EConTab: Explainable Contrastive Tabular Representation Learning with Regularization

Suiyao Chen*
Industrial and Management Systems Engineering
University of South Florida
Tampa, FL, USA
sc3740@columbia.edu

Jing Wu *
Mechanical Science and Engineering
University of Illinois at Urbana-Champaign
Champaign, IL, USA
jingwu6@illinois.edu

Handong Yao
College of Engineering
University of Georgia
Athens, GA, USA
handong.yao@uga.edu

***Abstract*—Representation learning stands as one of the critical machine learning techniques across various domains. Through the acquisition of high-quality features, pre-trained embeddings significantly reduce input space redundancy, benefiting downstream pattern recognition tasks such as classification, regression, or detection. Nonetheless, in the domain of tabular data, feature engineering and selection still heavily rely on manual intervention and explanation, leading to time-consuming processes and necessitating domain expertise. In response to this challenge, we introduce EConTab, an explainable deep automatic representation learning framework with regularized contrastive learning. Agnostic to any type of modeling task, EConTab constructs an asymmetric autoencoder based on the same raw features from model inputs, producing low-dimensional representative embeddings. Specifically, regularization techniques are applied for raw feature selection and contrastive learning is leveraged to distill the most pertinent information for downstream tasks. Meanwhile, model explanation is demonstrated through feature weights and SHAP-value based model explainer. Experiments conducted on extensive real-world datasets substantiate the framework's capacity to yield substantial and robust performance improvements. Furthermore, we empirically demonstrate that pre-trained embeddings can seamlessly integrate as easily adaptable features, enhancing the performance of various traditional methods such as XGBoost and Random Forest.**

***Index Terms*—Representation Learning, Contrastive Learning, Tabular Data, Regularization, Feature Engineering, Model Explanation**

I. Introduction

Over the past decade, representation learning has achieved remarkable advancements in domains such as computer vision and natural language processing, fundamentally transforming how we extract meaningful insights from both image and text data. However, several critical industries, including healthcare [1]–[4], manufacturing [5]–[8], agriculture [9]–[11] and various engineering fields [12]–[22], still heavily rely on structured tabular data. Researchers traditionally leverage domain expertise for feature selection and uncertainty quantification [23]–[27]. It's commonly believed that traditional tree-based models, equipped with carefully crafted features, can automatically discern the importance of features and their interactions without the need of additional tuning.

Despite its effectiveness, manual feature engineering presents some challenges. Crafting high-quality features for tabular data is labor-intensive and doesn't ensure optimal performance. Feature selection typically involves iterative experimentation, with extensive consumption of resources and time. Moreover, a significant limitation lies in the explainability of models trained on such engineered features. Understanding how and why a model makes predictions is crucial for trust and adoption. To address these challenges, recent research efforts have aimed to leverage deep representation learning to streamline feature engineering in tabular data while enhancing model explainability.

However, the integration of tabular data with the remarkable success seen in deep learning across other domains encounters distinctive obstacles. Unlike text data, which inherently consists of discrete tokens, or images, where pixels exhibit spatial correlations, tabular data comprises a diverse mixture of continuous, categorical, and ordinal values. These values can demonstrate complex interdependencies and correlations, introducing layers of complexity to the modeling process. Additionally, unlike the structured nature of images or the sequential nature of text, tabular data lacks inherent positional information necessary to capture intrinsic meanings or learn explicit representations.

In this paper, we proposed EConTab, a transformer-based framework to automatically generate high-quality embeddings as features for classification improvement and establish a comprehensive model explanation process. Our framework consists of an asymmetric autoencoder (AE) architecture, which is able to extract the most critical information from raw features to provide substantial performance improvement and robustness for downstream classification tasks. Moreover, EConTab can be effectively trained in both self- and supervised modes. This adaptability ensures the model to perform well across various training scenarios, irrespective of the availability of labeled data. The contributions are summarized as follows:

- We introduced a transformer-based automatic feature engineering framework designed to be task-agnostic, featuring scalability and adaptability.
- We developed an innovative autoencoder architecture incorporating regularization and contrastive learning to enrich the feature learning process.
- We conducted a comprehensive empirical study on various public datasets that demonstrates the superiority of the

* These authors contributed equally to this work.

proposed work in performance lift and robustness.
- We demonstrated that representative embeddings extracted from raw features can serve as readily applicable features, seamlessly augmenting the performance of various conventional classification models such as logistic regression and tree-based models, etc., and achieve straightforward model explanation.

II. Related Work

A. Classical Models

Various traditional machine-learning methods have been developed for tabular data classification and regression tasks. Logistic Regression (LR) [28] and Generalized Linear Models (GLM) [29] are the prominent choices to model linear relationships. Decision Trees (DT) [30] sets the fundamental options for tree-based models. Furthermore, various ensemble methods based on DT, such as XGBoost [31], Random Forest [32], CatBoost [33], and LightGBM [34] are developed. These ensemble methods are highly favored in industry for their capacity to capture intricate non-linear relationships, improve interpretability, and effectively manage diverse feature types, encompassing null values and categorical features.

B. Deep Learning Models

A notable trend in current research has been focusing on applying deep learning techniques to tabular data [35]. This movement has led to the emergence of various neural architectures, each tailored to improve performance within the tabular data domain. These architectures can be broadly classified into several categories [5], [36], [37]. First, supervised models like ResNet [38], SNN [39], AutoInt [40], and DCN-V2 [41], have been well-known to harness the power of neural networks and improve the handling of tabular data. Second, innovative hybrid approaches like NODE [42], GrowNet [43], TabNN [44], and DeepGBM [45] could seamlessly integrate decision trees with neural networks, resulting in end-to-end training. Third, transformer-based methods such as TabNet [46], TabTransformer [47], and FT-Transformer [36] allow models to learn from attention-spanning features and data samples. Finally, representation learning methods such as VIME [48], SCARF [49], SAINT [50] and SwitchTab [51] are gaining prominence, emphasizing effective information extraction through self- and semi-supervised learning techniques. These approaches align seamlessly with the growing emphasis on representation learning in the field.

