XCapsUTL: Cross-domain Unsupervised Transfer Learning Framework using a Capsule Neural Network

Naman Khetan
Amazon
Bengaluru, India
nkhetan@amazon.com

Sanyog Dewani
Amazon
Delhi, India
dewanisd@amazon.com

Gokul Swamy
Amazon
Seattle, USA
swagokul@amazon.com

Vikalp Gajbhiye
Amazon
Delhi, India
gabhiyev@amazon.com

ABSTRACT

As e-commerce stores broaden their reach into new regions and introduce new products within established markets, the development of effective machine learning models becomes increasingly challenging due to the scarcity of labeled data. Traditional transfer learning methods typically require some labeled data from the target domain and often face computational bottlenecks. Despite the availability of a few transfer learning techniques, most are primarily developed for vision and text applications, making them unsuitable for other types of data. In many industries, however, tabular data is a predominant and crucial data type. Our work introduces XCapsUTL, a novel unsupervised transfer learning framework specifically designed for tabular data, aiming to fill this significant gap. Our approach leverages Capsule Neural Networks (CapsNet) to extract domain-agnostic knowledge. This knowledge is then refined using a constrained fine-tuning process, ensuring adaptability to the target task while preserving learned representations. XCapsUTL's unique feature encapsulation capabilities within CapsNet promote effective knowledge transfer without the need for designing effective feature-wise interaction approaches to capture higher-level semantics. Extensive experiments demonstrate the robustness and generalization capabilities of XCapsUTL across multiple domains and datasets, highlighting its practical significance and utility in addressing the unique challenges of tabular data in industry settings.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Unsupervised learning**; • **Applied computing** → *Electronic commerce*; • **Information systems** → Data management systems; Information retrieval.

KEYWORDS

Domain Adaptation, Transfer learning, Machine learning

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3680053

ACM Reference Format:
Naman Khetan, Sanyog Dewani, Gokul Swamy, and Vikalp Gajbhiye. 2024. XCapsUTL: Cross-domain Unsupervised Transfer Learning Framework using a Capsule Neural Network. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3627673.3680053

1 INTRODUCTION

As e-commerce stores expand their reach into new geographies and introduce innovative products, the need to rapidly deploy effective machine learning (ML) solutions for various tasks becomes increasingly crucial. A prime example of this is the extension of various operational prediction models to new geographies, which can significantly reduce financial losses and lower operational costs, resulting in cost savings for these e-commerce stores. One domain where this is particularly relevant is the financial sector, which involves the use of various financial instruments and lending products to enable customers to make purchases on credit and repay through equated monthly installments (EMIs). While such credit products can increase affordability and customer engagement, its misuse can result in operational losses. With the lack of sufficient data for these products, it becomes a challenge to train ML models and take necessary actions to prevent misuse of these products.

There are multiple approaches to mitigate this bottleneck, wherein for instance we could use 'similar domain' models which rely on the availability of a closely related domain with sufficient training data to serve as a proxy for the given target label. One could also leverage unsupervised methods such as ranking[2] or clustering [31] to power nearest-neighbor type approaches. However, the most potent approach is to use transfer learning wherein a model that is learnt on a related domain (e.g., another geography) is leveraged to make predictions for the target domain. In most instances, it is not possible to use the model from the source domain as-is for prediction in the target domain owing to feature distribution shifts between the source and target domains. Multiple approaches have been developed to utilize a limited amount of training samples from the target domain to help address this co-variate shift. However, in the absence of any labelled data, it becomes a much harder problem with only a few techniques that have been developed towards completely unsupervised transfer learning. Moreover, it is important to note that most existing methods are tailored primarily for

vision [24][30][10][12][33] and text[1][15][20][29][13] data. This specialization leaves a significant gap in transfer learning applications for tabular data, which is both prevalent and critical across many industries. The absence of dedicated methods for tabular data underscores the need to develop new frameworks tailored specifically to this domain. To bridge this gap, we have developed a novel unsupervised transfer learning technique, *XCapsUTL*, designed for cross-domain tabular data learning. The key contributions of our work are as follows:

- Our proposed novel framework requires no labeled target domain data during training, facilitating transfer learning across diverse and dissimilar domains.
- To the best of our knowledge, we are the first to introduce Capsule Neural Networks for tabular data within the transfer learning paradigm.
- Rather than processing each input feature individually, capsule network encapsulates all the feature values of an input into vectorial features. This allows our model to directly learn higher level representations, avoiding the need to design effective feature-wise interaction approaches to capture these semantics.
- We present experimental results using the proposed approach on both an internal e-commerce dataset and a publicly available external dataset. The internal dataset focuses on detecting bad behaviour in usage of credit products, while the external dataset pertains to predicting diabetes in patients. The results demonstrate the robustness and generalization capabilities of the XCapsUTL framework when compared with multiple state-of-the-art baselines.

