obtained using UMAP, while the graph on the right shows the results using UMAP DA. Each point's color indicates its corresponding class. These graphs demonstrate that both methods exhibit similar class-separation capabilities.

the same domain of origin is used. This approach allows assessing proximity between the domains in the space generated by the representations, providing a quantitative indication of the alignment promoted by the methods used.

Figure 9 shows the results of this metric, visualized using *boxplots*. In this figure, the *boxplots* in red represent the results obtained with the traditional UMAP, while the *boxplots* in blue correspond to the results achieved by the UMAP DA method. The number associated with each method in the *x-axis* of the figure indicates the value of N used to calculate the neighbors' similarity.

The results show that, in all the cases analyzed, UMAP DA presents representations in which the data from different domains are closer together in the latent space compared to traditional UMAP. This proximity is essential for promoting domain adaptation, since it reduces the *domain-shift* and generates more similar representations between different domains.

The reduction in distance between domains observed with UMAP DA reflects its effectiveness in creating more aligned and invariant representations. The method manages to reduce the barriers imposed by differences in distribution between the source and target domains, while preserving

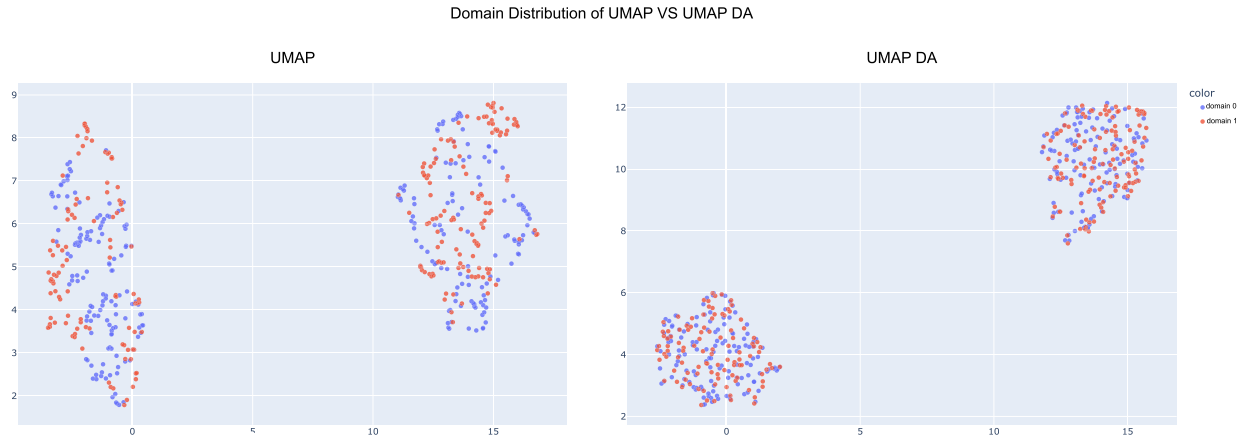

FIGURE 8. Domain distribution of UMAP and UMAP DA applied to the syntactic dataset. The graph on the left represents the results obtained using UMAP, while the graph on the right shows the results using UMAP DA. Each point's color indicates its corresponding domain. The graphs reveal that UMAP generates a representation where data from different domains are more easily distinguishable, with a clearer separation between domains. In contrast, the representations produced by UMAP DA make it more challenging to establish a distinct division between domains, providing evidence that UMAP DA brings the target and source domain distributions closer together.