C. Self- and Semi-supervised Representation Learning

In computer vision, deep representation learning methodologies have emerged as potent tools, capitalizing on self- and semi-supervised training paradigms [52]–[56]. These methodologies exhibit a dichotomy, falling into two distinct categories of innovation. The first category of deep representation learning methods is rooted in generative models, particularly autoencoders [57]. A striking exemplar within this genre is the Masked AutoEncoder (MAE) architecture introduced by [58]. MAE features an asymmetric encoder-decoder architecture purposefully crafted for the extraction of embeddings from images. Impressively, the framework demonstrates the capability to capture spatiotemporal information [59] and extends seamlessly to various domains such as 3D space [60] and multiple scales [61]. Notably, akin masking strategies, prevalent in the Natural Language Processing (NLP) community [62], have also been transposed into the tabular data landscape [46], [47]. Furthermore, VIME [48] presents a method reminiscent of MAE in the tabular data context. VIME perturbs and encodes each data sample within the feature space through the involvement of two estimators. Subsequently, these estimators use decoders to reconstruct both a binary mask and the original, uncorrupted data samples, demonstrating versatility in information extraction.

The second category predominantly revolves around the contrastive learning paradigm and strategically employs data augmentation techniques. Within this domain, prominent models harnessed momentum-update strategies [63], [64], embraced the concept of large batch sizes [65], incorporated stop-gradient operations [66], spatiotemporal information [67], or even introduced an online network tasked with predicting the output of a target network [68]. Notably, these concepts, initially designed for image data, have gracefully transcended into the arena of tabular data. An exemplar of such adaptation is found in SCARF [49], which ingeniously incorporates the principles of SimCLR [65] to pre-train the encoder. This pre-training procedure employs a subset of feature corruption as a pivotal data augmentation method. Furthermore, the work of [50] exemplifies a contrastive framework tailored to tabular data, introducing SAINT, computing both column- and row-wise attentions.

D. Regularization

Regularization techniques, pivotal in machine learning and statistical modeling, mitigate overfitting and enhance generalization by introducing penalty terms into the loss function. Early approaches such as Ridge Regression, which applies L2 regularization to linear models [69], and Lasso Regression [70], which implements L1 regularization, paved the way for modern regularization methods. The Elastic Net [71] combines these approaches to strike a balance between feature selection and coefficient shrinkage, while Dropout [72] and Batch Normalization [73] cater specifically to neural networks, fostering robust and generalized representations. Other techniques like early stopping [74] and weight decay [75] further complement the regularization arsenal. Bayesian approaches introduce probabilistic frameworks, such as Bayesian regression [76] and Gaussian Processes [77], integrating prior beliefs and data likelihood. Recent trends encompass adversarial training [78] to enhance model robustness and graph regularization techniques [79] for graph-based data modeling tasks. As machine learning continues to advance, regularization remains vital for model generalization and robustness.

E. Model Explanation

Model explanation in machine learning [80] is a critical area of research aimed at making the behavior of complex models transparent, understandable, and accountable. Various techniques have been designed to elucidate how predictive models reach their conclusions. Traditional models [81] can leverage decision tree structures and calculate feature importance to understand the decision weights. For deep learning models, to unravel the black-box nature and allow better comprehension of model decisions, various explanation methods [82] are developed to make specific aspects of the internal representations and information flow in neural network interpretable by humans. Among the methods, SHapley Additive exPlanations (SHAP) values [83] stand out with foundation in cooperative game theory, offering a robust and principled approach to attribute the impact of each feature on model predictions.

III. Method

In this section, we present EConTab, our comprehensive approach for tabular data self- and supervised representation learning. First, we outline the process of the regularization method. Second, we formulate the feature corruption process. The self-supervised training process is illustrated in the third sub-section, without knowing the task labels. The fourth sub-section elucidates our novel supervised training method, wherein we leverage labels for contrastive learning. Finally, we expound on our utilization of pre-trained encoders and embeddings to improve downstream tasks and develop the model explanation process.

A. Regularization

We apply regularization [24], [84] on the input layer by introducing a penalty term $\lambda\|\boldsymbol{W}\|_p$ into the loss function, where $\boldsymbol{W}$ represents the input weights, λ is the regularization parameter and p is the specific norm for the penalty. The idea behind is to prevent similar features from weighing too much in loss objective and to learn more robust representation, especially when highly correlated features are present. For example, if we can reconstruct features A, B, and C with only feature A, then B and C should be assigned less weights.

B. Feature Corruption

It's common for the generative-based representation approach to use data augmentation techniques to generate robust feature embeddings. One of the most promising approaches is feature corruption, which has also been used in this paper to enhance our model's performance. Considering the original dataset $\mathcal{X} \subseteq \mathbb{R}^M$, given any tabular data point x_i, we have its j-th feature as x_{i_j}, where $x_i = (x_{i_1}, x_{i_2}, ..., x_{i_M}), j \subseteq M$, with M representing the dimension of features and i denoting the sample index. In our approach, for each sample, we stochastically select t features from the pool of M features and replace them with corrupted features denoted as c. To elaborate, we generate c from the distribution $\widehat{\mathcal{X}}i_j$, where $\widehat{\mathcal{X}}i_j$ represents the uniform distribution over $\mathcal{X}_{i_j} = \left\{x_{i_j} : x_i \in \mathcal{X}\right\}$.

C. Self-supervised Learning

Self-supervised learning of EConTab aims to learn informative representations from unlabeled data (Algorithm 1). For each of the two data samples, x_1 and x_2, we apply input weights and add feature corruption to obtain corrupted data. Then we encode the corrupted data using an encoder, f, resulting in two features, z_1 and z_2. The decoder d will decode the learned embeddings to reconstruct $\hat{x}_1$ and $\hat{x}_2$ respectively, from where we can define the reconstruction loss $\mathcal{L}_{\text{reconstruction}}$ for two samples x^1 and x^2 as the mean squared error (MSE) between input features and reconstructions, shown as:

$$\mathcal{L}_{\text{reconstruction}} = \frac{1}{M}\sum_{j=1}^{M}(x_{1_j} - \hat{x}_{1_j})^2 + \frac{1}{M}\sum_{j=1}^{M}(x_{2_j} - \hat{x}_{2_j})^2. \tag{1}$$