The rest of the paper is organized as follows. Section 2 provides a comprehensive review of related work in the field. Section 3 details the methodology of our proposed XCapsUTL framework, outlining its design principles, mathematical formulation, and theoretical basis. Section 4 describes the datasets used, the experimental setup, the presentation of results, and an ablation analysis. Section 5 summarizes the key findings, acknowledges potential limitations, and outlines promising directions for future research.

2 RELATED WORK

Transfer learning is a technique focused on transferring knowledge acquired in one domain to another closely related domain, particularly when it's impractical to train a localized prediction model due to a scarcity of labeled training examples. One widely adopted class of methods in this field is domain adaptation [32], which aims to address the distribution shift between source and target domains to generate robust predictions for the latter. Various approaches have been proposed for domain adaptation, encompassing techniques such as importance weighting [16], which assigns higher importance to samples in the source domain that are more relevant to the target domain, and subspace alignment methods [14][6] for mapping the distribution shift between domains. Additionally, feature augmentation-based strategies [18] and minimax learning schemes [26] have been explored to minimize prediction risk in the target domain. While most transfer learning approaches assume some level of labeling for the target domain, recent research in the unsupervised domain adaptation (UDA) literature has gained momentum. Notable UDA approaches include statistical divergence alignment [14][23], adversarial learning [9][7], generative domain mapping [27], self-supervision-based methods [28][21], and clustering-based techniques [25]. Despite their effectiveness, these approaches have inherent drawbacks and limitations.

For instance, [14] exhibits effectiveness primarily on small datasets, while being computationally expensive and sensitive to outliers on larger datasets, [23] leads to negative transfer when strict alignment is being enforced whereas [9] and [7] suffer from susceptibility to mode collapse, a phenomenon where the discriminator becomes too powerful, collapsing multiple real data modes into one, thereby hindering adaptation. Additionally, [7] relies on a Gradient Reversal Layer, which is often unstable during training, leading to convergence issues or sub-optimal results. Moreover, [27][28][21] are not applicable to tabular datasets. Although [25] offers a promising approach for domain adaptation tasks, insufficient distillation can lead to inaccurate clustering and classification in the target domain, raising questions about the approach's effectiveness.

Furthermore, most of these approaches fail to account for inherent noise in the feature spaces of source and target domains, resulting in poor predictive performance in the target domain. In our work, we aim to address these issues by effectively clustering features using a potent capsule net architecture [4][22], and employing self-supervised learning on a pretext task to anchor domain adaptation on a clustered knowledge base capable of characterizing customer traits across different geographies. We compare our approach against multiple state-of-the-art baselines from the UDA literature and highlight performance improvements on our internal and external datasets.

3 XCAPSUTL - METHODOLOGY

In this section we outline our proposed approach, XCapsUTL, for unsupervised transfer learning with application towards tabular datasets. To clearly explain the modeling methodology, we take the example of the internal dataset i.e. customer bad behaviour intent detection on credit products. The XCapsUTL architecture is summarized in figure 1. XCapsUTL employs a two-phase training approach. Phase 1 focuses on knowledge transfer by identifying a domain-agnostic (pretext) classification task. Jointly training the model on this task encourages the discovery of shared data patterns, establishing a foundation for the subsequent phase. In the second phase, we leverage the pre-trained model from phase 1 and use it for bad behaviour intent detection in the source domain, while anchoring the weights of the pre-trained network through a suitably chosen regularization scheme. This helps to generalize the model for target domain bad behaviour detection basis joint representation learning. In the following sub-sections we describe each of the components of XCapsUTL in more detail.

3.1 Domain-Agnostic Task: Rank Quartile Classification

To establish a common ground between potentially dissimilar domains, we developed an unsupervised customer ranking methodology, which aims to split the customers into multiple cohorts characterized by their interactions with the e-commerce stores ecosystem. Our innovation lies in applying this ranking concept