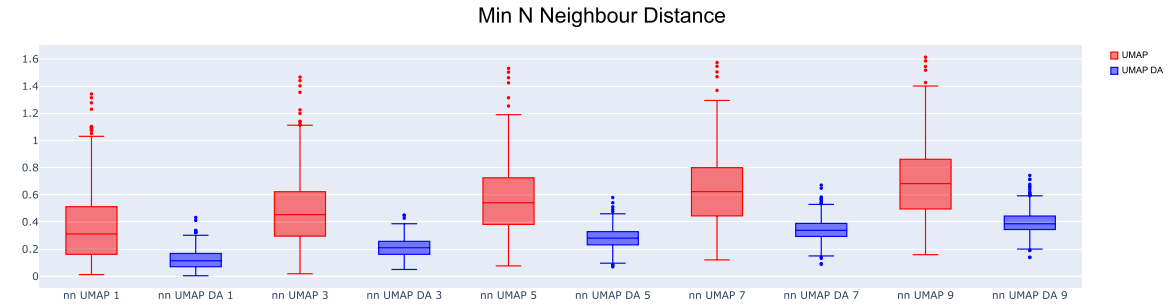

FIGURE 9. Distribution of nearest neighbors distances for UMAP and UMAP DA on the synthetic dataset. The color of each boxplot indicates the method used. The graph highlights that UMAP DA produces representations with shorter nearest neighbor distances between the source and target domains, demonstrating greater alignment between their data.

relevant information for class separation. This feature is particularly important in unsupervised or semi-supervised learning scenarios, where generalization to new domains is essential. Finally, we have made this tool available on github, which can be accessed via the following link: UMAP DA.

V. CONCLUSION

In this paper, we introduce a novel approach called *Pseudo-labeling Domain Adaptation* (PDA), which leverages pseudo-labels to address domain adaptation challenges, particularly in scenarios where labeled data is limited. To enhance this process, we also introduce an embedding space reduction technique specifically tailored for domain adaptation, called UMAP DA. PDA mitigates the issue of domain shift by generating robust data representations using multiple classification models that are subsequently projected into a two-dimensional space via UMAP DA, enhancing visualization while also reducing the domain shift. Building upon these representations, we construct a bipartite hetero graph. This graph serves as a base structure for the next phase, where a neural network is employed to perform the final classification task.

Our experimental results show that this approach leads to substantial improvements in *F1-score* across six datasets, underscoring both its effectiveness and robustness. In addition to the improved classification performance, we provide an in-depth statistical analysis comparing PDA with other domain adaptation methods reported in the literature. Furthermore, we emphasize not only the enhanced classification outcomes but also the quality of the low-dimensional data representations generated by our method, which contributes for the interpretability of the model.

Regarding future work, an interesting direction would be to integrate the *embeddings* generated by the models into the pseudo-label-based representations, potentially leading to enriched and more informative embedding

representations [100]. Moreover, it would be valuable to explore the impact of different prompts on the generation of varied representations, as outlined in recent studies [101].

ACKNOWLEDGMENT

For open access purposes, the authors have assigned the Creative Commons CC BY license to any accepted version of the article.

