Adaptive Deep Learning for Entity Resolution by Risk Analysis

Qun Chen, Zhaoqiang Chen, Youcef Nafa, Tianyi Duan, Wei Pan*, Lijun Zhang, Zhanhuai Li

[a]*School of Computer Science, Northwestern Polytechnical University, Xi'an, China*
[b]*Key Laboratory of Big Data Storage and Management, Northwestern Polytechnical University, Ministry of Industry and Information Technology Xi'an, China*

Abstract

The state-of-the-art performance on entity resolution (ER) has been achieved by deep learning. However, deep models usually need to be trained on large quantities of accurately labeled training data, and can not be easily tuned towards a target workload. However, in real scenarios, there may not be sufficient training data; even if they are abundant, their distribution is almost certainly different from target data to some extent.

To alleviate such limitation, this paper proposes a novel risk-based adaptive training approach for ER that can tune a deep model towards its target workload by the workload's particular characteristics. Built on the recent advances on risk analysis for ER, the proposed approach first trains a deep model on labeled training data, and then fine-tunes it on unlabeled target data by minimizing its misprediction risk. Our theoretical analysis shows that risk-based adaptive training can correct the label status of a mispredicted instance with a fairly good chance. Finally, we empirically validate its efficacy on real benchmark data by a comparative study. Our extensive experiments show that it can considerably improve the performance of deep models. Furthermore, in the scenario of distribution misalignment, it can similarly outperform the state-of-the-art alternatives of transfer learning by considerable margins. Using ER as a test case, we demonstrate that risk-based adaptive training is a promising approach potentially applicable to various challenging classification tasks.

Keywords: Entity Resolution, Risk Analysis, Deep Learning, Domain Adaptation

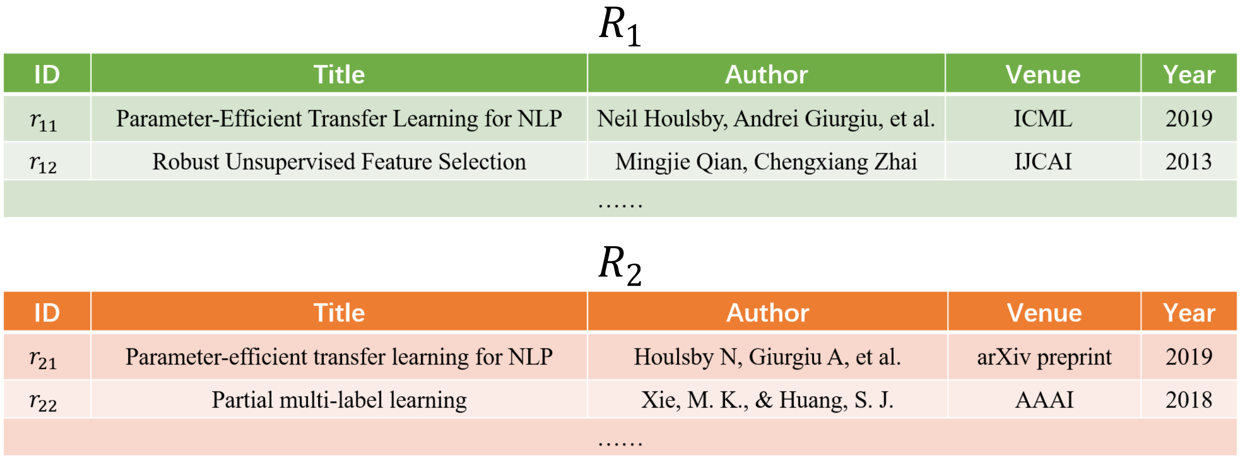
1. Introduction

As a very important problem for data integration [1], ER aims to identify equivalent records that refer to the same real-world entity. Considering the running example shown in Figure 1, ER needs to match the paper records between two tables, R_1 and R_2. A pair of $< r_{1i}, r_{2j} >$, in which r_{1i} and r_{2j} denote a record in R_1 and R_2 respectively, is called an *equivalent* pair if and only if r_{1i} and r_{2j} refer to the same paper; otherwise, it is called an *inequivalent* pair. In this example, r_{11} and r_{21} are *equivalent* while r_{11} and r_{22} are *inequivalent*. ER can be considered as a binary classification problem tasked with labeling record pairs as *matching* or *unmatching*.

