A Study on Deep Semi-supervised learning method using Data-adaptive Augmentation Technique

Seri Park

Department of Statistics

The Graduate School of Sungshin Women's University

A Study on Deep Semi-supervised learning method using Data-adaptive Augmentation Technique

A Master's Thesis
Submitted to the
Graduate School of Sungshin Women's University

in partial fulfillment of the requirements
for the degree of
Master of Statistics

Seri Park

November, 2023

This is to certify that we have examined the
Master's Thesis of
Seri Park
Submitted to Department of Statistics

Approved as to style and content:

Thesis Advisor Dongha Kim

Committee Chairman Seongoh Park

Committee Member Heewon Park

Committee Member Kwan-Young Bak

The Graduate School of Sungshin Women's University

Abstract

This study introduces a novel semi-supervised learning approach specifically designed for tabular data, featuring a unique learnable data augmentation technique that preserves the labeled data' information. The approach is mainly motivated by two methods: MixMatch, known as one of the state-of-the-art semi-supervised learning methods in image data, and Neutral AD, a self-supervised learning method for anomaly detection. These inspirations are adapted to tabular data through an innovative loss function comprising three distinct parts: one for labeled data, one for unlabeled data, and another for deterministic contrastive learning. This loss function is pivotal in guiding transformations that produce diverse and informative data augmentations, while preserving the characteristics of the original data.

To validate our proposed method, we perform experiments on three tabular datasets, where our method demonstrates remarkable state-of-the-art performance, especially on the two datasets. The results not only show superior test accuracy over several baselines, but also highlight the importance of each components' role by the tuning hyperparameters proposed in the ablation studies.

Contents

Ⅰ. Introduction

The importance of semi-supervised learning in machine learning cannot be overstated. It is a pivotal methodology, mainly when acquiring an extensive amount of labeled data is costly and time-consuming.

Semi-supervised learning becomes crucial as it leverages labeled and unlabeled data to improve learning accuracy. However, most semi-supervised learning research and applications have been predominantly focused on the image domain. Within this domain, various examples of semi-supervised learning exist, but the data augmentation technique is particularly paramount. Data augmentation in image processing involves rotation, reflection, and cropping techniques, significantly enriching the dataset without additional labeling.

Despite the success of data augmentation techniques in the image domain, their direct application to the tabular data domain needs to be improved. In the tabular data domain, with the structure of rows and columns retaining the data's inherent meaning, it is challenging when any data augmentations (transformations) are performed.

Gaussian perturbation and random noise injection are employed to augment tabular datasets. Gaussian perturbation involves adding noise drawn from a Gaussian distribution to numerical features, which can help the model generalize better by simulating the variability within the data.

Masking, another common technique, involves randomly replacing specific values in the data with a placeholder or a 'masked' value.

There are also methods that use these data augmentation, such as VIME, which introduces novel learning frameworks for tabular data by creating pretext tasks like estimating mask vectors from corrupted data and data reconstruction.

However data augmentation techniques like masking can affect critical data like a patient's diagnosis by changing datasets (e.g., presence of illness, sugar levels in a diabetes patient) to zero. At the same time, random noise does not consider the distribution of the data, which is particularly problematic when dealing with continuous datasets like debt. These methods can lead to information loss or the production of irrelevant and inaccurate data.

Therefore, for our method, we will make a semi-supervised learning method using data augmentation, and the method of doing it will be made by learning the data augmentation itself.

We were inspired by the two existing methods to develop semi-supervised learning for the tabular data domain. First is Mixmatch (Berthelot et al., 2019), which allows the performance of semi-supervised learning in the image domain. For the tabular data to perform semi-supervised learning, another inspiration was needed. We adapted NeuTraL AD (Qiu et al. 2021), a self-supervised learning method that was originally used to perform anomaly detection. This method allows us to

obtain the embedding vector, providing high-quality augmented tabular data.

By leveraging the NeuTraL AD and MixMatch framework, our method seeks to transform tabular data efficiently in semi-supervised learning scenarios for tabular datasets, mainly focusing on augmenting the data without losing the data information as much as possible.

With our method, one of the advantages is that the performance has increased. Moreover, because it uses data-specific augmentation rather than depending on specific data augmentation, it has the advantage of being more flexible than selecting a specific augmentation methodology. The following chapters will explore the details of this method and show its effectiveness with practical examples.

This paper is constructed as follows. Chapter 2 explains in detail the methodology of semi-supervised, self-supervised, and self- and semi-supervised learning, introducing each of its methods with the related works. Chapter 3 introduces the new model and learning method proposed in this study, explaining each component of the loss function in detail. Chapter 4 shows the results of the experiment performed by the proposed and comparative methods, and compares and analyzes them in terms of the test accuracy. Finally, Chapter 5 discusses the conclusions of this study and future research directions.

Ⅱ. Related Works

1. Semi-Supervised Learning

In machine learning, especially supervised learning, we use supervised learning when the training data set contains labeled input-output pairs $(x_1, y_1), (x_2, y_2), \ldots$. The objective is to utilize the patterns observed in the labeled examples to make predictions or classify the data. The prediction model is supervised during training, as it receives guidance from the correct labels, y. A large amount of labeled data are required to train a model effectively. However, obtaining a labeled data set is often expensive and time-consuming.

Semi-supervised learning is beneficial when access to labeled data is limited but extensive volumes of unlabeled data are available. For unlabeled data to be used meaningfully, several assumptions about data distribution are required.

First is the smoothness assumption. Smoothness assumption is that if the input values x_1 and x_2 in a region with high probability density are close, so should the labels y_1 and y_2 associated with each. Not only is this assumption well applied to supervised learning but also with the semi-supervised learning method imposing this assumption on the unlabeled data.

Say x_1 is labeled data, and x_2 and x_3 are unlabeled data. If x_1 is close

to x_2, and x_2 and x_3 are close, the label of x_1 can be expected to be close to x_3 even when x_1 is not close to x_3.