Algorithm 1 Self-supervised learning

Require: unlabeled data $\mathcal{X} \subseteq \mathbb{R}^M$, batch size B, encoder f, decoder d, mean squared error (MSE), input weights $\boldsymbol{W} \subseteq \mathbb{R}^M$, regularization parameter λ and specific norm for penalty p.
for two sampled mini-batch $\left\{x_i^1, y_i^1\right\}_{i=1}^B \subseteq \{\mathcal{X}, \mathcal{Y}\}$ and $\left\{x_i^2, y_i^2\right\}_{i=1}^B \subseteq \{\mathcal{X}, \mathcal{Y}\}$ **do**
  for each sample x_i^1 and x_i^2,
  apply input weights $x_i^1 = x_i^1\boldsymbol{W}$, $x_i^2 = x_i^2\boldsymbol{W}$, for $i \in [B]$
  apply feature corruption, define the corrupted feature as: $\breve{x}_i^1$ and $\breve{x}_i^2$, for $i \in [B]$
  data encoding:
    $z_i^1 = f(\breve{x}_i^1)$, $z_i^2 = f(\breve{x}_i^2)$, for $i \in [B]$
  data reconstruction:
    $\hat{x}_i^1 = d(z_i^1)$, $\hat{x}_i^2 = d(z_i^2)$, for $i \in [B]$
  define reconstruction loss $\mathcal{L}_{\text{reconstruction}} = \text{MSE}(x_i^1, \hat{x}_i^1)+\text{MSE}(x_i^2, \hat{x}_i^2)$
  define penalty as $\lambda\|\boldsymbol{W}\|_p$
  update encoder f and decoder d to minimize $\mathcal{L}_{\text{reconstruction}}$ and $\lambda\|\boldsymbol{W}\|_p$ using RMSProp.
end for

Therefore, the loss function for self-supervised learning can be defined as:

$$\mathcal{L}_{\text{self}} = \mathcal{L}_{\text{reconstruction}} + \lambda\|\boldsymbol{W}\|_p, \tag{2}$$

D. Supervised Learning with Labels

We further improve the pre-training process through supervised learning with labels to take advantage of labeled data, as shown in Figure 1. In self-supervised learning, we only compute the MSE between reconstructed data and original data as the reconstruction loss $\mathcal{L}_{\text{reconstruction}}$. With labels introduced, we can pose additional constraints to the encoded embeddings z_1 and z_2. One is for label prediction to compute the prediction loss (illustrated by classification loss $\mathcal{L}_{\text{classification}}$ through the context). To be specific, z_1 and z_2 are fed to the same multi-layer perceptron (MLP) that maps from the embedding space to the label space. We can also define the cross-entropy loss for classification task as:

$$\mathcal{L}_{\text{classification}} = -\left(y_1 \log(\hat{y}_1) + y_2 \log(\hat{y}_2)\right), \tag{3}$$

where $\hat{y}_1$ and $\hat{y}_2$ are predicted labels computing a MLP, i.e., $\hat{y}_1 = \text{MLP}(z_1)$ and $\hat{y}_2 = \text{MLP}(z_2)$.

1) Contrastive Loss: We further introduce the contrastive loss $\mathcal{L}_{\text{contrastive}}$ in the loss function by forming contrastive pairs (z_1, z_2) of embeddings in the bottleneck layer with respect to the classification labels (y_1, y_2). With this constraint, the model is enforced to maximize the similarity between embeddings with the same label and minimize the similarity between embeddings with different labels, thus capturing the discriminative features for the classification labels and better aligning with downstream tasks. Algorithm 2 formally defines the contrastive loss in the proposed model, which is a variation from the original contrastive learning [85] and relevant to these extensions [65], [86], [87].

Algorithm 2 Contrastive Loss for supervised learning with labels

Require: data embeddings $\mathcal{Z}$ from unlabeled data $\mathcal{X} \subseteq \mathbb{R}^M$, binary labels $\mathcal{Y} \subseteq \mathbb{R}$, batch size B, encoder f, decoder d, contrastive loss margin m, distance function $D(\cdot)$

for two sampled mini-batch $\left\{z_i^1, y_i^1\right\}_{i=1}^B \subseteq \{\mathcal{X}, \mathcal{Y}\}$ and $\left\{z_i^2, y_i^2\right\}_{i=1}^B \subseteq \{\mathcal{X}, \mathcal{Y}\}$ **do**
  for each sample embedding z_i^1 and z_i^2,
  define contrastive loss:
  for $i = 1$ to B **do**
    if $y_i^1 = y_i^2$ **then**
      $y_i^{12} = 1$ for the pair (z_i^1, z_i^2) // z_i^1 is deemed similar to z_i^2
    else
      $y_i^{12} = 0$ for the pair (z_i^1, z_i^2) // z_i^1 is deemed dissimilar to z_i^2
    end if
    $d_i = D(z_i^1, z_i^2)$ // calculate the distance of embeddings pair
    $c_i = (y_i^{12})\frac{1}{2} d_i^2 + (1 - y_i^{12})\frac{1}{2} \max(0, m - d_i)^2$ // calculate the contrastive loss of the pair
  end for
  $\mathcal{L}_{contrastive} = \frac{1}{B} \sum c_i$
  update encoder f and decoder d to minimize $\mathcal{L}_{contrastive}$ using RMSProp.
end for

During the optimization stage, we combine the two additional losses with the self-supervised learning loss $\mathcal{L}_{\text{self}}$ and define the supervised learning loss function $\mathcal{L}_{\text{sup}}$ as follows:

$$\mathcal{L}_{\text{sup}} = \mathcal{L}_{\text{self}} + \alpha * \mathcal{L}_{\text{classification}} + \beta * \mathcal{L}_{\text{contrastive}}, \tag{4}$$

where α and β are used to seek balance among multiple losses and set to 1 as default, respectively.

E. Downstream Fine-Tuning

Drawing inspiration from established representation learning paradigms [49], [63]–[65], we embrace an end-to-end fine-tuning strategy for the pre-trained encoder f from EConTab, utilizing the complete labeled dataset. This approach entails the seamless integration of the encoder with an additional linear layer, thereby granting the flexibility to unlock and adapt all its parameters to align with the specific requirements of downstream supervised tasks. Additionally, we can harness the potential of the salient feature s as a versatile plug-and-play embedding. Through the fusion of z with its original counterpart x, we construct enriched data points. This innovative approach serves to amplify inherent data characteristics, thereby assisting in the establishment of distinct decision boundaries. As a result, we anticipate notable enhancements in classification tasks when employing the concatenated features as the input for conventional models like Random Forest or LightGBM.

F. Model Explanation

There are three major considerations in model explanation for EConTabs. First, the feature weights are learnable through training with regularization, which could explain the importance of input features for overall training tasks. Second, the model structure, especially the encoder part, could be further explained through Deep SHAP [88], which could better understand how the latent representations are formulated from inputs. Third, when latent representations are used in downstream tasks as features, the feature importance could be leverage to explain the final predictions.