REFERENCES

[1] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A brief review of domain adaptation," in *Advances in Data Science and Information Engineering*. Cham, Switzerland: Springer, 2021, pp. 877–894.
[2] A. Ramponi and B. Plank, "Neural unsupervised domain adaptation in NLP—A survey," 2020, *arXiv:2006.00632*.
[3] X. Liu, C. Yoo, F. Xing, H. Oh, G. El Fakhri, J.-W. Kang, and J. Woo, "Deep unsupervised domain adaptation: A review of recent advances and perspectives," *APSIPA Trans. Signal Inf. Process.*, vol. 11, no. 1, pp. 1–51, Aug. 2022.
[4] P. Villalobos, J. Sevilla, T. Besiroglu, L. Heim, A. Ho, and M. Hobbhahn, "Machine learning model sizes and the parameter gap," 2022, *arXiv:2207.02852*.
[5] Y. Wang, Y. Pan, Z. Su, Y. Deng, Q. Zhao, L. Du, T. H. Luan, J. Kang, and D. Niyato, "Large model based agents: State-of-the-art, cooperation paradigms, security and privacy, and future trends," 2024, *arXiv:2409.14457*.
[6] Y. Li, L. Guo, and Y. Ge, "Pseudo labels for unsupervised domain adaptation: A review," *Electronics*, vol. 12, no. 15, p. 3325, Aug. 2023.
[7] V. A. K. Tomita, F. M. F. Lobato, and R. M. Marcacini, "Weak supervision for question and answering sentiment analysis," in *Proc. Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2023, pp. 1895–1900.
[8] J. Liang, R. He, Z. Sun, and T. Tan, "Exploring uncertainty in pseudo-label guided unsupervised domain adaptation," *Pattern Recognit.*, vol. 96, Dec. 2019, Art. no. 106996.
[9] Y. Li, J. Yin, and L. Chen, "Informative pseudo-labeling for graph neural networks with few labels," *Data Mining Knowl. Discovery*, vol. 37, no. 1, pp. 228–254, Jan. 2023.
[10] P. Kage, J. C. Rothenberger, P. Andreadis, and D. I. Diochnos, "A review of pseudo-labeling for computer vision," 2024, *arXiv:2408.07221*.
[11] P. Wang, X. Wang, Z. Wang, and Y. Dong, "Learning accurate pseudo-labels via feature similarity in the presence of label noise," *Appl. Sci.*, vol. 14, no. 7, p. 2759, Mar. 2024.
[12] P. Ray, S. S. Reddy, and T. Banerjee, "Various dimension reduction techniques for high dimensional data analysis: A review," *Artif. Intell. Rev.*, vol. 54, no. 5, pp. 3473–3515, Jun. 2021.
[13] W. Jia, M. Sun, J. Lian, and S. Hou, "Feature dimensionality reduction: A review," *Complex Intell. Syst.*, vol. 8, no. 3, pp. 2663–2693, Jan. 2022.
[14] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 1, pp. 56–70, May 2020.
[15] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv:1802.03426*.
[16] S. Wold, K. H. Esbensen, and P. Geladi, "Principal component analysis," *Nature Rev. Methods Primers*, vol. 2, nos. 1–3, pp. 37–52, Aug. 1987.
[17] B. Kang, D. García García, J. Lijffijt, R. Santos-Rodríguez, and T. De Bie, "Conditional t-SNE: More informative t-SNE embeddings," *Mach. Learn.*, vol. 110, no. 10, pp. 2905–2940, Oct. 2021.
[18] H. Guan and M. Liu, "Domain adaptation for medical image analysis: A survey," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 3, pp. 1173–1185, Mar. 2022.
[19] H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon, "Adversarial multiple source domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–12.
[20] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 59, pp. 1–35, Jan. 2016.
[21] X. Gu, J. Sun, and Z. Xu, "Spherical space domain adaptation with robust pseudo-label loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9098–9107.
[22] S. Saha, S. Zhao, and X. X. Zhu, "Multitarget domain adaptation for remote sensing classification using graph neural network," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.
[23] J. Yuan, F. Hou, Y. Du, Z. Shi, X. Geng, J. Fan, and Y. Rui, "Self-supervised graph neural network for multi-source domain adaptation," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 3907–3916.
[24] H. Wu and X. Shi, "Adversarial soft prompt tuning for cross-domain sentiment analysis," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2022, pp. 2438–2447.
[25] Z. Hu, Z. Yang, X. Hu, and R. Nevatia, "SimPLE: Similar pseudo label exploitation for semi-supervised classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15094–15103.
[26] C. Li, W. Chen, X. Luo, Y. He, and Y. Tan, "Adaptive pseudo labeling for source-free domain adaptation in medical image segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 1091–1095.
[27] S. Alsafari, "Ensemble-based semi-supervised learning for hate speech detection," in *Int. FLAIRS Conf. Proc.*, vol. 34, Apr. 2021, pp. 1–6.
[28] S. Zhao, B. Li, C. Reed, P. Xu, and K. Keutzer, "Multi-source domain adaptation in the deep learning era: A systematic survey," 2020, *arXiv:2002.12169*.
[29] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," 2017, *arXiv:1702.05374*.
[30] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 5, pp. 1–46, Oct. 2020.
[31] V. A. K. Tomita, A. C. M. da Silva, and R. M. Marcacini, "Cluster fusion training: Exploring cluster analysis to enhance cross-domain sentiment classification," in *Proc. Anais do Encontro Nacional de Inteligência Artif. e Computacional*, 2023, pp. 330–344.
[32] L. Du, J. Tan, H. Yang, J. Feng, X. Xue, Q. Zheng, X. Ye, and X. Zhang, "SSF-DAN: Separated semantic feature based domain adaptation network for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 982–991.
[33] A. Sicilia, X. Zhao, and S. J. Hwang, "Domain adversarial neural networks for domain generalization: When it works and how to improve," *Mach. Learn.*, vol. 112, no. 7, pp. 2685–2721, Jul. 2023.
[34] A. de Mathelin, G. Richard, F. Deheeger, M. Mougeot, and N. Vayatis, "Adversarial weighting for domain adaptation in regression," in *Proc. IEEE 33rd Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2021, pp. 49–56.
[35] H. Daumé III, "Frustratingly easy domain adaptation," 2009, *arXiv:0907.1815*.
[36] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2016, vol. 30, no. 1, pp. 1–8.
[37] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2960–2967.
[38] C. Du, H. Sun, J. Wang, Q. Qi, and J. Liao, "Adversarial and domain-aware BERT for cross-domain sentiment analysis," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4019–4028.
[39] A. Karimi, L. Rossi, and A. Prati, "Adversarial training for aspect-based sentiment analysis with BERT," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 8797–8803.
[40] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, vol. 32, no. 1, pp. 1–8.
[41] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," 2017, *arXiv:1712.02560*.
[42] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," *Eng. Appl. Artif. Intell.*, vol. 115, Jul. 2022, Art. no. 105151.
[43] S. González, S. García, J. Del Ser, L. Rokach, and F. Herrera, "A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities," *Inf. Fusion*, vol. 64, pp. 205–237, Dec. 2020.
[44] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain adaptive ensemble learning," *IEEE Trans. Image Process.*, vol. 30, pp. 8008–8018, 2021.