The second is the low-density assumption. This assumption presumes that the model's decision boundary ideally pass through regions of low data density. In other words, the low-density assumption suggests that the areas where the data is sparse (i.e., where there are few data points) are the most suitable places for the decision boundary. This is because it is less likely for data points from different classes to be mixed in these low-density regions, making it a more natural and effective place to separate the classes. The assumptions can be associated with one another. For instance, if the model is defined according to the low-density assumption, it is less likely to violate the smoothness assumption. In contrast, if a decision boundary is placed in the high-density region with an extensive amount of data, the nearby data are more likely to have the same label, violating the smoothness assumption.

Manifold assumption assumes that high-dimensional input data lies along a manifold in a low-dimensional space, which means that the intrinsic dimensions of the data are much lower than the number of dimensions used to represent them. For example, images of faces might be represented in thousands of dimensions, but changes in facial features that are meaningful for tasks like recognition or expression analysis might exist in much lower-dimensional space. This assumption alleviates the curse of dimensionality by letting the model focus more on the lower-dimensional representation, thereby reducing the problems associated with

high-dimensional space. There are more assumptions regarding semi-supervised learning, such as cluster assumption, where if data belong to the same cluster, the data belong to the same class, and so on.

Entropy minimization and consistency regularization are techniques often used in semi-supervised learning, and they are particularly effective when combined with the low-density separation assumption.
Entropy measures uncertainty in a probabilistic distribution, and in machine learning, it plays a crucial role in classification models. High entropy in a model's prediction for an input datum suggests uncertainty in classifying that datum. This uncertainty could be due to the datum's proximity to the model's decision boundary, where the model is less decisive. Entropy minimization[1] was introduced for the following reason. Suppose the entropy is high when learning the hidden representation of unlabeled data. The performance is degraded in that case because the feature vector will likely be located around the decision boundary with other classes. Therefore, the entropy should be added to the loss function. We aim to enhance the model's confidence by minimizing entropy, ideally positioning data points further from the decision boundary in regions of lower uncertainty. Generally, data points in high-density regions of the feature space are expected to exhibit lower entropy, indicating more precise classification by the model.

1) Grandvalet, Y., &Bengio, Y. (2004). Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, *17*.

Consistency regularization is another crucial technique in machine learning, enhancing the robustness of models against changes in input data. This method is particularly effective in semi-supervised learning scenarios, where it plays a dual role: ensuring stable model performance on varied inputs and leveraging limited labeled data to improve predictions on unlabeled data.

The core principle of consistency regularization is to make constant predictions across the altered versions of the same input. It uses labeled data as a benchmark, guiding the model to generalize this knowledge to unlabeled data. When faced with an unlabeled data point, the model is exposed to several transformed versions created through perturbations and augmentations. Despite these changes, the model must consistently classify these variations according to the patterns learned from the labeled data. For instance, if a labeled datum is classified into a particular category, the model, guided by consistency regularization, aims to categorize both labeled and unlabeled variants of similar data similarly.

To summarize both techniques above, entropy minimization enhances model performance by encouraging confident predictions and reducing uncertainty in its outputs. Consistency regularization focuses on the stability of the model. This technique ensures that the model's predictions remain consistent even with the slightly altered (transformed) input data.

Two techniques, entropy minimization and consistency regularization, can be considered to deal with the model's output behavior and the interaction with the input data. Mixup introduces an approach to augmenting the input data itself.

Mixup is a data augmentation technique used to create new training samples. Originally Mixup is a method where two labeled data are randomly selected and the two are mixed to create new data and use it for learning in an supervised method. It assumes that for a stable model, the predicted value for a linear combination of feature vectors should be a linear combination of labels.
The process involves taking two input samples and their labels, say (x_1, y_1) and (x_2, y_2), and creating a new sample (x', y') as follows:

$$x' = \lambda x_1 + (1-\lambda)x_2$$
$$y' = \lambda \cdot y_1 + (1-\lambda) \cdot y_2,$$

where $\lambda \sim B(\alpha, \alpha)$ is a value for mixing two data at a certain ratio.
Instead of creating loss for x_1, y_1 and x_2, y_2, Mixup creates the loss for x', y' such as cross entropy, L_2 loss.

As mentioned above, the original purpose of using Mixup lies in supervised learning. When it is used in semi-supervised learning, the (sharpened) pseudo-labels play the role of the labels y in supervised learning.

Below, some papers include each method in the algorithm.

MixMatch (Berthelot et al., 2019) is one of the widely used semi-supervised learning framework. It enhances the accuracy of semi-supervised learning models by integrating entropy minimization, consistency regularization, using Mixup. The techniques, in combination, complement each other in semi-supervised learning.

Mixmatch is a method designed to create a classifier within a semi-supervised learning framework, which is particularly useful when dealing with image data. It involves data augmentation and a Mixup technique represented by (2.1) below.

$$MU_{\lambda}(a,b) = (1-\lambda)a + \lambda b, \qquad (2.1)$$

where $\lambda \sim Unif(0,0.5)$. Let $x, x^u \in R^p$ as labeled, unlabeled data respectively.

$$L^{mm} = \widetilde{L}_x + \lambda_u \widetilde{L}_u \qquad (2.2)$$

(2.2) shows the entire loss function of the Mixmatch. It comprises two terms, $\widetilde{L}_x$ for supervised loss and $\widetilde{L}_u$ for the unsupervised loss.

Data augmentation is applied to a batch of labeled and unlabeled samples, resulting in the corresponding augmented versions of the labeled and the K augmented versions of the same data point. Pseudo-labels are used to leverage the unlabeled data. Here, sharpening technique is included. The sharpened pseudo-label is the average of the predictions for the unlabeled samples, $p(x^u)$. It is formulated as below:

$$p(x^u) \propto \left(\frac{1}{K} \sum_{k=1}^{K} f(aug(x^u)) \right)^{1/\tau},$$

where $aug()$ is the data augmentation function, and τ is the temperature parameter.