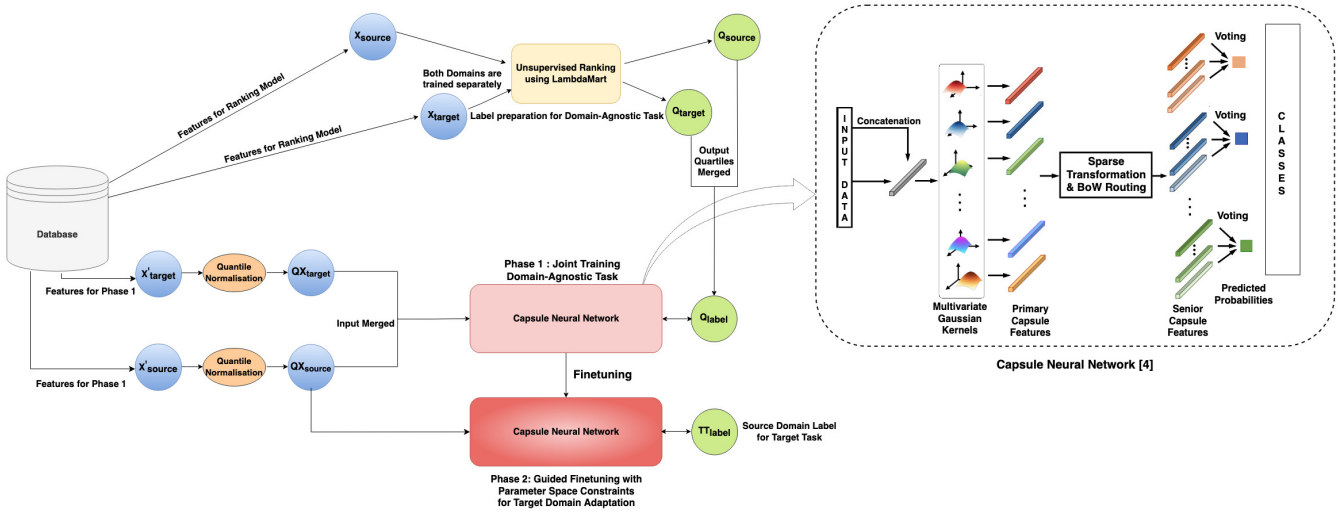

Figure 1: XCapsUTL Framework

specifically for cross-domain transfer learning, ensuring our XCapsUTL framework's adaptability and novelty across diverse and dissimilar domains. We generate customer behavior scores within each domain (source and target). Customers within each domain are then sorted by these scores and divided into quartiles which are broadly representative of a customers spend, type of purchases and other such attributes. These quartile rankings serve as labels for phase 1 training, allowing the model to learn patterns that generalize across domains. Once we have the quartile labels for each customer in both domains, we mix the data from both to prepare the final phase 1 training set.

3.1.1 ***Unsupervised Ranking Model and Quartile Generation.*** Accurately ranking millions of customers based on behavioral data presents a formidable computational challenge. The rank aggregation problem, central to this task, is inherently NP-Hard; thus, as the size of the dataset increases, the complexity of finding an optimal solution grows exponentially. Even with a relatively small set of customers, such as 50, the problem becomes computationally infeasible due to the exponential increase in the number of possible ranking combinations. This complexity arises from the necessity to compare each customer against every other, assessing a multitude of behavioral traits to establish a comprehensive ranking. Each additional customer exponentially increases the number of comparisons, rendering a direct aggregation approach unscalable for large customer bases typical in major e-commerce stores.

Given this, we seek an approximate version of the problem that can be solved efficiently. Instead of focusing on generating a fully ranked list, we consider a pairwise ranking scheme with the overall objective of minimizing pairwise discordance between a randomly selected subset of customers. Given a finite feature set, the pairwise ranking can be performed in constant time. More specifically, to rank customers from a given geography, we consider an array of real-valued features, $\mathbf{f} = [f^1, f^2, \ldots, f^k]$, which represent various aspects of customer behavior, including spending patterns (e.g., last 12-months order value), subscription status (e.g., premium membership), device preferences (e.g., Apple user), and payment methods (e.g., credit card usage). We focus on features that are commonly present in both the source and target domains, helping to generalize customer traits across multiple geographies. Given the list of features, we design a pairwise ranking scheme where the ranks of customer-a (c_a) and customer-b (c_b) are determined based on the following inequality:

$$order(\{c_a, c_b\}) = \begin{cases} (c_a, c_b), & \text{if } \sum_{i=1}^{k} \mathbb{I}(f_a^i > f_b^i) > \frac{n}{2}, \\ (c_b, c_a), & \text{otherwise} \end{cases} \tag{1}$$

where f^i denotes the i-th feature, and $\mathbb{I}$ is the indicator function which helps to establish the count of features along which customer-a outperforms customer-b. In the ordered set above, we follow a convention where first element is ranked higher than the second element. Given a large set of these pairwise preferences, $I = \{(i, j) : c_i \succ c_j\}$, our goal is to learn a scoring function $M(\mathbf{f})$ such that $M(\mathbf{f}_i) > M(\mathbf{f}_j)$ if $c_i \succ c_j$. We use a LambdaMart framework to learn this scoring function wherein we define the cross-entropy loss as:

$$\ell(i, j) = -p_{ij} \cdot \log(\hat{p}_{ij}) - (1 - p_{ij}) \cdot \log(1 - \hat{p}_{ij}), \tag{2}$$

where p_{ij} indicates the actual probability of i^{th} customer being ranked higher than j^{th} customer which is 1 in our case. And, $\hat{p}_{ij}$ indicates the predicted probability for the same. We compute the predicted probability of i^{th} customer ranked higher than j^{th} customer using the sigmoid of the difference between two scores i.e.

$$\hat{p}_{ij} = \frac{1}{1 + e^{-\lambda o_{ij}}}, \quad \text{where, } o_{ij} = M(\mathbf{f}_i) - M(\mathbf{f}_j). \tag{3}$$