The state-of-the-art performance on ER has been achieved by deep learning [2, 3, 4, 5, 6]. However, the efficacy of these deep models depends on large quantities of accurately labeled training data, which may not be readily available in real scenarios. Furthermore, in the typical setting of deep learning, a classifier tunes its model parameters on labeled training data to ensure that its predictions on the training instances are consistent with their ground-truth labels. The resulting classifier is then supposed to be directly applied on a target workload. It can be observed that the typical process of model training does not involve unlabeled data in a target workload, even though to alleviate the over-fitting problem, labeled validation data are usually provided as a proxy workload and leveraged for hyperparameter tuning. Theoretically, the efficacy of this approach is based on the assumption that training and target data are **i**ndependently

[*] Corresponding author

Email addresses: `chenbenben@nwpu.edu.cn` (Qun Chen), `chenzhaoqiang@mail.nwpu.edu.cn` (Zhaoqiang Chen), `youcef.nafa@mail.nwpu.edu.cn` (Youcef Nafa), `tianyiduan@mail.nwpu.edu.cn` (Tianyi Duan), `panwei1002@nwpu.edu.cn` (Wei Pan*), `zhanglijun@nwpu.edu.cn` (Lijun Zhang), `lizhh@nwpu.edu.cn` (Zhanhuai Li)



R_1

ID	Title	Author	Venue	Year
r_{11}	Parameter-Efficient Transfer Learning for NLP	Neil Houlsby, Andrei Giurgiu, et al.	ICML	2019
r_{12}	Robust Unsupervised Feature Selection	Mingjie Qian, Chengxiang Zhai	IJCAI	2013
……				

R_2

ID	Title	Author	Venue	Year
r_{21}	Parameter-efficient transfer learning for NLP	Houlsby N, Giurgiu A, et al.	arXiv preprint	2019
r_{22}	Partial multi-label learning	Xie, M. K., & Huang, S. J.	AAAI	2018
……				

Figure 1: An ER running example.

and **i**dentically **d**istributed (the **i.i.d** assumption). Unfortunately, in real scenarios, even when training and target data come from the same domain, the **i.i.d** assumption may not hold due to: 1) training data are not sufficient to fully represent the statistical characteristics of a target workload; 2) even though training data are abundant, its inherent distribution may be to some extent different from a target workload. Therefore, it is common in real scenarios that a well trained deep model does not perform well on a target workload.

Many adaptation approaches have been proposed to alleviate distribution misalignment, most notably among them *transfer learning* [7, 8, 9] and *adaptive representation learning* [10, 11, 12, 13, 14]. Transfer learning aims to adapt a model learned on training data in a source domain to a target domain. Similarly, adaptive representation learning, which was originally proposed for image classification, mainly studies how to learn domain-invariant features shared among diversified domains. Unfortunately, distribution misalignment remains very challenging. The main reason is that the existing approaches focus on how to extract and leverage the common knowledge shared between a source task and a target task; however, they can not effectively tune a classifier towards its target task by the task's particular characteristics.

It has been well recognized that in real scenarios, with or without adaptation, a well-trained classifier may not be accurate in its predictions. Even worse, it may provide high-confidence predictions which turn out to be wrong [15]. Such prediction uncertainty has emerged as a critical concern to AI safety [16]. Therefore, various approaches [17, 18, 19, 20, 21] have been proposed for the task of risk analysis, which aims to estimate the misprediction risk of a deep classifier when applied to a certain workload. Since risk analysis can measure the misprediction risk of a classifier on unlabeled data, it provides classifier training with a viable way to adapt towards a particular workload. Hence, we propose a risk-based approach to enable adaptive deep learning for ER in this paper. Since the recently proposed LearnRisk [21] is more interpretable and more accurate in identifying mispredictions than previous alternatives, we build the solution of adaptive deep training upon LearnRisk in this paper.