(2.3) shows the cross-entropy loss, the standard supervised loss.

$$\widetilde{L_x} := \mathop{\mathbb{E}}_{(x,y)\sim L} \mathop{\mathbb{E}}_{x^u \sim U} \mathop{\mathbb{E}}_{k,l \sim Unif1,\ldots,K} \mathop{\mathbb{E}}_{\lambda \sim \mathrm{Unif}(0,0.5)} \left[\mathrm{CE}\left[(\mathrm{MU}_\lambda(\mathrm{y}, \mathrm{p}(\mathrm{x}^u)), \mathrm{f}(\mathrm{MU}_\lambda(\mathrm{aug}(\mathrm{x}), \mathrm{aug}(\mathrm{x}^u))) \right] \right] \quad (2.3)$$

(2.4) shows the L_2 loss for the unlabeled data.

$$\widetilde{L_u} := \mathop{\mathbb{E}}_{x_1^u, x_2^u \sim U} \mathop{\mathbb{E}}_{k,l \sim Unif1,\ldots,K} \mathop{\mathbb{E}}_{\lambda \sim \mathrm{Unif}(0,0.5)} \| (\mathrm{MU}_\lambda(\mathrm{p}(\mathrm{x}_1^u), \mathrm{p}(\mathrm{x}_2^u)) - \mathrm{f}(\mathrm{MU}_\lambda(\mathrm{aug}(\mathrm{x}_1^u), \mathrm{aug}(\mathrm{x}_2^u))) \|_2^2, \quad (2.4)$$

Although Mixup can be conducted in both cross-entropy loss and L_2 loss, it is known that using L_2 loss for term $\widetilde{L_u}$ produces better results.

FixMatch (Sohn et al., 2020) is a related technique to MixMatch, where both methods aim to leverage labeled and unlabeled data to improve model performance. Though the two papers are relevant, FixMatch simplifies the 'data augmentation' step, using the weak augmentations (i.e., random cropping and flipping) to follow the 'teacher' using the strong augmentation. It focuses on the consistency between model predictions on different views of the same data point. FixMatch also introduces pseudo-labeling in assigning labels to the unlabeled data. The data points that surpass the confidence threshold are assigned to the pseudo-labels. Even though this method is derived from the Mixmatch paper, due to the strong augmentation it introduces, it was not an

optimal approach for our semi-supervised learning technique.

Contrastive Mixup (Darabi et al., 2021) introduces a semi-supervised learning method that leverage both labeled and unlabeled data, utilizing the manifold assumption to interpolate data in a latent space, thereby creating meaningful samples for training. This contrasts with traditional data augmentation techniques which are less effective for tabular data. The framework employs a Mixup-based interpolation strategy, focusing on interpolating between same-class samples in the latent space using a supervised contrastive loss term. This method is demonstrated to be effective on both public and clinical datasets, where large annotated datasets are often unavailable.

VAT (*Virtual Adversarial Training*) (Miyato et al., 2018) is designed to force the model to output smooth predictions even with any random perturbations. To achieve this, the VAT introduces "small" virtual adversarial perturbations that maximize model prediction changes. VAT computes the distance of the two prediction distributions between the original input and its corresponding perturbed input by using KL divergence (Kullback-Leibler divergence) as a regularization term. Below is the loss function.

$$L_u^{VAT}(x,\theta) = D[q(y|x_*), p(y|x_* + r_{vadv}, \theta)], \tag{2.5}$$

where $r_{vadv} = \arg max_{r;\|r\| \le \epsilon} D[q(y|x_*), p(y|x_* + r, \theta)]$ is the virtual adversarial

perturbation. $D[\ ,\ ]$ in (2.5) is the KL divergence that quantifies the difference between two probability distributions. r_{vadv} shows the distance between $q(y|x_*)$ and $p(y|x_* + r_{vadv}, \theta)]$. $q(y|x_*)$ is the output distribution given by the current model for input x_*. $p(y|x_* + r_{vadv}, \theta)$ is the output distribution given by the model for the input $x_* + r_{vadv}$, both for parameters θ. The model is trained by using the KL divergence between prediction of normal inputs and input that has been perturbation using r_{adv} as a target function.

It is where the smoothness comes in, helping propagate information from the labeled data set to the unlabeled. The decision boundary of the model then moves toward low-density regions of the data distribution, making it more robust to changes in the input data.

ICT (*Interpolation Consistency Training for semi-supervised learning*) (Verma et al., 2022) employs to train a model to provide consistent predictions at interpolations of unlabeled data points. (2.6) shows the total loss function of this method.

$$L = L_S + wL_U \tag{2.6}$$

The loss function is the sum of supervised learning loss L_S, the cross entropy for the labeled samples, and an unsupervised consistency loss L_U with the weight (w) applied to it shown in (2.7).

$$L_u := \mathop{\mathbb{E}}_{u_j, u_k \sim P_X} \mathop{\mathbb{E}}_{\lambda \sim B(\alpha,\alpha)} l(f_\theta(MU(u_j, u_k)), MU(f_{\theta'}(u_j, f_{\theta'}(u_k)))), \tag{2.7}$$

where $MU_\lambda(a,b) = (1-\lambda)a + \lambda b$. The unsupervised consistency loss L_U is the measure between the model's prediction on the interpolated unlabeled points. ($f_\theta(u_m)$) and the pseudo labels ($y_{\hat{m}}$).

f_θ is the prediction model, u_m is the interpolation of two unlabeled points, u_j and u_k. L_U penalizes the model when the predictions for interpolated unlabeled data points do not match the interpolation of the model's predictions for the original points. When the model enhances the dataset by adding more interpolations and encourages consistency in its prediction it uses Mixup. It suggests that selecting two unlabeled data samples from different classes and applying a Mixup between the two will create interpolated data points near the model's decision boundary, this is shown below in (2.8).

$$f_\theta(MU_\lambda(u_j,u_k)) \approx MU_\lambda(f_{\theta'}(u_j), f_{\theta'}(u_k)), \qquad (2.8)$$

where θ' is the EMA (Exponential Moving Average) of θ, which is used to ensure the predictions for unlabeled data are stable to the noise of iterative updates.