Substituting above in the loss expression in Equation 2, it simplifies to

$$\ell(i, j) = -\log(\hat{p}_{ij}). \tag{4}$$

Further, by summing the pairwise loss over all input pairs in I the final loss can be expressed as,

$$L = \sum_{(i,j)\in I} -\log(\hat{p}_{ij}). \tag{5}$$

We minimize the overall loss represented by Equation 5 to learn a scoring function M for each domain. With the learnt scoring function, we score each customer in the respective geography and sort customers into quartiles to setup the pretext learning task. Before proceeding, we address a potential concern regarding the proposed methodology. The synthesized pairs only capture the ordinal relationship between two customers, while the information about the magnitude of their differences is lost. However, a closer examination of the formulation reveals that this is not a significant issue, thanks to the "law of transitivity." To illustrate this, consider a

Figure 2: A toy example to illustrate the role of transitivity in the proposed approach.

feature ranking example where feature values are represented on a real line. If we take two random pairs, such as (P1, P2) and (P2, P7), which serve as inputs to our model, it initially appears that we miss information about the distance: P1 and P2 are much closer compared to P2 and P7. However, this concern is mitigated because the input will also include pairs like (P2, P3) and (P3, P7), which help the model learn that there are many more customers between P2 and P7 than between P1 and P2. While the final scores may not reflect the exact magnitudes of differences, they will accurately represent the ordinal relationships, which is sufficient for our purpose.

3.2 Phase 1 : Capsule Network for Knowledge Discovery

In this phase, we leverage a Capsule Neural Network outlined in [4] for the transfer learning task. We chose Capsule Neural Network over a traditional neural network due to its ability to directly learn data-level semantics without feature-wise interactions, this helps the model capture data patterns quite effectively and efficiently when compared to traditional networks. We jointly train the network on both source and target domain data for the quartile classification task, Joint training enables the discovery of domain-agnostic patterns, forming a strong foundation for the subsequent phase. The input to the model consists of tabular features that are pertinent for bad behaviour intent detection and are common across source and target domains. We apply quantile normalization [17] on these raw features for improved feature representation before passing them to the network.

3.2.1 ***Capsule Network Architecture Overview.*** The architecture begins by linearly projecting input features to enhance their representation. These enhanced features are combined with the original input features to obtain basal vectorial feature, $z \in \mathbb{R}^m$. We then transform this vectorial feature into primary capsules (vector representations) using learnable multivariate Gaussian kernels. Unique communication mechanisms between capsule layers are a defining characteristic: Sparse projection generates "votes"—projections of primary capsule representations onto higher-level capsules. BoW Routing (Routing by Agreement) intelligently selects and merges these votes, determining the features of higher-level (senior) capsules.

Primary Capsules : The Architecture uses different primary capsules to represent different *profiles* of a single input. These profiles are modeled using multivariate Gaussian kernels. Each input vectorial feature z is transformed into a primary capsule feature $u_i \in \mathbb{R}^m$ for the i-th primary capsule using a multivariate Gaussian kernel k_i, as:

$$u_i = k_i(z; \mu_i, \Sigma_i) = \frac{\exp\left(-\frac{1}{2}(z-\mu_i)^T \Sigma_i^{-1}(z-\mu_i)\right)}{(2\pi)^{m/2}|\Sigma_i|}, \tag{6}$$

The parameters of the kernel μ_i and Σ_i are learned during training, allowing the model to discover patterns specific to the dataset. The model uniquely includes the original vectorial feature z as an additional capsule feature, preserving raw information alongside the learned representations.
Essentially, the activation value of a primary capsule indicates how closely the input record matches the pattern it represents. By identifying dataset-specific patterns through the Gaussian kernel parameters, model can implicitly capture feature relationships within the dataset.

Senior Capsules : Model derives senior capsules, representing target class semantics, through a multi-step process inspired by ensemble learning. First, primary capsule features are transformed into "feature votes" using a sparse weight matrix, ensuring focused information flow. Then, a specialized BoW Routing mechanism selectively combines the most relevant feature votes to synthesize the features of each senior capsule. This approach allows model to flexibly model complex target patterns while maintaining computational efficiency.

Sparse Projection : The model replaces the standard affinity projection [22] with a sparse approach. A learned weight matrix is modified using entmax [19] to select only the most relevant primary capsule features for generating feature votes.

$$\hat{u}_{j|i} = u_i W'_{j|i} \quad , \quad W'_{j|i} = \text{entmax}_\alpha(W_{j|i}) \tag{7}$$

where $\alpha = 1.5$ as default, $W'_{j|i} \in \mathbb{R}^{m\times n}$ is sparse learnable weight matrix, $\hat{u}_{j|i} \in \mathbb{R}^n$ is the feature vote for the j-th senior capsule from the i-th primary capsule. Unlike sparse attention mechanisms, applying entmax to the weights ensures consistent feature selection across different inputs.