We have sketched the proposed approach in Figure 2. It consists of two phases, the phase of *traditional training* followed by the phase of *risk-based training*. In the first phase, a deep model is trained on labeled training data in the traditional way; in the second phase, it is further tuned on unlabeled target data to minimize its misprediction risk. The main contributions of this paper can be summarized as follows:

- We propose a novel risk-based approach to enable adaptive deep learning.
- We present a solution of adaptive deep learning for ER based on the proposed approach.
- We theoretically analyze the performance of the proposed solution for ER. Our analysis shows that risk-based adaptive training can correct the label status of a mispredicted instance with a fairly good chance.

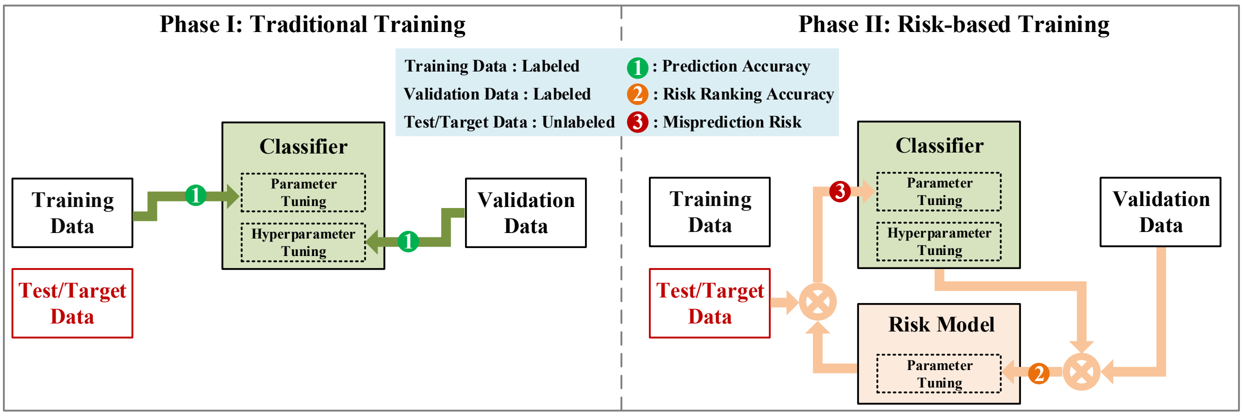


Figure 2: Risk-based Adaptive Training.

- We empirically validate the efficacy of the proposed solution on real benchmark data by a comparative study.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 presents the preliminaries. Section 4 presents the adaptive training approach and its theoretical results. Section 5 presents our empirical evaluation results. Finally, Section 6 concludes this paper with some thoughts on future work.

2. Related Work

We review related work from three mutually orthogonal perspectives: entity resolution, model training and ensemble learning. For more detailed review on deep learning for ER, please refer to the survey [22].

Entity Resolution. Playing a key role in data integration, ER has been extensively studied in the literature [23, 24, 25]. ER can be automatically performed based on rules [26, 27, 28], probabilistic theory [29, 30] and machine learning models [1, 31, 32, 33]. The state-of-the-art solutions for ER have been built upon various DNN models [2, 3, 4, 5, 6, 22, 34, 35]. Specifically, the first deep learning architecture template for ER was proposed in [3]. In the following work [6], they presented an improved solution based on pre-trained Transformer-based language models (e.g., BERT). The authors of [35] proposed a solution that combines the Transformer attention and hierarchical graph attention network to exploit various relationships among ER decisions. The authors of [34] proposed a siamese structure to accelerate the process of BERT-based ER. To reduce the labeling effort required by deep models, the authors of [36] applied adversarial active learning on deep ER. In addition, various domain adaption techniques have been proposed for deep ER to reduce required training data on new workloads [37, 38, 39]. In particular, the authors of [39] systematically explored various existing domain adaptation methods and found that adversarial-based method performs the best on the ER task. It is noteworthy that this paper does not attempt to propose a new deep model for ER. It instead focuses on how to tune a deep model towards its target workload via risk analysis. Therefore, the existing work on deep learning for ER is orthogonal to ours. In principle, our proposed approach can work with any deep model for ER.