2. Self-supervised learning

In self-supervised learning, the goal is to learn useful representations or features from data without relying on explicit labels. Making an embedding vector, where embedding is a vector representation of the input data that captures its essential characteristics in a lower-dimensional

space, is one of the ways for the model to attain the information well. The model is trained to extract valuable representations from the unlabeled data.

Contrastive learning is a widely representative self-supervised learning technique that trains the models to learn valuable data representations by contrasting similar and dissimilar examples. For each data point, there are pairs of examples: positive pairs and negative pairs. Positive pairs are augmented versions of the same data point, while negative pairs are created from different data points. The model is trained to bring positive pairs closer in the feature space while pushing negative pairs apart.

NeuTraL AD (Qiu et al., 2021) employs self-supervised learning techniques to analyze tabular data for semi-supervised anomaly detection. Unlike the image data, where various data augmentation techniques such as rotation, flipping, cropping, and saturation are applicable, tabular data presents unique challenges for data augmentation. The paper addresses the critical challenge of preserving the essential attributes of tabular data during augmentation, a task effectively tackled using self-supervised learning methods.

The main idea of this paper is embedding the transformed data into a semantic space, ensuring that these data representations are transformed while preserving the essential characteristics of the original data. This method employs contrastive learning, which encourages the model to distinguish between different data transformations, as a critical component of its self-supervised learning approach. This is done by the

loss function DCL shown below on (2.9).

$$L_{dcl} = E_{x \sim D}[-\sum_{k=1}^{K} \log \frac{h(x_k, x)}{h(x_k, x) + \sum_{l \neq k} h(x_k, x_l)}] \quad (2.9)$$

(2.9) the loss function is computed by the similarity score function, $h(x_k, x) := \exp(\mathrm{sim}(g(T_k(x^u)), g(T_l(x^u)))/\tau)$, where τ is the temperature parameter and $\mathrm{sim}(z, z') := \frac{z \top z'}{\|z\| \cdot \|z'\|}$ as the cosine similarity function.

We adapt the self-supervised learning technique from NeuTraL AD, mainly loss function of DCL. We will further discuss on how we have utilized L_{dcl} in our method in section 3(Ⅲ).

SimCLR (Chen et al., 2020) and *MoCo* (He et al., 2020) are two popular techniques in the field of contrastive learning, both designed to learn meaningful embedding vectors. They both aim to maximize the similarity between the positive pair and minimize that of the negative.

SimCLR (Chen et al., 2020) employs two identical encoders to independently process two augmented views of the same input data. The data augmentation creates two different views of each data sample and applies the encoder to both views. These representations made from the encoder are used to compute similarity scores between the data points. SimCLR uses a temperature parameter to control the sharpness of the similarity distribution.

$$l_{i,j} = -\log \frac{\exp(\mathrm{sim}(z_i, z_j))/\tau)}{\sum_{k=1}^{2N} I_{[k \neq i]} \exp(\mathrm{sim}(z_i, z_j))/\tau)}, \qquad (2.10)$$

where $\mathrm{sim}(u,v) = \frac{u \top v}{\| u \| \cdot \| v \|}$ is the similarity score function. SimCLR learns the representations by maximizing the agreement between different augmentations of the same data with the contrastive loss.

MoCo (He et al., 2020) has a dual-encoder architecture with the query encoder and the key encoder. The query encoder is updated more frequently through standard backpropagation during each training step, reflecting the most recent changes in the training process. Whereas the key encoder's update is updated less frequently, as its parameters are updated as a moving average of the query encoder's parameters. By using the contrastive loss function, InfoNCE in (2.11) helps the model stabilize and improve the training process. q is the encoded query, k_+ is the dictionary keys.

$$L_q = -\log \frac{\exp(q \cdot k_+/\tau)}{\sum_{i=0}^{K} \exp(q \cdot k_+/\tau)}, \qquad (2.11)$$

where τ is the temperature parameter, a sum over one positive and K negatives.

SubTab (Ucar et al., 2021) introduces a transformative approach to

self-supervised learning in tabular data. Unlike methods that rely on corrupting data by adding noise, SubTab operates the learning process by dividing input features into multiple subsets. These subsets can have overlapping parts. Each one is fed into the same encoder which share hyperparameters, generating latent representations. A shared decoder then reconstructs the data, either the subset itself or the entire dataset from these subset features. The network can optionally include a contrastive loss using projections from all subsets.

Scarf (Bahri et al., 2021) introduces a technique that forms views by corrupting a random subset of features in tabular data. This method employs a self-supervised, contrastive pre-training procedure.
First is the pre-training step. In this step, views are generated from a given unlabeled input. This corruption is done by replacing the selected features with random values drawn from their empirical distributions. This method computes embeddings for these corrupted views and then calculates contrastive loss using pairwise similarities.

The second is the fine-tuning step, in which the pre-trained model is applied to a supervised task. The model learned representations from the unlabeled data from the pre-training step. In this step, the model is trained on a smaller portion of the labeled dataset to perform a classification task. This step allows the model to adjust its learned dataset better for accuracy during its task. Cross-entropy loss is calculated in this step. The networks used in the pre-training step are f (encoder), g (pre-training head). However, in the fine-tuning step, the same encoder and the h (classifier)

are used, discarding the pre-training head.

3. Self- and Semi-supervised learning

VIME (Yoon et al., 2020) proposes a self- and semi-supervised learning method in the tabular data domain. This framework comprises three networks: an encoder(e), a decoder, and a mask estimator, as well as two types of losses: reconstruction loss and cross-entropy loss.