BoW Routing : The model employ a differentiable Bag-of-Words (BoW) Routing algorithm to selectively merge feature votes $\hat{u}_{j|i}$ and synthesize senior capsule features. For a comprehensive explanation of this algorithm, please refer to [5]. This process ultimately yields the final senior capsule features v_j:

$$v_j = BoW\_Routing(\hat{u}_{j|i}) \tag{8}$$

Classification Voting : In this architecture, the vector lengths of the senior capsule features represent the existence probabilities of one semantics, and each target class is modelled by multiple senior capsules motivated by ensemble learning. Suppose there are

K classes. We divide all the senior capsules into K groups, such that the features in each group vote for one particular class.

$$\begin{aligned} l_k &= \sum_{j \in G_k} ||v_j||_2 / ||G_k|| \\ c &= \text{softmax}([l_1, \ldots, l_k, \ldots, l_K]) \\ \hat{y} &= \text{argmax}(c) \end{aligned} \tag{9}$$

where l_k denotes the mean of the senior capsule feature vector lengths for the k-th class (i.e., in the group G_k), vector c represents the predicted probabilities for all the K classes, and $\hat{y}$ denotes the predicted label. The model is trained to minimize the following margin loss function

$$L_1 = \sum_{k=1}^{K} T_k \max(0, t^+ - l_k)^2 + \lambda_1 \cdot (1 - T_k) \max(0, l_k - t^-)^2, \tag{10}$$

where $T_k = 1$ if the k-th class is the target, and otherwise $T_k = 0$. We set $t^+ = 0.9$, $t^- = 0.1$, and $\lambda_1 = 0.5$ basis [22].

3.3 Phase 2: Cross-domain Transfer Learning with Capsule Nets

In Phase 2, we leverage the domain-agnostic knowledge acquired in Phase 1 and focus on our target task of bad behaviour intent detection. We fine-tune the XCapsUTL model using the source domain's labeled bad behaviour data, enabling it to identify patterns associated with the problem at hand.

To optimize knowledge transfer, we strategically constrain the model's parameter space to prevent the model from significantly deviating from the domain-agnostic representations learned in Phase 1. This approach guides the model to learn source domain's bad behaviour patterns while maintaining a parameter space conducive to generalization for the target domain. Consequently, the refined model facilitates unsupervised bad behaviour intent detection within the target domain, showcasing XCapsUTL framework's versatility.

To optimize knowledge transfer during Phase 2, we augment the margin loss in Equation 10, with an L2 regularization term. This term constrains the model's parameter space, encouraging it to retain similarities with the domain-agnostic representations acquired in Phase 1. The complete loss function is then given as:

$$L_2 = L_1 + \lambda_2 \cdot ||\theta_2 - \theta_1||_2^2 \tag{11}$$

Where L_1 denotes the margin loss(Equation 10) for optimization on the bad behaviour intent detection task. θ_1 denotes the network parameters learned in phase 1 and θ_2 represents the current network parameters during phase 2 training. Additionally, λ_2 is a regularization parameter set to $\lambda_2 = 0.001$.

In the next section, we detail experimental results of applying XCapsUTL on internal and external datasets.

4 EXPERIMENTAL SETUP

4.1 Datasets

4.1.1 ***Internal Dataset.*** Our internal dataset consists of a source country S_1 dataset along with datasets of two different target geographies, T_1 and T_2. As the product was launched much earlier in the source country, we have access to significant labeled data for bad behaviour intent detection. However, at the time of product launch in the two new geographies, no labeled datasets were available. We utilized the S_1 dataset to build models for detecting bad behaviour intent using both the XCapsUTL framework and other transfer learning baselines. The ultimate goal was to generate predictions for bad behaviour intent in the target geographies, T_1 and T_2. For XCapsUTL model development, we prepared the data in three groups:

- **Unsupervised Ranking :** For the input features, we utilized data from the customer base with at-least one purchase in last 12 months in a geography. We selected customer-level aggregated features that are commonly present in both the source and target geographies to construct the ranking model. For this task, we chose features that are crucial for determining the ranking between two customers. Since this was an unsupervised task, no labels were required during training.
- **Phase 1 - Joint training :** In this phase, we conducted joint training with both the source and target geographies. Similar to the unsupervised ranking task, the input included data from purchase active customer base in last 12 months. However, this time the features selected were pertinent to bad behaviour intent detection. The labels used were the quartiles to which each customer belongs within their respective geography. These quartiles were generated based on the rankings from the previously mentioned unsupervised ranking task.
- **Phase 2 - Guided Fine-tuning :** In this phase, we fine-tuned the model from Phase 1 using labeled data for bad behaviour intent from the source geography S_1. We employed the same input features as those used in the Phase 1 training.

We evaluated our model's performance using three months of data for bad behaviour labels in T_1 and T_2 geographies, which became available after the model launch. For details on model productionisation approach refer section 4.2.3.