IV. Experiments and Results

In this section, we present the results of our extensive experiments conducted on diverse public datasets to highlight the effectiveness of our proposed method, EConTab. This section is structured into two parts for clarity and comprehensiveness. In the first part, we provide essential details regarding the experiments. This includes information about the public datasets for experiments, the preprocessing steps applied to these datasets, the architecture of our models, and specific training procedures. This transparency ensures the reproducibility of our findings.

In the second part, we assess the performance of EConTab through various empirical studies. We conduct a thorough comparison between EConTab and mainstream deep learning methods as well as traditional methods. Meanwhile, we showcase the versatility of EConTab using it as an automatic feature engineering tool. Specifically, we demonstrate how EConTab can enhance the performance of traditional models such as XGBoost, Random Forest, and LightGBM by seamlessly integrating its salient features as plug-and-play embeddings, as shown in Figure 2. This strategy simplifies the feature engineering process and eliminates additional complexity in traditional models training. In addition, we also conducted ablation studies and illustrate the model explanation process.

A. Preliminaries for Experiments

1) Public Datasets: We evaluate the performance of EConTab on a standard benchmark from [50], including Bank (BK) [89], Blastchar (BC) [90], Arrhythmia (AT) [91], Arcene (AR) [92], Shoppers (SH) [93], Volkert (VO) [94] and MNIST (MN) [95]. Five of the datasets focus on binary classification, and two of them focus on multi-class classification tasks. Importantly, the datasets employed in our experiments exhibit significant diversity. They encompass a wide range of characteristics, including varying sample sizes, ranging from 200 to 495,141 samples, and feature dimensions spanning from 8 to 784, encompassing both categorical and numerical features. Among these datasets, some exhibit missing data, while others are

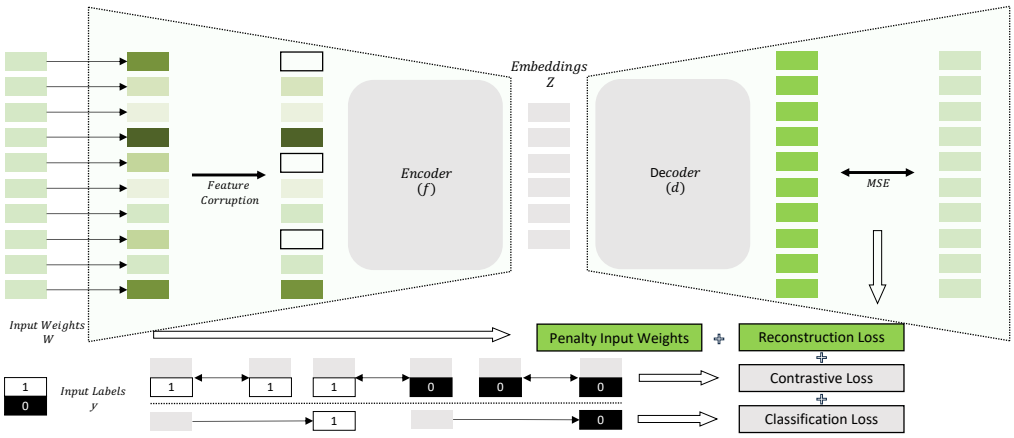


Fig. 1: Proposed AE architecture with contrastive loss and input weights regularization

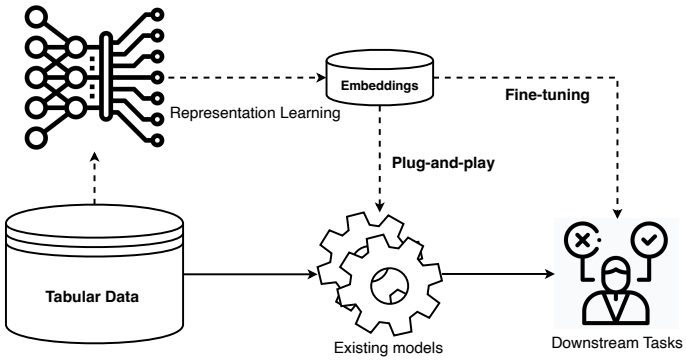


Fig. 2: Illustration of usages of pre-trained encoders and embeddings. 1) The first option could be to fine-tune the pre-trained encoder directly for downstream tasks. This option usually achieves the optimal results but needs additional computation. 2) The second option is to concatenate the pre-trained embeddings with the original datasets, which requires no additional training and computation but still benefits the downstream tasks with considerable improvements in evaluation metrics.

complete, and there is a mix of well-balanced datasets as well as those presenting highly skewed class distributions. This diversity allows us to comprehensively evaluate the performance and robustness of our proposed approach across a spectrum of real-world data scenarios.

2) Preprocessing of Datasets: To handle categorical features, we employ a backward difference encoder as described in [96]. Addressing the issue of missing data, we take a two-step approach. Initially, we remove any features that lack values across all samples. Subsequently, for the remaining missing values, we apply distinct imputation strategies based on the feature type. Numerical features are imputed using the mean value, while categorical features are filled with the most frequent category observed within the dataset. Moreover, we ensure data uniformity by employing a min-max scaler for dataset scaling. In cases involving image-based data, we flatten the images into vectors, treating them akin to tabular data. This approach aligns with established practices found in prior works such as [48] and [50].

3) Model Architectures: The EConTab model architecture features a transformer-based shared network with three layers and two attention heads. This architecture is tailored for processing input data with a dimensionality determined by the shape of the training dataset. Additionally, the decoder remains a one-layer network with a sigmoid activation function. In the downstream fine-tuning stage, we add a linear layer after the encoder f to accommodate classification or regression tasks as needed.

4) Training Details: The EConTab model is trained with a batch size of 128 over 1000 epochs, employing a learning rate of 0.0001. Gaussian masking is applied to the input data with a masking ratio of 0.3. The model's output dimension is set to half of the input data dimension. A contrastive loss with a margin of 2 is used during training, along with L2 normalization. Additionally, a regularization coefficient of 0.01 is applied to introduce a penalty term based on the L2 norm of the standard deviation of the Gaussian mask. During training, data is divided into two batches, and various loss components, including feature reconstruction loss, classification loss, contrastive loss, and regularization penalty, are computed to guide the optimization process. These training configurations ensure effective representation learning while controlling model behavior.