This method employs two pretext tasks. The first one is the mask vector estimator s_m, where it predicts which features have been masked. The second pretext task is the feature vector estimator s_r which predicts the values of the features that are corrupted.

$$l_m(m,\hat{m}) = -\frac{1}{d}[\sum_{j=1}^{d} m_j \log[(s_m \circ e)_j(\tilde{x})] + (1-m_j)\log(s_m \circ e)_j(\tilde{x})], \qquad (2.12)$$

where $\hat{m} = (s_m \circ e)(\tilde{x})$ is the recovered mask and $\hat{x} = (s_r \circ e)(\tilde{x})$ is the recovered feature.

The loss function l_m is to learn how the mask which transforms the data is applied by the sum of the binary cross-entropy losses for each dimension of the mask vector.

$$l_r(x,\hat{x}) = \frac{1}{d}[\sum_{j=1}^{d}(x_j - (s_r \circ e)_j(\tilde{x}))^2], \qquad (2.13)$$

where $\hat{x}=(s_r \circ e)(\tilde{x})$. l_r is the reconstruction loss where it prompts the model to look at the corrupted data $\tilde{x}$ to x the same. The model is able to learn about the data (x), as it learns x and $\tilde{x}$ to be equal to each other.

After the self-supervised learning (pretraining step), the model then re-learns the mask estimator and encoder using cross-entropy loss, which assesses the existing masks and finds correlations between features. The method also employs column-wise swap noise to corrupt the data, a strategy well-suited for tabular data. For labeled samples, the encoder generates feature representations (z). The predictor then determines the labels of these samples, with supervised loss (shown in (2.14)) assessing the accuracy of these predictions against existing labels. $f_e = f \circ e$ and $\hat{y}=(f \circ e)(x_u)$ which generates corrupted samples ($\tilde{x}$).

$$L_s := \mathop{\mathbb{E}}_{(x,y) \sim P_{XY}} \left[l_s \left[y, f_e(x) \right] \right], \qquad (2.14)$$

where l_s is a standard supervised loss function such as cross-entropy loss which learns to equalize the ground truth label y with the model's prediction.

The method's innovation lies in its use of unlabeled samples. Applying the randomly generated mask creates K labels from K unlabeled data samples m. Since these labels come from the same sample, their outputs are identical. The model calculates the consistency loss to facilitate the learning of the predictor. In other words L_u in (2.15) learns to equalize

the result from the corrupted samples and the result from the original samples.

$$L_u := \mathop{\mathbb{E}}_{x \sim P_X, m \sim p_m, \tilde{x} \sim g_m(x,m)} \left[(f_e(\tilde{x}) - f_e(x))^2 \right], \qquad (2.15)$$

where the consistency loss means unsupervised learning. By combining the two losses the final loss is conducted shown in (2.16).

$$L_{final} = L_s + \beta L_u, \qquad (2.16)$$

where β is the hyperparameter controlling the ratio between supervised and unsupervised losses.

Ⅲ. Proposed Method

Our goal is to develop a robust semi-supervised learning framework tailored for the tabular data domain. Our objective is to enhance classification performance by using the data augmentation techniques that are specialized in classification.

1. Notations & definitions

Let $L=(x_1,y_1),\ldots,(x_{n_l},y_{n_l})$ be labeled dataset and $U=x_1^u,\ldots,x_{n_u}^u$ be unlabeled dataset. Here $x_i, x_j^u \in R^P$, $y_i \in 1,\ldots,\mathrm{C}$, and C is the total number of classes. We denote one-hot encoded version of y_i as $y_i^{oh} \in 0,1^{\mathrm{C}}$, and we will use y_i and y_i^{oh} interchangeably if there is no confusion.

The architecture we employ in our study includes augmentation (or transformation) functions $T_1,\ldots,T_K$, an encoder g, and a classifier f.

$T_1,\ldots,T_K : R^p \rightarrow R^p$ describe the augmentation functions, where each function outputs a transformed (or augmented) input while maintaining the feature related to its label. An encoder $g : R^p \rightarrow R^d$ maps into d-dimensional embedding vector (we set $d < p$). The encoder reduces the dimensionality of the input data for better feature learning. The function $f(x)=(f_1(x),\ldots,f_C(x)) : R^p \rightarrow S^C$ is a classifier where S^C is the C-dimensional simplex. In other words, $f_c(x)$ refers to the conditional

probability of x belonging to the label c.

2. Proposed objective function

For this section, we will explain the objective (loss) function that is used. Equation (3.1) below shows the total loss function of our proposed method. The loss function is composed of three components, L_x, L_u and L_{dcl}.

$$L = L_x + \lambda_u . L_u + \lambda_{dcl} . L_{dcl}, \qquad (3.1)$$

where λ_u, λ_{dcl} is the hyper-parameters which respectively control the effect of L_u and L_{dcl} in the total loss function. In the experiment analysis, we choose the optimal hyper-parameters based on validation accuracy. L_x, L_u and L_{dcl}, whose detailed explanations will be as follows.

1) L_x

Let $(x,y) \in L$ and $x^u \in U$ be given labeled and unlabeled samples. As we do not know the true label of x^u, we use its pseudo label $p(x^u) = (p_1(x^u), \ldots, p_C(x^u))$ with the current models, which is formulated as follows:

$$p_c(x^u) \propto \left(\frac{1}{K} \sum_{k=1}^{K} f_c(T_k(x^u)) \right)^{1/\tau_1},$$

where τ_1 represents the temperature hyper-parameter. Then the averaged prediction is sharpened with the hyper-parameter τ_1. We note that smaller the value of τ_1 is, the sharper the distribution of $p(x^u)$ would be, i.e., smaller entropy of $p(x^u)$.