ER remains very challenging in real scenarios due to prevalence of dirty data. Therefore, there is a need for *risk analysis*, alternatively called *trust scoring* and *confidence ranking* in the literature. The proposed solutions range from those simply based on the model's output probabilities to more sophisticated and interpretable ones [17, 19, 21, 40]. Most recently, we proposed an interpretable and learnable framework for ER, LearnRisk[21]. In our following work [41], we also proposed to actively select training data for ER models based on the results of risk analysis. In this paper, we investigate how to leverage the results of risk analysis to fine-tune a deep model toward its target workload.

Model Training. A common challenge for model training is the *over-fitting*, which refers to the phenomenon that a model well tuned on training data performs unsatisfactorily on target data. The de facto standard approach to alleviate *over-fitting* is by leveraging validation data for hyperparameter tuning and model selection (e.g., cross validation) [42]. Another noteworthy complementary technique is the *regularization* [43, 44, 45, 46], which aims to reduce the number of model parameters to a manageable level. Both hyperparameter tuning and model selection are to a large extent orthogonal to model training considered in this paper.

The classical way to alleviate the insufficiency of labeled training data is by semi-supervised learning [47, 48]. However, semi-supervised learning investigated how to leverage unlabeled training data, which usually have a similar distribution with labeled training data. It is obvious that the techniques for semi-supervised learning can be straightforwardly incorporated into the traditional training phase of our proposed approach. They are therefore orthogonal to our work. Another way to reduce labeling cost is by active learning [49, 50]. While active learning focuses on how to select training data for labeling, we focus on how to adapt a model towards its target workload via risk analysis on unlabeled target data. Therefore, active learning is also orthogonal to our work.

Ensemble Learning. The classical way to alleviate the limitations of a single classifier is by ensemble learning [51, 52]. Ensemble learning first trains multiple classifiers using different training data (e.g., bagging [53]) or different information in the same training data (e.g., boosting [54]), and then combines probably conflicting predictions to arrive at a final decision. While our risk analysis approach, LearnRisk, uses the ensemble of risk features to measure misprediction risk, risk-based adaptive training is fundamentally different from ensemble learning due to: 1) unlike the traditional labeling functions, *LearnRisk* aims to estimate an instance's misclassification risk as predicted by a classifier; 2) more importantly, the ensemble approach trains multiple models and tunes predictions based on training data; in contrast, risk-based adaptive training trains only one model, and tunes the model towards its target workload. It is noteworthy that since ensemble learning trains models based on training data, it is in fact orthogonal to our work. In principle, our proposed approach can also work with an ensemble learning model. However, how to tune an ensemble model via risk measure requires further investigation in future.

3. Preliminaries

In this section, we first define the task of ER and then introduce the risk analysis approach of LearnRisk.

3.1. Task Statement

This paper considers ER as a binary classification problem. A classifier needs to label every unlabeled pair as *matching* or *unmatching*. As usual, we measure the quality of an ER solution by the standard metric of *F1*, which is a combination of *precision* and *recall* as follows

$$F1 = \frac{2 \times precision \times recall}{precision + recall}. \tag{1}$$

Table 1: The Frequently Used Notations

Notation	Description
D	an ER workload
D^s, D^v, D^t	subsets of D, corresponding to training set, validation set and test set
d_i	an instance pair in D
$\mathbf{x}_i$	the feature representation vector of d_i
y_i	the label of d_i
μ_{d_i}	the expectation of equivalence probability of d_i
$\sigma_{d_i}(resp.\sigma^2_{d_i})$	the standard deviation (resp. variance) of equivalence probability of d_i
f_i	a risk feature
w_i	the feature weight of f_i