5) Metrics: Given that the majority of the tasks in our analysis involve binary classification, we employ the AUROC (Area Under the Receiver Operating Characteristic curve) as our primary metric for assessing performance. AUROC effectively quantifies the model's ability to distinguish between the two classes in the dataset. However, for the two multi-class datasets, VO and MN, we utilize accuracy on the test set as the metric for comparing performance.

B. Results on the Benchmarks

We show performance comparisons using chosen datasets and present the summarized results in Table I. These results

Dataset size		45211			7043			452			200			12330			58310			518012	
Feature size		16			20			226			783			17			147			54	
Dataset	**BK**			**BC**			**AT**			**AR**			**SH**			**VO★**			**MN★**		
Raw Feature (x)	✓		✓	✓		✓	✓		✓	✓		✓	✓		✓	✓		✓	✓		✓
Distilled Feature (s)		✓	✓		✓	✓		✓	✓		✓	✓		✓	✓		✓	✓		✓	✓
Logistic Reg.	0.907	0.907	0.909	0.892	0.892	0.895	0.862	0.864	0.866	0.916	0.914	0.918	0.870	0.871	0.873	0.539	0.540	0.543	0.899	0.902	0.905
Random Forest	0.891	0.892	0.894	0.879	0.880	0.884	0.850	0.856	0.861	0.809	0.809	0.811	0.929	0.928	0.930	0.663	0.665	0.669	0.938	0.938	0.942
XGboost	0.929	0.928	0.930	0.906	0.903	0.906	0.870	0.871	0.883	0.824	0.822	0.826	0.925	0.925	0.927	**0.690**	0.690	0.692	0.958	0.959	0.963
LightGBM	**0.939**	0.933	0.939	0.910	0.909	0.912	0.887	0.888	0.907	0.821	0.822	0.825	**0.932**	0.933	0.936	0.679	0.680	0.682	0.952	0.953	0.954
CatBoost	0.925	0.928	0.932	0.912	0.910	0.914	0.879	0.880	0.889	0.825	0.827	0.833	0.931	0.932	0.935	0.664	0.665	0.670	0.956	0.958	0.968
MLP	0.915	0.919	0.920	0.892	0.893	0.898	0.902	0.904	0.908	0.903	0.904	0.904	0.887	0.887	0.890	0.631	0.631	0.636	0.939	0.940	0.940
VIME	0.766	-	-	0.510	-	-	0.653	-	-	0.610	-	-	0.744	-	-	0.623	-	-	0.958	-	-
TabNet	0.918	-	-	0.796	-	-	0.521	-	-	0.541	-	-	0.914	-	-	0.568	-	-	**0.968**	-	-
TabTransformer	0.913	-	-	0.817	-	-	0.700	-	-	0.868	-	-	0.927	-	-	0.580	-	-	0.887	-	-
EConTab(Self-Sup.)	0.908	-	-	0.898	-	-	0.873	-	-	0.887	-	-	0.920	-	-	0.619	-	-	0.956	-	-
EConTab(Sup.)	0.929	-	-	**0.913**	-	-	**0.907**	-	-	**0.918**	-	-	0.931	-	-	0.680	-	-	**0.968**	-	-

" – " indicates the experiments are not applicable for the corresponding methods to demonstrate the benefits of plug-and-play embeddings.

TABLE I: Comparison of different methods on the classification tasks. For each method and dataset, we report three categories 1) raw features only, 2) salient features only, 3) plug-and-play features. The best results are shown in **Bold**, second-best results are Underlined. Columns added with ★ are multi-class classification tasks, reporting accuracy. The other results of binary classification tasks are evaluated with AUROC.

encompass evaluations employing both traditional models and more recent deep-learning techniques. In the majority of cases, EConTab exhibits remarkable improvements, outperforming all baseline methods and reaffirming its superiority across a range of datasets and scenarios. However, it is important to note that, on BK, SH, and VO datasets, EConTab achieved suboptimal results when compared to the best models. This observation aligns with previous research conclusions that the tabular domain presents unique challenges, with no single method universally excelling [36]. Nonetheless, EConTab still gives the best performance over all of the deep-learning-based models and the second-best results over all of the methods. Meanwhile, this outcome warrants further investigation to uncover the specific factors contributing to this variation in performance.

C. Results as Plug-and-Play Embeddings

As previously mentioned, EConTab has learned features that can significantly impact the decision boundaries in classification tasks. In the plug-and-play setting from Figure 2, our experimental results demonstrate the immense value of integrating these salient features with the original data as additional features. To be more specific, the performance of traditional models obtains relatively marginal improvement with only distilled features, as shown in the light gray columns of Table I. While the improvement is relatively modest, it aligns with our expectations. The potential absence of original information in this scenario results in a less substantial performance boost. Larger gains without fine-tuning come from the concatenation of original and distilled features. Notably, this integration enhances the performance of every method, leading to improvements in evaluation metrics (e.g., AUROC) across various datasets, as shown in the dark gray columns of Table I.

D. Ablation Studies

In this section dedicated to ablation studies, we delve into the crucial components of EConTab, assessing the significance of the parameter, i.e., feature corruption rate. Our analysis encompasses all the datasets listed in Table I, employing consistent data preprocessing and optimization strategies throughout the experiments. In Table II, we thoroughly examine the most advantageous feature corruption ratio. After extensive analysis, we find that the optimal corruption ratio is approximately 0.3. Therefore, we've adopted this value as the default for all previously reported experiments. However, it's important to emphasize that this chosen ratio may not always be the best fit for every dataset. Additionally, we've noticed interesting patterns in the datasets themselves. Datasets with more complex features, like VO or MN, tend to benefit from larger corruption ratios because they often contain redundant features. This observation aligns with previous research discussed in [81] regarding tabular data. On the flip side, for datasets with simpler, lower-dimensional features like BC, using smaller corruption ratios in our experiments might lead to better results.

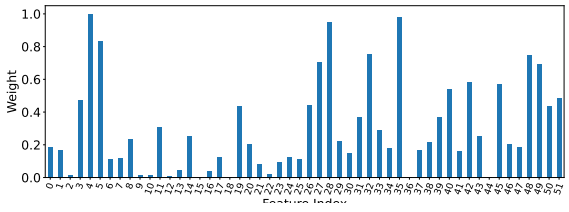
Ratio	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**
BK	0.918	0.920	0.928	**0.929**	0.922	0.917	0.881
BC	0.889	0.897	0.906	**0.913**	0.910	0.901	0.896
AT	0.889	0.894	0.901	**0.905**	0.903	0.890	0.884
AR	0.904	0.911	0.913	**0.918**	0.915	0.909	0.901
SH	0.902	0.914	0.924	**0.931**	0.920	0.909	0.904
VO★	0.667	0.674	0.676	0.680	**0.681**	0.670	0.663
MN★	0.935	0.942	0.951	**0.959**	**0.959**	0.941	0.932

TABLE II: Ablation of corruption ratio. Columns added with ★ are multi-class classification tasks, reporting their accuracy. The other results of binary classification tasks are evaluated with AUC.