The formulation of L_x is given as:

$$L_x := \mathop{\mathbb{E}}_{(x,y)\sim L}[CE[y, f(x)]] + \mathop{\mathbb{E}}_{(x,y)\sim L}\mathop{\mathbb{E}}_{x^u\sim U}\mathop{\mathbb{E}}_{k,l\sim Unif1,\ldots,K}\mathop{\mathbb{E}}_{\lambda\sim \mathrm{Unif}(0,0.5)}\left[\mathrm{CE}\left[(\mathrm{MU}_\lambda(\mathrm{y}, \mathrm{p}(\mathrm{x}^u)), \mathrm{f}(\mathrm{MU}_\lambda(\mathrm{T}_k(\mathrm{x}), \mathrm{T}_l(\mathrm{x}^u)))\right)\right], \quad (3.2)$$

where $M_\lambda(a,b) = (1-\lambda)a + \lambda b$, CE refers to standard cross-entropy loss.

The objective function L_x is built upon the Mixup cross-entropy loss function of the labeled sample (x,y) and the tuple of transformed unlabeled input and its pseudo label $(T_k(x^u), p(T_k(x^u))$. Minimizing L_x with respect to f and T_ks encourages the classifier to provide accurate prediction results and the transformation function to generate augmented data which preserves the label's information.

2) L_u

The objective function L_u is formularized as:

$$L_u := \mathop{\mathbb{E}}_{x_1^u, x_2^u\sim U}\mathop{\mathbb{E}}_{k,l\sim Unif1,\ldots,K}\mathop{\mathbb{E}}_{\lambda\sim \mathrm{Unif}(0,0.5)} \| (\mathrm{MU}_\lambda(\mathrm{p}(\mathrm{x}_1^u), \mathrm{p}(\mathrm{x}_2^u)) - \mathrm{f}(\mathrm{MU}_\lambda(\mathrm{T}_l(\mathrm{x}_1^u), \mathrm{T}_k(\mathrm{x}_2^u))) \|_2^2, \quad (3.3)$$

where $MU_\lambda(a,b) = (1-\lambda)a + \lambda b$.

The loss function (3.3) is the squared Mixup L_2-distance of two

augmented unlabeled samples and their corresponding pseudo labels. The use of L_2-distance instead of the standard cross-entropy is inspired by the MixMatch paper. By minimizing (3.3) with respect to f and T_ks, we can obtain consistent and smooth f over unlabeled data support and let the transformation functions maintain label information of given input data as well.

3) L_{dcl}

We adopt the same loss function firstly developed in *Neutral AD* called the deterministic contrastive loss (DCL), which is given as:

$$L_{dcl} := \mathop{\mathbb{E}}_{X^u \sim U}\left[-\sum_{k=1}^{K}\log\left(\frac{h(g \circ T_k(x^u), g(x^u))}{h(g \circ T_k(x^u), g(x^u)) + \sum_{k \neq l} h(g \circ T_k(x^u), g \circ T_l(x^u)}\right)\right], \quad (3.4)$$

where $h(a,b) = \exp\left(\frac{a \top b}{\tau_2 \times \| a \|_2 \cdot \| b \|_2}\right)$ is the cosine similarity between two vectors a and b and τ_2 is the hyper-parameter. Minimizing (3.4) with respect to T_ks and g leads to the data generated by transformation functions to learn the variety of data created while not significantly losing the properties of the original data. Minimizing this and the two loss functions mentioned above makes various augmentations possible without losing label information.

The main goal of the loss function all combined, shown in equation (3.1),

is to augment data while retaining the original dataset's properties. The loss function L that combines losses from labeled data L_x, unlabeled data L_u, and deterministic contrastive loss L_{dcl}. The classier f, encoder g and transformations T_ks are trained to ensure that f accurately learns from the labeled and unlabeled data without overfitting. T_k is designed to allow for as diverse augmentation as possible without compromising data features or labels. By minimizing these loss components, the approach aims to produce a classifier capable of handling various data augmentations without losing label information.

Ⅳ. Experiments

1. Datasets and preprocessing

The experimental data used in this study are Adult, Covertype, and Mnist data.

Adult is data for binary classification of whether an individual's annual income exceeds $50,000 based on demographic information. There are 48,842 observations, and 9 of the 15 variables are categorical variables.

Covertype, which is data that determines the type of forest in the area based on geographic and geological information, is used in multi-classification. There are 52,292 observations, and 45 of the 55 variables are categorical. All 45 categorical variables except 'Cover_Type,' which represents the forest type, are binary variables.

MNIST is extensively utilized for benchmarking classification algorithms. It consists of a collection of handwritten digits ranging from 0 to 9. The dataset comprises 70,000 images, each representing a 28×28 pixel grayscale digit representation. Of these, 60,000 images are designated for training and 10,000 for testing. Each image is labeled with the corresponding digit it represents, making it a multi-class classification problem. MNIST dataset primarily used for image classification is vectorized to a tabular data format. This version of MNIST includes 784-dimensional vectors, each corresponding to the flattened grayscale pixel values of 28×28 images of handwritten digits. Adult data consists of continuous and categorical features. For the

categorical features, one-hot encoding is used. Continuous variables are set to have the numbers range between -1 and 1. Categorical variables are treated as the continuous, as the 0 are treated as the value of -1, and 1 being 1. 10% of the training data is randomly picked as the labeled data, and the rest is used as unlabeled.

In the experimental process, 65% of Covertype data were divided into training data, 15% as the validation data and 20% as test data. MNIST and Adult data did not need this procedure since the split form of training and test data was already done. For MNIST and Adult dataset, we split the existing training data into 8:2, to make the validation data and leave the test data as it is. This results having used the same proportion for all three datasets.