4.1.2 ***External Dataset.*** To validate the robustness of our proposed approach, we bench-marked our model using the 'CDC Diabetes Health Indicators' external dataset[3]. This dataset includes healthcare statistics, lifestyle survey information, and diabetes diagnoses for individuals. It comprises 21 variables, such as demographics, laboratory test results, and survey responses for each patient. The target variable for binary classification indicates whether a patient is diabetic/pre-diabetic or healthy. Owing to the paucity of a public-domain transfer learning dataset for the tabular domain, we leveraged the diabetes dataset to define a source and a target domain for our analysis. We split the dataset based on patients' age wherein Group 1 (50A) includes patients aged over 50, while Group 2 (50B) includes patients aged below 50. We chose age 50 for the split as it represents the midpoint of the age distribution. To ensure generalizability, we first treated Group 1 as the source domain and Group 2 as the target domain, then reversed the roles, treating Group 1 as the target domain and Group 2 as the source domain. The dataset contains 253,680 patient records, with 74,220 below and 179,460 above fifty years of age. For XCapsUTL model training and scoring, we prepared the data in three groups as follows:

- **Unsupervised Ranking :** We selected input features essential for determining the ranking between two patients. Out of the 21 available features, we used the following 10: HighBP, HighChol, BMI, Smoker, Stroke, HeartDiseaseorAttack, PhysActivity, Fruits, Veggies, and HvyAlcoholConsump. Since this task was unsupervised, no labels were required during training. We performed this task separately for both domains to obtain the quartiles for each patient.
- **Phase 1 - Joint training :** In this phase, we conducted joint training with both the source and target domains. We used all 21 available features for each patient. The labels were the quartiles to which each patient belongs within their respective domain, generated based on the rankings from the previous unsupervised ranking task.
- **Phase 2 - Guided Fine-tuning :** In this phase, we fine-tuned the model from Phase 1 using the source data. We used Diabetes_binary as the target label and employed the same input features as in the Phase 1 training.

We conducted Phase 2 twice separately: first with Group 1 (50A) as the source domain and Group 2 (50B) as the target domain, and second with Group 1 (50A) as the target domain and Group 2 (50B) as the source domain. The results are presented in Table 2.

4.2 Experiments

4.2.1 ***Results.*** In order to test the efficacy of our proposed method, we compared the performance of our model against multiple state-of-the-art baselines which includes Unsupervised Domain Adaptation via Distilled Discriminative Clustering (DisClusterDA) [25], Maximum Classifier Discrepancy for Unsupervised Domain Adaptation (MCD) [23], Maximum density divergence for domain adaptation (MMD)[14], Domain-adversarial training of neural networks (DANN) [7]. We used hyperopt [11] for hyper-parameter tuning of all the models and the models are trained on a single Nvidia Tesla T4 GPU. For XCapsUTL, we did a 80-20 split of training data into train and validation set in each phase for all the datasets. Table 1 summarizes the model performance of different techniques on bad behaviour intent detection task for the target domains of T_1 and T_2. We observe that proxy model like customer ranking [2] under-perform as compared to all transfer learning methods. Within transfer learning methods, XCapsUTL has the highest model performance for both the geographies (T_1 and T_2).Specifically, XCapsUTL achieves an AUC improvement of 3.3% for T_1 and 2.49% for T_2 over the next best method, DisClusterDA. This highlights the superior capability of our approach in capturing domain-agnostic patterns and effectively transferring knowledge across dissimilar domains.

Additionally, the results in Table 2 further validate the robustness of XCapsUTL using the external CDC Diabetes Health Indicators dataset. When evaluating the model with Group 1 (50A) as the source domain and Group 2 (50B) as the target domain, XCapsUTL achieves an AUC of 89.2, outperforming DisClusterDA by 0.53. Similarly, when the roles are reversed, our model maintains its superior performance with an AUC of 84.02, surpassing DisClusterDA by 1.57. These consistent improvements across different datasets and domains demonstrate the generalization capabilities of XCapsUTL and its potential for broader applicability in various cross-domain scenarios.

Table 1: Relative Performance Comparison of Methods AUC on internal dataset

Method	T_1	T_2
Source only	0%	0%
Customer Ranking (proxy)	39.88%	29.31%
DANN	43.77%	29.5%
MMD	40.85%	29.88%
MCD	46.49%	32.37%
DisClusterDA	50.00%	33.90%
XCapsUTL (Ours)	**53.30%**	**36.39%**

Table 2: Performance Comparison of Methods AUC on external dataset

Method	$50A \rightarrow 50B$	$50B \rightarrow 50A$
Source only	83.55	73.71
DANN	86.12	79.62
MMD	86.19	80.3
MCD	87.93	80.97
DisClusterDA	88.67	82.45
XCapsUTL (Ours)	**89.2**	**84.02**

4.2.2 ***Ablation studies.*** We conducted an ablation study to validate our hypothesis that a Capsule Network architecture is much more effective at learning domain-agnostic representations for transfer learning compared to other architectures. In our approach, we replaced the Capsule Network base model with a vanilla feed-forward network and an FT-Transformer [8], as is common in several transfer learning approaches.