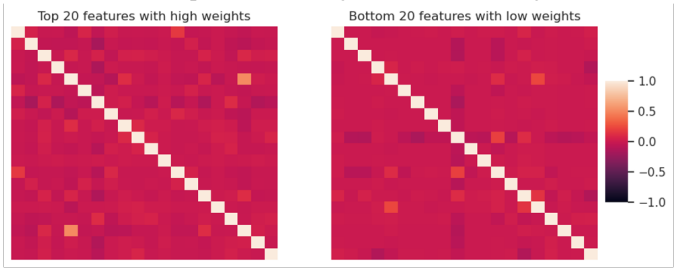
E. Model Explanation Illustration

To illustrate how the model explanation can be conducted, we provide visualizations on all three major steps, including input feature weights, encoder SHAP values, and downstream embedding feature importance. We use BK data as an example, which has 16 features and 10 of them are categorical. After one-hot encoding on categorical features and , the input feature dimension becomes 52. With input feature weights regularization, the contribute of each input feature on overall training process can be cleanly identified, as shown in Figure 3a. We could further compare the feature correlation based on the learned weights. From Figure 3b, the top 20 features with higher

weights tend to be less correlated with each other and represent more independent contributions. In the contrast, bottom 20 features with lower weights are having various correlations among each other. This could be used to unveil the shadow on how the features are learned through their interrelationships.



(a) Input feature weights after fine-tuning



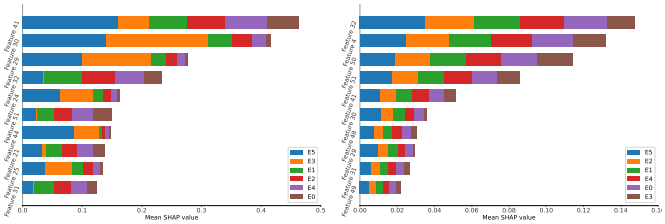
(b) Feature correlation comparison by weights

Fig. 3: Input feature determination through weights regularization

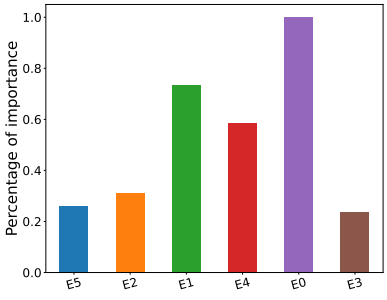
Using SHAP values for EConTab encoder could better understand how each input feature is contributing to the formation of each latent embedding. As shown in Figure 4, the mean SHAP values for top features are visualized and decomposed across all latent embeddings, for both pretraining only and after fine-tuning. Once the latent embeddings are used in downstream models (e.g., XGboost), the feature importance can be calculated and compared against other features if used in a plug-and-play manner, as shown in Figure 4c.

V. Conclusion

As we observe the evolution of potent representation learning techniques tailored for different types of data from computer vision and natural language processing, we embark on a journey to extend their remarkable performance into new domains, such as tabular data. Drawing inspiration from related endeavors that address this challenge from the vantage points of contrastive learning and generative modeling, we present EConTab — an innovative self- and supervised framework designed for representation learning and feature distillation. The features learned through EConTab exhibit superior performance in downstream tasks, obviating the need for extensive exploration of hand-crafted features. Furthermore, these features manifest as discernible, low-dimensional representations that seamlessly enhance the capabilities of various traditional models with explainability. We hold a strong conviction that this research marks a pivotal milestone in the pursuit of more representative, efficient, and structured representations for tabular data.



(a) Pretraining only (b) Pretraining and Finetuning



(c) Feature Importance

Fig. 4: Mean SHAP values for each input feature to contribute on each latent embedding

References

[1] A. Qayyum, J. Qadir, M. Bilal, and A. Al-Fuqaha, "Secure and robust machine learning for healthcare: A survey," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 156–180, 2020.

[2] S. Chen, W. D. Kearns, J. L. Fozard, and M. Li, "Personalized fall risk assessment for long-term care services improvement," in *2017 Annual Reliability and Maintainability Symposium (RAMS)*. IEEE, 2017, pp. 1–7.

[3] S. Chen, N. Kong, X. Sun, H. Meng, and M. Li, "Claims data-driven modeling of hospital time-to-readmission risk with latent heterogeneity," *Health care management science*, vol. 22, pp. 156–179, 2019.

[4] S. Chen, X. Liu, Y. Li, J. Wu, and H. Yao, "Deep representation learning for multi-functional degradation modeling of community-dwelling aging population," *arXiv preprint arXiv:2404.05613*, 2024.

[5] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[6] S. Chen, L. Lu, and M. Li, "Multi-state reliability demonstration tests," *Quality Engineering*, vol. 29, no. 3, pp. 431–445, 2017.

[7] B. Wang, L. Lu, S. Chen, and M. Li, "Optimal test design for reliability demonstration under multi-stage acceptance uncertainties," *Quality Engineering*, vol. 0, no. 0, pp. 1–14, 2023. [Online]. Available: https://doi.org/10.1080/08982112.2023.2249188

[8] S. Chen, "Some recent advances in design of bayesian binomial reliability demonstration tests," *Doctoral dissertation, University of South Florida*, 2020. [Online]. Available: https://digitalcommons.usf.edu/etd/8170

[9] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, p. 2674, 2018.

[10] J. Wu, R. Tao, P. Zhao, N. F. Martin, and N. Hovakimyan, "Optimizing nitrogen management with deep reinforcement learning and crop simulations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1712–1720.

[11] R. Tao, P. Zhao, J. Wu, N. F. Martin, M. T. Harrison, C. Ferreira, Z. Kalantari, and N. Hovakimyan, "Optimizing crop management with reinforcement learning and imitation learning," *arXiv preprint arXiv:2209.09991*, 2022.