2. Architecture

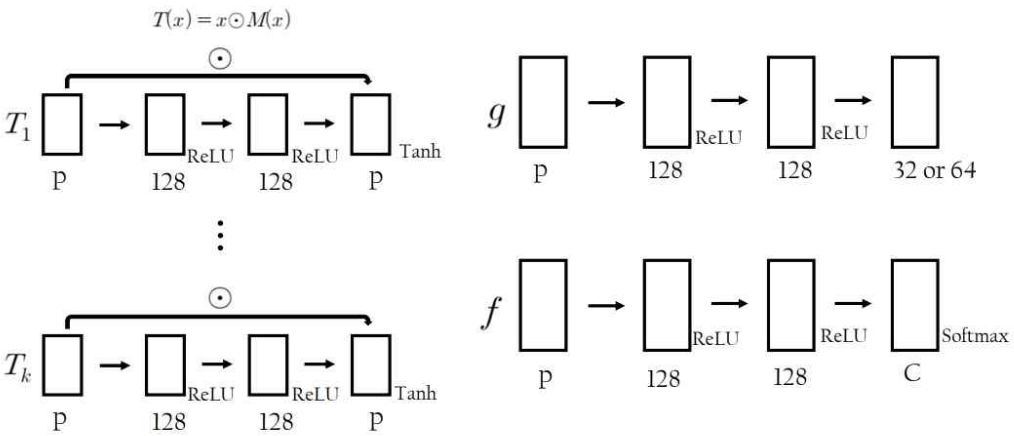

Figure 1. Architecture of our model

[Figure 1] shows the architecture of our model, consisting of transformation function, encoder and a classifier, T_ks, g, and f respectively. Every architecture use a two-layer neural network with a rectified linear unit (ReLU) activation function, set to 128 dimensions. The transformation function T_ks are made into k sets, with p dimension as the input value, through 2 hidden layers, and the output as p, the same dimension as the input. The encoder function g, takes in p dimension as the input, and through the 2 hidden layers, outputs 32, 64 (for MNIST) dimension, the reduced dimension of the input. For the classifier function f, uses the p dimension as input and outputs the C dimension as the probability of belonging to each class.

3. Implementation details

The hyperparameters λ_u and λ_{dcl} are the tuning parameters of this method. We have split the training dataset into 8:2, and by training only 80% of the data, the hyperparameters are set within the highest validation accuracy.

Table 1 shows the selected tuning parameters

	λ_u	λ_{dcl}
Adult	125	1.5
Covertype	150	0.5
MNIST	150	0.5

Table 1: Tuning parameters

Our model is trained over 150 epochs, and a mini-batch set to 100, for (Covertype dataset 500). The proposed method was learned using the Adam optimizer, the learning rate set at 0.001. and the test data was later used to measure supervised learning performance, a usability evaluation index. In the experiment, we compared the performance with AE, Manifold Mixup, Vime and Contrastive Mixup.

4. Evaluation metric

The primary metric for evaluating model performance was test accuracy, the percentage of correct predictions made by the model on the validation dataset. The test accuracy is a direct measure of a model's generalization capabilities to new, unseen data.

5. Results

		Dataset		
Type	Method	Adult	Covertype	MNIST
Supervised learning	Logistic Regression	82.41	70.54	90.12
	XGB	-	-	97.41
	MLP	83.19	75.95	93.69
Semi-supervised learning	AE	84.18	79.97	94.72
	Manifold Mixup	84.68	78.79	94.92
	VIME	84.54	79.02	95.71
	Constrastive Mixup	85.42	80.41	97.58
	Ours	**83.83**	**85.46**	**97.98**

Table 2: Comparison on public tabular datasets

As a result of the performance comparison, our proposed method

outperforms in having the highest learning performance in Covertype and MNIST datasets than the existing methods, and slightly lower performance for the Adult datasets. We note that the results shown in Table 2 are referenced from the Contrastive Mixup. Contrastive Mixup is known to be one of the state-of-the-art methods, and for our method to have outperformed in the MNIST, Covertype datasets is something to emphasize. As for the Adult dataset, we are currently looking for the reason. However even for the Adult dataset where, the test accuracy is slightly lower, our method outperforms the learning performance when compared to the supervised learning. These results demonstrate that our method effectively utilizes the unlabeled data to enhance learning, leading to improved performance on the tabular dataset.

Dataset	Test Accuracy	
	$\lambda_u = 0$	$\lambda_{dcl} = 0$
Adult	83.42	83.89
Covertype	83.54	84.15
MNIST	97053	96.96

Figure 3. Test accuracy when $\lambda_u = 0$, $\lambda_{dcl} = 0$

Figure 3 shows the ablation study on what the test accuracy would be when the hyperparameters, λ_u, λ_{dcl} would each have a value of zero. The results demonstrate that using all three components, L_x, L_u and L_{dcl} with the optimal tuning parameters λ_u, λ_{dcl} brings out the highest learning performance. Also, the components each have their own role to perform for an optimal result.

Ⅴ. Conclusion

We present a novel semi-supervised learning method tailored explicitly for tabular data incorporating a learnable (data adaptive) data augmentation technique. Semi-supervised learning can be advantageous when labeled data are limited and expensive to obtain, utilizing unlabeled data to enhance model performance.

Our proposed method is inspired by MixMatch, a semi-supervised learning method for images, and Neutral AD, a self-supervised learning method for anomaly detection. These methods are adapted to handle tabular data through a novel loss function. The loss function is composed of three parts: L_x for labeled data, L_u for unlabeled data, and L_{dcl} for deterministic contrastive learning. This loss function guides the learning of transformations that generate diverse and informative data augmentations without losing the original data's characteristics.

Experimentally, our method is validated on three datasets: Adult, Covertype, and MNIST, demonstrating state-of-the-art performance, particularly on Covertype and MNIST. An architecture comprising a two-layer neural network with ReLU activation is used, and hyperparameters are tuned for optimal performance. The method outperforms several baselines regarding test accuracy, and an ablation study confirms the significance of the proposed hyperparameters.

We developed a semi-supervised learning method on tabular data using a learnable augmentation technique. We plan on investigating more diverse

datasets to validate the method's superiority more clearly. We intend to provide theoretical justification for the data augmentation method. Finally, additional experiments will be performed to show the visualization of the augmentations.

References

Bahri, D., Jiang, H., Tay, Y., & Metzler, D. (2021). Scarf: Self-supervised contrastive learning using random feature corruption. arXiv preprint arXiv:2106.15147.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems, 32.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020, November). A simple framework for contrastive learning of visual representations. In International conference on machine learning (pp. 1597-1607). PMLR.