As shown in Table 3, the Capsule Network consistently outperformed the alternative architectures across both internal and external datasets. Specifically, for the internal dataset, the Capsule Network achieved AUC improvements of 5.45% for T_1 and 3.64% for T_2 compared to the FT-Transformer. Similarly, for the external dataset, our approach resulted in AUC gains of 1.61 for $50A \rightarrow 50B$ and 2.78 for $50B \rightarrow 50A$ over the FT-Transformer. These results underscore the superior capability of Capsule Networks in learning complex feature interactions and enhancing transfer learning performance in diverse domains.

4.2.3 ***Model launch and Business Impact.*** XCapsUTL model was launched in production for bad behaviour intent detection use case in two geographies where labelled data was unavailable. While at the time of launch we limited the population leveraging model scores (to limit any downside risks) but post evaluation we gradually scaled up the usage of model to handle volume of several million orders per week using the specific credit instrument. The model was deployed in real time with average latency of 250 milliseconds per order. The model score generated by XCapsUTL was used to drive specific interventions (not disclosed) aimed at preventing bad behaviour entities from exploiting the system. Through these interventions, we have been able to achieve several millions of dollars of cost savings on an annualized basis. We plan to utilize this framework for bad behaviour intent prediction for all future geographical launches.

Table 3: Ablation Study: Relative Performance Comparison of Methods AUC on internal dataset and absolute AUC values on external dataset.

Methods	T_1	T_2	$50A \rightarrow 50B$	$50B \rightarrow 50A$
Feed Forward Neural Network	40.27%	29.88%	84.8	78.23
FT-transformer	47.85%	32.75%	87.59	81.24

4.2.4 ***Visualization of Primary Capsules.*** Figure 3 presents a t-SNE plot for one of the target domain input datasets from the internal dataset, where different colors represent various primary capsules of the final XCapsUTL model. The visualization clearly shows that each primary capsule captures distinct profiles of the input data. Notably, this distinction is achieved without the model being explicitly trained on the target domain, demonstrating the effectiveness of the final trained XCapsUTL model. To create this visualization, we first reduced the m-dimensional input data to two t-SNE components. Each input point was then assigned to the primary capsule with the highest activation value, solely for visualization purposes. Each primary capsule encapsulates a specific customer profile. For example, the gray-colored primary capsule (top-right) captures customers with high spending and low degree of bad behaviour, while the green-colored primary capsule (bottom-left) represents customers with high degree of bad behaviour and low spending. This visual distinction highlights the Capsule Network's ability to effectively distill high-level semantic information across domains and leverage it for cross-domain transfer learning.

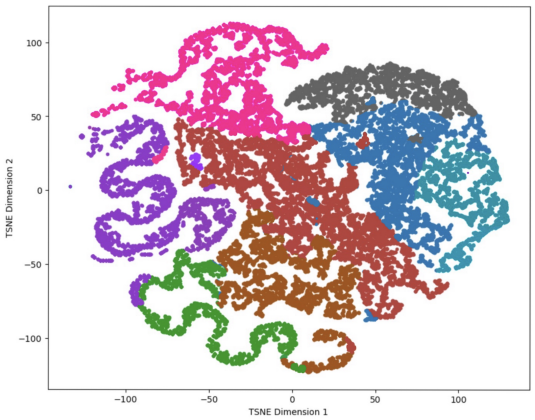

Figure 3: t-SNE plot showing distinct customer profiles captured by primary capsules in the XCapsUTL model, demonstrating effectiveness without explicit training on the target domain.

5 CONCLUSION AND FUTURE WORK

In this study, we present a novel unsupervised transfer learning approach, XCapsUTL, which effectively learns a joint representation of source and target domains through a pretext quartile ranking task. Subsequently, it employs a capsule net framework to transfer domain knowledge between the source and target domains. The synergistic combination of a ranking task on low-level features and knowledge aggregation through capsule networks is particularly effective at distilling high-level information from the source domain, aiding cross-domain transfer learning. Through rigorous testing on an internal cross-geography bad behaviour intent detection task and an external dataset for predicting diabetes in patients by leveraging patient age for domain stratification, XCapsUTL has demonstrated a significant performance edge over state-of-the-art baselines. Its real-world deployment for bad behaviour intent detection across new geographies (with new products being launched) in a large e-commerce domain has yielded substantial cost savings, fueled by enhanced cold-start risk management capabilities. XCapsUTL's is use-case agnostic and could be adapted for a wide array of problems involving tabular data based transfer learning. However, its current limitation lies in its inability to handle feature heterogeneity between source and target domains. Addressing this challenge and extending the model's capabilities to accommodate such use cases is a subject of future research.