[45] W. Lin, Q. Lin, L. Feng, and K. C. Tan, "Ensemble of domain adaptation-based knowledge transfer for evolutionary multitasking," *IEEE Trans. Evol. Comput.*, vol. 28, no. 2, pp. 388–402, Apr. 2023.

[46] H.-C. Dong, Y.-F. Li, and Z. Zhou, "Learning from semi-supervised weak-label data," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, vol. 32, no. 1, pp. 1–8.

[47] M. Yu, J. Shen, C. Zhang, and J. Han, "Weakly-supervised hierarchical text classification," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 6826–6833.

[48] Y. Zhou, F. Zhu, P. Song, J. Han, T. Guo, and S. Hu, "An adaptive hybrid framework for cross-domain aspect-based sentiment analysis," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 16, pp. 14630–14637.

[49] Y. Ge, D. Chen, and H. Li, "Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification," 2020, *arXiv:2001.01526*.

[50] D. Hou, S. Wang, X. Tian, and H. Xing, "PCLUDA: A pseudo-label consistency learning- based unsupervised domain adaptation method for cross-domain optical remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5600314.

[51] OpenAI et al., "GPT-4 technical report," 2023, *arXiv:2303.08774*.

[52] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2013, vol. 3, no. 2, p. 896.

[53] S. Motiian, Q. Jones, S. M. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2017, pp. 1–11.

[54] X. Zhao and S. Wang, "Adversarial learning and interpolation consistency for unsupervised domain adaptation," *IEEE Access*, vol. 7, pp. 170448–170456, 2019.

[55] J. Choi, M. Jeong, T. Kim, and C. Kim, "Pseudo-labeling curriculum for unsupervised domain adaptation," 2019, *arXiv:1908.00262*.

[56] J. Li, G. Li, Y. Shi, and Y. Yu, "Cross-domain adaptive clustering for semi-supervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2505–2514.

[57] X. Ma, T. Zhang, and C. Xu, "GCAN: Graph convolutional adversarial network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8258–8268.