Darabi, S., Fazeli, S., Pazoki, A., Sankararaman, S., & Sarrafzadeh, M. (2021). Contrastive mixup: Self-and semi-supervised learning for tabular domain. arXiv preprint arXiv:2108.12296.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9729-9738).

Miyato, T., Maeda, S. I., Koyama, M., & Ishii, S. (2018). Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE transactions on pattern analysis and

machine intelligence, 41(8), 1979-1993.
Qiu, C., Pfrommer, T., Kloft, M., Mandt, S., & Rudolph, M. (2021, July). Neural transformation learning for deep anomaly detection beyond images. In International Conference on Machine Learning (pp. 8703-8714). PMLR.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., ... & Li, C. L. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Advances in neural information processing systems, 33, 596-608.

Ucar, T., Hajiramezanali, E., & Edwards, L. (2021). Subtab: Subsetting features of tabular data for self-supervised representation learning. Advances in Neural Information Processing Systems, 34, 18853-18865.

Verma, V., Kawaguchi, K., Lamb, A., Kannala, J., Solin, A., Bengio, Y., & Lopez-Paz, D. (2022). Interpolation consistency training for semi-supervised learning. Neural Networks, 145, 90-106.

Qiu, C., Pfrommer, T., Kloft, M., Mandt, S., & Rudolph, M. (2021, July). Neural transformation learning for deep anomaly detection beyond images. In International Conference on Machine Learning (pp. 8703-8714). PMLR.

Yoon, J., Zhang, Y., Jordon, J., & van der Schaar, M. (2020). Vime: Extending the success of self-and semi-supervised learning to tabular domain. Advances in Neural Information Processing Systems, 33, 11033-11043.

국문초록

데이터 맞춤형 증강기법을 이용한 심층 준지도학습법에 대한 연구

박세리
성신여자대학교
일반대학원
통계학과

본 연구는 표(tabular) 형태의 데이터를 위해 설계된 새로운 준지도 학습 접근 방식을 소개하였다. 라벨이 있는 데이터만을 사용하기에 제한적이고 비용이 많이 든다는 문제를 해결하기 위해 저희의 방법은 라벨링 되지 않은 데이터를 효과적으로 활용하여 모델 성능을 향상하게 시킨다.

이 접근 방식은 이미지 데이터에서 준지도 학습으로 알려진 MixMatch와 이상 감지를 위한 자체 지도 학습 (self-supervised learning) 방법인 Neutral AD의 두 가지 방법에서 영감을 받았다. 손실함수는 라벨이 있는 데이터, 라벨이 없는 데이터 및 DCL(Deterministic Contrastive Loss)의 세 가지 부분으로 구성되어 각 부분이 표 형태의 데이터에 맞게 조정된다. 이 손실함수는 원본 데이터의 특성을 유지하면서 다양하고 라벨의 정보를 지닌 데이터 증강을 생성하는 변환을 하게 해준다.

이 방법론을 검증하기 위해 세 가지 표 형식의 데이터 세트에 대한 실험을 수행했으며, 여기서 저희 방법은 특히 두 가지 데이터 세트에서 주목할 만한 최첨단 성능을 입증하였다. 결과는 여러 기준선에 비해 우수한 테스트 정확도를 보여줄 뿐만 아니라 절제 연구를 통해 제안된 하이퍼 파라미터의 중요성을 강조합니다.

결론적으로, 우리의 연구는 새로운 학습 가능한 증강기법을 사용하여 표 데이터에 대한 준지도 학습 방법론을 개발했다.

감사의 글(Acknowledgements)

대학원에 처음 입학했을 때가 엊그제 같은데 벌써 졸업의 시간이 다가왔습니다. 2년간의 대학원에서는 학업뿐만 아니라, 더불어 살아가는 방법도 배우며, 좋은 인연들도 만나게 되었던 것 같습니다. 이 글을 통해 제가 대학원에서 많은 것을 경험할 수 있게 도와주신 분들께 감사 인사드리고자 합니다.

부족함이 많은 저를 지도 학생으로 삼아주시고 차근차근 연구자로서 자질을 길러주시며, 저의 모든 대학원 생활을 살피시고 이끌어주신 김동하 지도교수님께 진심으로 감사를 드립니다. 2021년 겨울부터 많은 도움과 관심을 주셔서 전반적인 대학원 생활에 잘 적응할 수 있게끔 도와주셨던 박만식 교수님, 언제나 따듯한 미소로 격려와 조언을 해주셨던 이성건 교수님, 학교 선배님으로서 걸어오신 길을 알려주시며 격려를 해주신 박희원 교수님, 학업뿐만 아니라 앞으로의 진로에 큰 관심을 주시고 아낌없는 조언을 해주신 박관영 교수님, 전반에 중심을 잘 잡아주시면서 친절하게 도움을 주신 박성오 교수님, 잘할 수 있을 것이라 응원해주시고 격려해주신 정호현 교수님께 깊은 감사를 드립니다.

대학원 생활에 정말 많은 의지가 되고 덕분에 많은 것을 배우게 해준 선배 지우, 외로운 길을 함께 걸으면서 옆에서 든든한 동반자 되어준 윤아에게 진심으로 감사를 드립니다. 항상 버팀목이 되어 많은 공감을 나눈 서영과 지혜 그리고 602호에서 동고동락한 서연, 혜민 그리고 항상 즐거웠던 때가 그리운 9층에서의 나경, 윤진, 수지에게도 고마움을 전합니다. 앞으로 대학원 생활을 시작하게 된 진주, 세영, 주이, 민서도 응원합니다.

더불어 언제나 저를 fully support 하신다며 항상 믿고 응원해주신 사랑하는 아빠, 엄마, 오빠 그리고 대흠이까지 너무 사랑하고 감사드립니다.

대학원에서 배운 것을 토대로 앞으로 더욱 성장하는 박세리가 되겠습니다.