REFERENCES

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL]

[2] Christopher Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11 (01 2010).

[3] Nilka Rios et al. Burrows. 2017. Incidence of End-Stage Renal Disease Attributed to Diabetes Among Persons with Diagnosed Diabetes - United States and Puerto Rico, 2000-2014. https://www.cdc.gov/brfss/annual_data/annual_2014.html

[4] Jintai Chen, KuanLun Liao, Yanwen Fang, Danny Chen, and Jian Wu. 2023. TabCaps: A Capsule Neural Network for Tabular Data Classification with BoW Routing. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=OgbtSLESnI

[5] Jintai Chen, Kuanlun Liao, Yanwen Fang, Danny Z Chen, and Jian Wu. 2023. TabCaps: A Capsule Neural Network for Tabular Data Classification with BoW Routing. In *ICLR*.

[6] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. 2014. Subspace Alignment For Domain Adaptation. arXiv:1409.5241 [cs.CV]

[7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-Adversarial Training of Neural Networks. arXiv:1505.07818 [stat.ML]

[8] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2023. Revisiting Deep Learning Models for Tabular Data. arXiv:2106.11959 [cs.LG]

[9] Seydi V. HassanPour Zonoozi, M. 2022. A Survey on Adversarial Domain Adaptation. *IEEE Access* (2022). https://doi.org/10.1007/s11063-022-10977-5

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90

[11] https://hyperopt.github.io/hyperopt/. [n. d.]. hyperopt.

[12] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2018. Densely Connected Convolutional Networks. arXiv:1608.06993 [cs.CV]

[13] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics,

Online, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703
[14] Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Heng Tao Shen. 2021. Maximum Density Divergence for Domain Adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 11 (Nov. 2021), 3918–3930. https://doi.org/10.1109/tpami.2020.2991050
[15] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL]
[16] Nan Lu, Tianyi Zhang, Tongtong Fang, Takeshi Teshima, and Masashi Sugiyama. 2021. Rethinking Importance Weighting for Transfer Learning. arXiv:2112.10157 [cs.LG]
[17] Quantile Normalisation. [n. d.]. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.quantile_transform.html
[18] Mauricio Orbes-Arteaga, Thomas Varsavsky, Lauge Sorensen, Mads Nielsen, Akshay Pai, Sebastien Ourselin, Marc Modat, and M Jorge Cardoso. 2022. Augmentation based unsupervised domain adaptation. arXiv:2202.11486 [eess.IV]
[19] Ben Peters, Vlad Niculae, and André F. T. Martins. 2019. Sparse Sequence-to-Sequence Models. arXiv:1905.05702 [cs.CL]
[20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html
[21] Mohamed Ragab, Emadeldeen Eldele, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, and Xiaoli Li. 2024. Self-Supervised Autoregressive Domain Adaptation for Time Series Data. *IEEE Transactions on Neural Networks and Learning Systems* 35, 1 (Jan. 2024), 1341–1351. https://doi.org/10.1109/tnnls.2022.3183252
[22] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic Routing Between Capsules. arXiv:1710.09829 [cs.CV]
[23] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation. arXiv:1712.02560 [cs.CV]
[24] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs.CV]
[25] Hui Tang, Yaowei Wang, and Kui Jia. 2023. Unsupervised Domain Adaptation via Distilled Discriminative Clustering. arXiv:2302.11984 [cs.CV]
[26] Chao Wang, Caixing Wang, Xin He, and Xingdong Feng. 2023. Minimax Optimal Transfer Learning for Kernel-based Nonparametric Regression. arXiv:2310.13966 [stat.ML]
[27] Yi Wu, Ziqiang Li, Chaoyue Wang, Heliang Zheng, Shanshan Zhao, Bin Li, and Dacheng Tao. 2023. Domain Re-Modulation for Few-Shot Generative Domain Adaptation. arXiv:2302.02550 [cs.CV]
[28] Jiaolong Xu, Liang Xiao, and Antonio M. Lopez. 2019. Self-Supervised Domain Adaptation for Computer Vision Tasks. *IEEE Access* 7 (2019), 156694–156706. https://doi.org/10.1109/access.2019.2949697
[29] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. XLNet: Generalized Autoregressive Pretraining for Language Understanding. arXiv:1906.08237 [cs.CL]
[30] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? arXiv:1411.1792 [cs.LG]
[31] Sheng Zhou, Hongjia Xu, Zhuonan Zheng, Jiawei Chen, Zhao li, Jiajun Bu, Jia Wu, Xin Wang, Wenwu Zhu, and Martin Ester. 2022. A Comprehensive Survey on Deep Clustering: Taxonomy, Challenges, and Future Directions. arXiv:2206.07579 [cs.LG]
[32] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A Comprehensive Survey on Transfer Learning. arXiv:1911.02685 [cs.LG]
[33] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. 2018. Learning Transferable Architectures for Scalable Image Recognition. arXiv:1707.07012 [cs.CV]