[58] M. Wu, M. Yan, W. Li, X. Ye, D. Fan, and Y. Xie, "Survey on characterizing and understanding GNNs from a computer architecture perspective," 2024, *arXiv:2408.01902*.

[59] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[60] X. Li, L. Sun, M. Ling, and Y. Peng, "A survey of graph neural network based recommendation in social networks," *Neurocomputing*, vol. 549, Sep. 2023, Art. no. 126441.

[61] P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, H. Metni, C. van Hoesel, H. Schopmans, T. Sommer, and P. Friederich, "Graph neural networks for materials science and chemistry," *Commun. Mater.*, vol. 3, no. 1, p. 93, Nov. 2022.

[62] M. Malekzadeh, P. Hajibabaee, M. Heidari, S. Zad, O. Uzuner, and J. H. Jones, "Review of graph neural network in text classification," in *Proc. IEEE 12th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Dec. 2021, pp. 0084–0091.

[63] R. Bing, G. Yuan, M. Zhu, F. Meng, H. Ma, and S. Qiao, "Heterogeneous graph neural networks analysis: A survey of techniques, evaluations and applications," *Artif. Intell. Rev.*, vol. 56, no. 8, pp. 8003–8042, Aug. 2023.

[64] F. Chen, Y.-C. Wang, B. Wang, and C.-C. J. Kuo, "Graph representation learning: A survey," *APSIPA Trans. Signal Inf. Process.*, vol. 9, no. 1, p. 15, Jan. 2020.

[65] A. C. M. D. Silva, D. F. Silva, and R. M. Marcacini, "Heterogeneous graph neural network for music emotion recognition," in *Proc. Proceedings*, 2022, pp. 1–8.

[66] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jan. 2017, pp. 1–11.

[67] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu, "Graph learning: A survey," *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 109–127, Apr. 2021.

[68] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: A comprehensive review," *Comput. Social Netw.*, vol. 6, no. 1, pp. 1–23, Dec. 2019.

[69] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 1, Jul. 2020, pp. 1725–1735.

[70] U. A. Bhatti, H. Tang, G. Wu, S. Marjan, and A. Hussain, "Deep learning with graph convolutional networks: An overview and latest applications in computational intelligence," *Int. J. Intell. Syst.*, vol. 2023, no. 1, Jan. 2023, Art. no. 8342104.

[71] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.

[72] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[73] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" 2021, *arXiv:2105.14491*.

[74] Q. Dai, X.-M. Wu, J. Xiao, X. Shen, and D. Wang, "Graph transfer learning via adversarial domain adaptation with graph convolution," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4908–4922, May 2023.

[75] X. Yang, C. Deng, T. Liu, and D. Tao, "Heterogeneous graph attention network for unsupervised multiple-target domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 1992–2003, Apr. 2022.

[76] H. Sanz, C. Valim, E. Vegas, J. M. Oller, and F. Reverter, "SVM-RFE: Selection and visualization of the most relevant features through non-linear kernels," *BMC Bioinf.*, vol. 19, no. 1, pp. 1–18, Dec. 2018.

[77] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1, pp. 389–422, 2002.

[78] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, Jul. 1992, pp. 144–152.

[79] H. Zhou, X. Wang, and R. Zhu, "Feature selection based on mutual information with correlation coefficient," *Int. J. Speech Technol.*, vol. 52, no. 5, pp. 5457–5474, Mar. 2022.

[80] A. Atmakuru, G. D. Fatta, G. Nicosia, and A. Badii, "Improved filter-based feature selection using correlation and clustering techniques," in *Proc. Int. Conf. Mach. Learn., Optim., Data Sci.* Cham, Switzerland: Springer, Jan. 2024, pp. 379–389.

[81] R. Saidi, W. Bouaguel, and N. Essoussi, "Hybrid feature selection method based on the genetic algorithm and Pearson correlation coefficient," in *Machine Learning Paradigms: Theory and Application*. Cham, Switzerland: Springer, 2019, pp. 3–24.