[12] D. Xu, S. Hu, D. Zhang, Y. Xiong, Y. Yang, and Y. Ran, "Importance of sporopollenin structure and accessibility in the sorption of phenanthrene by biota spores and pollens," *Environmental science & technology*, vol. 53, no. 24, pp. 14 285–14 295, 2019.

[13] J. Ye, B. Jahannia, H. Kang, H. Wang, E. Heidari, N. Asadizanjani, V. J. Sorger, and H. Dalir, "Oam modes classification and demultiplexing via fourier optical neural network," in *Complex Light and Optical Forces XVIII*, vol. 12901. SPIE, 2024, pp. 44–52.

[14] J. Ye, H. Kang, H. Wang, S. Altaleb, E. Heidari, N. Asadizanjani, V. J. Sorger, and H. Dalir, "Multiplexed oam beams classification via fourier optical convolutional neural network," in *2023 IEEE Photonics Conference (IPC)*. IEEE, 2023, pp. 1–2.

[15] J. Ye, H. Kang, H. Wang, C. Shen, B. Jahannia, E. Heidari, N. Asadizanjani, M.-A. Miri, V. J. Sorger, and H. Dalir, "Demultiplexing oam beams via fourier optical convolutional neural network," in *Laser Beam Shaping XXIII*, vol. 12667. SPIE, 2023, pp. 16–33.

[16] J. Ye, H. Kang, Q. Cai, Z. Hu, M. Solyanik-Gorgone, H. Wang, E. Heidari, C. Patil, M.-A. Miri, N. Asadizanjani *et al.*, "Multiplexed orbital angular momentum beams demultiplexing using hybrid optical-electronic convolutional neural network," *Nature Communications Physics*, vol. 7, no. 1, p. 105, 2024.

[17] S. Chen, L. Lu, Q. Zhang, and M. Li, "Optimal binomial reliability demonstration tests design under acceptance decision uncertainty," *Quality Engineering*, vol. 32, no. 3, pp. 492–508, 2020.

[18] L. Gao, G. Cordova, C. Danielson, and R. Fierro, "Autonomous multi-robot servicing for spacecraft operation extension," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 10 729–10 735.

[19] Y. Zhang, X. Wang, L. Gao, and Z. Liu, "Manipulator control system based on machine vision," in *International Conference on Applications and Techniques in Cyber Intelligence ATCI 2019: Applications and Techniques in Cyber Intelligence 7*. Springer, 2020, pp. 906–916.

[20] L. Gao, K. Aubert, D. Saldana, C. Danielson, and R. Fierro, "Decentralized adaptive aerospace transportation of unknown loads using a team of robots," *arXiv preprint arXiv:2407.08084*, 2024.

[21] L. Gao, C. Danielson, and R. Fierro, "Adaptive robot detumbling of a non-rigid satellite," *arXiv preprint arXiv:2407.17617*, 2024.

[22] S. Chen, L. Lu, Y. Xiang, Q. Lu, and M. Li, "A data heterogeneity modeling and quantification approach for field pre-assessment of chloride-induced corrosion in aging infrastructures," *Reliability Engineering & System Safety*, vol. 171, pp. 123–135, 2018.

[23] S. Liu and M. Zhu, "Meta inverse constrained reinforcement learning: Convergence guarantee and generalization analysis," in *The Twelfth International Conference on Learning Representations*, 2023.

[24] I. Covert, U. Sumbul, and S.-I. Lee, "Deep unsupervised feature selection," ' ', 2019.

[25] J. Ye, H. Kang, H. Wang, S. Altaleb, E. Heidari, N. Asadizanjani, V. J. Sorger, and H. Dalir, "Oam beams multiplexing and classification under atmospheric turbulence via fourier convolutional neural network," in *Frontiers in Optics*. Optica Publishing Group, 2023, pp. JTu4A–73.

[26] J. Ye, M. Solyanik, Z. Hu, H. Dalir, B. M. Nouri, and V. J. Sorger, "Free-space optical multiplexed orbital angular momentum beam identification system using fourier optical convolutional layer based on 4f system," in *Complex Light and Optical Forces XVII*, vol. 12436. SPIE, 2023, pp. 69–79.

[27] L. Yu, C. Li, L. Gao, B. Liu, and C. Che, "Stochastic analysis of touch-tone frequency recognition in two-way radio systems for dialed telephone number identification," in *2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE)*. IEEE, 2024, pp. 1565–1572.

[28] R. E. Wright, "Logistic regression." ' ', 1995.

[29] T. J. Hastie and D. Pregibon, "Generalized linear models," in *Statistical models in S*. Routledge, 2017, pp. 195–247.

[30] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[31] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[32] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[33] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," *Advances in neural information processing systems*, vol. 31, 2018.

[34] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

[35] S. Liu and M. Zhu, "Distributed inverse constrained reinforcement learning for multi-agent systems," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33 444–33 456, 2022.

[36] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting deep learning models for tabular data," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 932–18 943, 2021.

[37] S. Liu and M. Zhu, "Learning multi-agent behaviors from distributed and streaming demonstrations," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[39] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," *Advances in neural information processing systems*, vol. 30, 2017.

[40] W. Song, C. Shi, Z. Xiao, Z. Duan, Y. Xu, M. Zhang, and J. Tang, "Autoint: Automatic feature interaction learning via self-attentive neural networks," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1161–1170.

[41] R. Wang, R. Shivanna, D. Cheng, S. Jain, D. Lin, L. Hong, and E. Chi, "Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems," in *Proceedings of the web conference 2021*, 2021, pp. 1785–1797.

[42] S. Popov, S. Morozov, and A. Babenko, "Neural oblivious decision ensembles for deep learning on tabular data," *arXiv preprint arXiv:1909.06312*, 2019.

[43] S. Badirli, X. Liu, Z. Xing, A. Bhowmik, K. Doan, and S. S. Keerthi, "Gradient boosting neural networks: Grownet," *arXiv preprint arXiv:2002.07971*, 2020.

[44] G. Ke, J. Zhang, Z. Xu, J. Bian, and T.-Y. Liu, "Tabnn: A universal neural network solution for tabular data," ' ', 2018.

[45] G. Ke, Z. Xu, J. Zhang, J. Bian, and T.-Y. Liu, "Deepgbm: A deep learning framework distilled by gbdt for online prediction tasks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 384–394.

[46] S. Ö. Arik and T. Pfister, "Tabnet: Attentive interpretable tabular learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 6679–6687.

[47] X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, "Tabtransformer: Tabular data modeling using contextual embeddings," *arXiv preprint arXiv:2012.06678*, 2020.

[48] J. Yoon, Y. Zhang, J. Jordon, and M. van der Schaar, "Vime: Extending the success of self-and semi-supervised learning to tabular domain," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 033–11 043, 2020.

[49] D. Bahri, H. Jiang, Y. Tay, and D. Metzler, "Scarf: Self-supervised contrastive learning using random feature corruption," *arXiv preprint arXiv:2106.15147*, 2021.

[50] G. Somepalli, M. Goldblum, A. Schwarzschild, C. B. Bruss, and T. Goldstein, "Saint: Improved neural networks for tabular data via row attention and contrastive pre-training," *arXiv preprint arXiv:2106.01342*, 2021.

[51] J. Wu, S. Chen, Q. Zhao, R. Sergazinov, C. Li, S. Liu, C. Zhao, T. Xie, H. Guo, C. Ji *et al.*, "Switchtab: Switched autoencoders are effective tabular learners," *arXiv preprint arXiv:2401.02013*, 2024.

[52] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1920–1929.

[53] X. Li, Y. Guo, and D. Schuurmans, "Semi-supervised zero-shot classification with label representation learning," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4211–4219.

[54] J. Wu, Z. Lai, S. Chen, R. Tao, P. Zhao, and N. Hovakimyan, "The new agronomists: Language models are experts in crop management," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5346–5356.

[55] Z. Lai, X. Zhang, and S. Chen, "Adaptive ensembles of fine-tuned transformers for llm-generated text detection," *arXiv preprint arXiv:2403.13335*, 2024.

[56] Z. Lai, J. Wu, S. Chen, Y. Zhou, and N. Hovakimyan, "Residual-based language models are free boosters for biomedical imaging tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5086–5096.

[57] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[58] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the*

IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 16 000–16 009.

[59] C. Feichtenhofer, Y. Li, K. He *et al.*, "Masked autoencoders as spatiotemporal learners," *Advances in neural information processing systems*, vol. 35, pp. 35 946–35 958, 2022.

[60] J. Jiang, X. Lu, L. Zhao, R. Dazeley, and M. Wang, "Masked autoencoders in 3d point cloud representation learning," *arXiv preprint arXiv:2207.01545*, 2022.

[61] C. J. Reed, R. Gupta, S. Li, S. Brockman, C. Funk, B. Clipp, S. Candido, M. Uyttendaele, and T. Darrell, "Scale-mae: A scale-aware masked autoencoder for multiscale geospatial representation learning," *arXiv preprint arXiv:2212.14532*, 2022.

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

[63] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[64] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," *arXiv preprint arXiv:2003.04297*, 2020.

[65] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[66] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 750–15 758.

[67] J. Wu, D. Pichler, D. Marley, D. Wilson, N. Hovakimyan, and J. Hobbs, "Extended agriculture-vision: An extension of a large aerial image dataset for agricultural pattern analysis," *arXiv preprint arXiv:2303.02460*, 2023.

[68] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[69] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 regularization for learning kernels," *arXiv preprint arXiv:1205.2653*, 2012.

[70] L. Meier, S. Van De Geer, and P. Bühlmann, "The group lasso for logistic regression," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 70, no. 1, pp. 53–71, 2008.

[71] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 67, no. 2, pp. 301–320, 2005.

[72] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[73] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.

[74] L. Prechelt, "Early stopping-but when?" in *Neural Networks: Tricks of the trade*. Springer, 2002, pp. 55–69.

[75] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[76] C. M. Bishop, M. E. Tipping *et al.*, "Bayesian regression and classification," *Nato Science Series sub Series III Computer And Systems Sciences*, vol. 190, pp. 267–288, 2003.

[77] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer school on machine learning*. Springer, 2003, pp. 63–71.

[78] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[79] F. Feng, X. He, J. Tang, and T.-S. Chua, "Graph adversarial training: Dynamically regularizing based on graph structure," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2493–2504, 2019.

[80] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[81] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?" *Advances in Neural Information Processing Systems*, vol. 35, pp. 507–520, 2022.

[82] G. Ras, M. van Gerven, and P. Haselager, "Explanation methods in deep learning: Users, values, concerns and challenges," *Explainable and interpretable models in computer vision and machine learning*, pp. 19–36, 2018.

[83] A. Das and P. Rad, "Opportunities and challenges in explainable artificial intelligence (xai): A survey," *arXiv preprint arXiv:2006.11371*, 2020.

[84] Y. Wu, D. Zhu, and X. Wang, "Contrastive learning enhanced deep neural network with serial regularization for high-dimensional tabular data," *Expert Systems with Applications*, vol. 228, p. 120243, 2023.

[85] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[86] S. Tao, P. Peng, and H. Wang, "Supervised contrastive learning with tpe-based bayesian optimization of tabular data for imbalanced learning," *arXiv preprint arXiv:2210.10824*, 2022.

[87] Z. Gharibshah and X. Zhu, "Local contrastive feature learning for tabular data," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 3963–3967.

[88] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International conference on machine learning*. PMLR, 2017, pp. 3145–3153.

[89] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing," *Decision Support Systems*, vol. 62, pp. 22–31, 2014.

[90] IBM, "Telco customer churn (11.1.3+)," 2019. [Online]. Available: https://community.ibm.com/community/user/businessanalytics/blogs/steven-macko/2019/07/11/telco-customer-churn-1113

[91] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[92] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.

[93] C. O. Sakar, S. O. Polat, M. Katircioglu, and Y. Kastro, "Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks," *Neural Computing and Applications*, vol. 31, pp. 6893–6908, 2019.

[94] I. Guyon, L. Sun-Hosoya, M. Boullé, H. J. Escalante, S. Escalera, Z. Liu, D. Jajetic, B. Ray, M. Saeed, M. Sebag, A. Statnikov, W. Tu, and E. Viegas, "Analysis of the automl challenge series 2015-2018," in *AutoML*, ser. Springer series on Challenges in Machine Learning, 2019. [Online]. Available: https://www.automl.org/wp-content/uploads/2018/09/chapter10-challenge.pdf

[95] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[96] K. Potdar, T. S. Pardawala, and C. D. Pai, "A comparative study of categorical variable encoding techniques for neural network classifiers," *International journal of computer applications*, vol. 175, no. 4, pp. 7–9, 2017.