[82] M. Saarela and S. Jauhiainen, "Comparison of feature importance measures as explanations for classification models," *Social Netw. Appl. Sci.*, vol. 3, no. 2, p. 272, Feb. 2021.

[83] R. Zakharov and P. Dupont, "Ensemble logistic regression for feature selection," in *Proc. 6th IAPR Int. Conf. Pattern Recognit. Bioinf.*, Delft, The Netherlands. Cham, Switzerland: Springer, Jan. 2011, pp. 133–144.

[84] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[85] S. Uguroglu and J. Carbonell, "Feature selection for transfer learning," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, Jan. 2011, pp. 430–442.

[86] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, Oct. 2016, pp. 443–450.

[87] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, Apr. 2018, pp. 3723–3732.

[88] Y. Zhang, T. Liu, M. Long, and M. I. Jordan, "Bridging theory and algorithm for domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2019, pp. 7404–7413.

[89] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5716–5726.

[90] S. Sarkar. (2023). *Sentiment Analysis for Mental Health*. Accessed: Sep. 27, 2024. [Online]. Available: https://www.kaggle.com/datasets/suchintikasarkar/sentiment-analysis-for-mental-health/data

[91] K. Rajani. (2023). *3k Conversations Dataset for Chatbot*. Accessed: Sep. 29, 2024. [Online]. Available: https://www.kaggle.com/datasets/kreeshrajani/3k-conversations-dataset-for-chatbot

[92] I. Coder. (2023). *Depression Reddit Cleaned Dataset*. Accessed: Sep. 20, 2024. [Online]. Available: https://www.kaggle.com/datasets/infamouscoder/depression-reddit-cleaned
[93] K. Rajani. (2023). *Human Stress Prediction Dataset*. Accessed: Sep. 20, 2024. [Online]. Available: https://www.kaggle.com/datasets/kreeshrajani/human-stress-prediction
[94] S. Saha. (2022). *Students Anxiety and Depression Dataset*. [Online]. Available: https://www.kaggle.com/dsv/4493396
[95] N. Ghoshal. (2023). *Reddit Mental Health Data*. Accessed: Sep. 29, 2024. [Online]. Available: https://www.kaggle.com/datasets/neelghoshal/reddit-mental-health-data
[96] Aunanya. (2023). *Suicidal Tweet Detection Dataset*. Accessed: Sep. 20, 2024. [Online]. Available: https://www.kaggle.com/datasets/aunanya875/suicidal-tweet-detection-dataset
[97] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, Jul. 2006.
[98] W. Wang, H. Bao, S. Huang, L. Dong, and F. Wei, "MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers," 2021, *arXiv:2012.15828*.
[99] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[100] N. Pittaras, G. Giannakopoulos, G. Papadakis, and V. Karkaletsis, "Text classification with semantically enriched word embeddings," *Natural Lang. Eng.*, vol. 27, no. 4, pp. 391–425, Jul. 2021.
[101] Y. Mu, B. P. Wu, W. Thorne, A. Robinson, N. Aletras, C. Scarton, K. Bontcheva, and X. Song, "Navigating prompt complexity for zero-shot classification: A study of large language models in computational social science," 2023, *arXiv:2305.14310*.

VICTOR AKIHITO KAMADA TOMITA received the B.S. degree in computer science from the Institute of Mathematics and Computer Science, University of São Paulo, Brazil, in 2021, where he is currently pursuing the Ph.D. degree.

Since 2020, he has been conducting research in domain adaptation and sentiment analysis, which has resulted in the publication of three articles in these areas.

RICARDO MARCONDES MARCACINI received the Ph.D. degree in computer science from the Institute of Mathematics and Computer Science, University of São Paulo, Brazil, in 2014.

He is currently a Professor of computer science with the University of São Paulo. He has published papers in a number of international journals and conferences, such as *Decision Support Systems*, *Pattern Recognition Letters*, *Journal of Information and Data Management*, International Conference on World Wide Web, Web Intelligence Conference, and ACM Symposium on Document Engineering. His research interests include machine learning, data clustering, and data analytics systems.