For presentation simplicity, we summarize the frequently used notations in Table 1. As usual, we suppose that an ER task, D, consists a set of labeled training data, $D^s = \{(\mathbf{x}_i^s, y_i^s)|i\}$, where each $(\mathbf{x}_i^s, y_i^s)$ denotes a training instance with its feature representation $\mathbf{x}_i^s$ and ground-truth label y_i^s, a set of labeled validation data, $D^v = \{(\mathbf{x}_i^v, y_i^v)|i\}$, and a set of unlabeled test data, $D^t = \{(\mathbf{x}_i^t, ?)|i\}$. Note that D^t denotes the target workload, and D^v serves as a proxy workload of D^t. Formally, we define the task of ER as

Definition 1. ***[ER Classification Task].*** *Given an ER workload D consisting of D^s, D^v and D^t, the task aims to learn an optimal classifier, C_*, based on D such that the performance of C_* on D^t as measured by the metric of $F1$, or $F1(C_*, D^t)$, is maximized.*

3.2. Risk Analysis for ER: LearnRisk

The framework of LearnRisk consists of three main steps: *risk feature generation*, *risk model construction* and finally *risk model training*.

3.2.1. Risk feature generation

The step automatically generates risk features in the form of interpretable rules based on one-sided decision trees. The generation algorithm ensures that the resulting rule-set is discriminative, i.e, each rule is highly indicative of one class label over the other; and has a high data coverage, i.e, its validity spans over a considerable subpopulation of the workload. As opposed to the labeling functions used to label pairs as *matching* or *unmatching*, a risk rule focuses exclusively on one single class. Consequently, a risk feature acts as an indicator of the case where a classifier's prediction goes against the knowledge embedded in it. An example of risk rule is:

$$r_i[Year] \neq r_j[Year] \rightarrow inequivalent(r_i, r_j), \tag{2}$$

where r_i denotes a record and $r_i[Year]$ denotes r_i's attribute value at $Year$. With this knowledge, a pair predicted as *matching* but having different publication years is supposed to have high mislabeling risk.

The detailed process of risk feature generation as well as its computational complexity have been described in our previous work [21]. For computational complexity, let n denote the size of training data, m the number of basic metrics, and h the pre-specified depth of decision trees. Then, the total computational complexity of risk feature generation can be represented by $O(h \cdot (2m)^h \cdot n \cdot logn)$. It is worthy to point out that the number of basic metrics (m) is usually limited (e.g., dozens); to ensure interpretability, the maximum depth of decision tree (h) is also usually set to a small value (e.g., 3-4 in our implementation). Therefore, the algorithm for risk feature generation can be executed efficiently in practice. Our previous empirical evaluation has also shown that it scales well with the size of training data [21].

3.2.2. Risk model construction

Once high-quality features have been generated, the latter are readily available for the risk model to make use of, allowing it to be able to judge a classifier's outputs backing up its decisions with human-friendly explanations. To achieve this goal, LearnRisk, drawing inspiration from investment theory, models each pair's equivalence probability distribution (resp. portfolio reward) as the aggregation of the distributions of its compositional features (resp. stocks rewards).

Formally, LearnRisk models the equivalence probability of a pair d_i by a random variable p_i that follows a normal distribution $\mathcal{N}(\mu_i, \sigma_i^2)$, where μ_i and σ_i^2 denote expectation and variance respectively. Given a set of m risk features $f_1, f_2, ..., f_m$, let $w_1, w_2, ..., w_m$ denote their corresponding weights. Suppose that $\mu_F = [\mu_{f_1}, \mu_{f_2}, \ldots, \mu_{f_m}]^T$ and $\sigma_F^2 = [\sigma_{f_1}^2, \sigma_{f_2}^2, \ldots, \sigma_{f_m}^2]^T$ represent their corresponding expectation and variance vectors respectively, such that $\mathcal{N}(\mu_{f_j}, \sigma_{f_j}^2)$ denotes the equivalence probability distribution of the feature f_j. Then, the distribution of d_i can be estimated by:

$$\mu_i = \mathbf{z}_i(\mathbf{w} \circ \mu_F), \tag{3}$$

and

$$\sigma_i^2 = \mathbf{z}_i(\mathbf{w} \circ \mathbf{w} \circ \sigma_F^2), \tag{4}$$

where $\circ$ represents the element-wise product, and $\mathbf{z}_i$ is a one-hot feature vector. Specifically, $\mathbf{z}_i = [z_{i1}, z_{i2}, ..., z_{ij}, ..., z_{im}]$, where $z_{ij} = 1$ if d_i has the jth feature, otherwise $z_{ij} = 0$.

It is noteworthy that besides one-sided decision rules, LearnRisk also incorporates classifier output as one of its risk features. Provided with the equivalence distribution p_i for d_i, LearnRisk measures its risk by the metric of Value-at-Risk (VaR) [55], which can effectively capture fluctuation risk of label status. Provided with a confidence level of θ, the metric of VaR represents the maximum loss after excluding all worse outcomes whose combined probability is at most 1-θ.

3.2.3. Risk model training

Finally, LearnRisk trains a risk model on labeled validation data. It optimizes a learn-to-rank objective by tuning the weights of risk features (w_i) as well as their variances (σ_i^2). As for their expectations (μ_i), they are considered as prior knowledge, and estimated based on labeled training data. Once trained, the risk model can be used to assess the misprediction risk on an unseen workload labeled by a classifier.

4. Risk-based Adaptive Training

In this section, we present the approach of risk-based adaptive training for ER. We take *DeepMatcher* [3], the classical deep model for ER, as an example to illustrate our solution. However, in principle, our proposed approach can similarly work with other deep models. The rest of this section is organized as follows: Subsection 4.1 describes the traditional training approach. Subsection 4.2 presents the proposed approach of risk-based adaptive training. Finally, Subsection 4.3 presents the results of theoretical analysis.

4.1. Traditional Training

Given a workload, D={D^s, D^v, D^t}, let $g(\omega)$ denote a DNN classifier with the parameters of ω. The traditional approach, as shown in the left part of Figure 2, tunes ω towards the training data, D^s, based on a pre-specified loss function. Supposing that there are totally n_s training instances in D^s, DeepMatcher employs the classical cross-entropy loss function as follows:

$$\mathcal{L}_{train}(\omega) = \frac{1}{n_s}\sum_{i=1}^{n_s}\{-y_i^s log(g(\mathbf{x}_i^s;\omega)) - (1-y_i^s) log(1-g(\mathbf{x}_i^s;\omega))\}, \tag{5}$$

where y_i^s denotes the ground-truth label of a training instance, $(\mathbf{x}_i^s, y_i^s)$, and $g(\mathbf{x}_i^s;\omega)$ denotes its label probability as predicted by the classifier. DeepMatcher uses the Adam optimizer to search for the optimal parameters ω_* by gradient descent [56].

4.2. Risk-based Adaptive Training

Risk-based adaptive training, as shown in Figure 2, consists of two phases: the *traditional training* phase and the *risk-based training* phase. In the first phase, it tunes a deep model towards training data in the traditional way. Then, in the following *risk-based training* phase, it iteratively performs: i) using LearnRisk to learn a risk model based on a trained classifier and validation data; ii) fine-tuning the classifier by minimizing its misprediction risk upon the target workload.

Specifically, risk-based training defines the loss function as

$$\mathcal{L}_{test}^{risk}(\omega) = \frac{1}{n_t}\sum_{i=1}^{n_t}\{-[1-VaR^+(d_i)] log(g(\mathbf{x}_i^t;\omega)) - [1-VaR^-(d_i)] log(1-g(\mathbf{x}_i^t;\omega))\}, \tag{6}$$

in which n_t denotes the total number of instances in D^t, $VaR^+(d_i)$ (resp. $VaR^-(d_i)$) denotes the estimated misprediction risk of d_i if it is labeled as *matching* (resp. *unmatching*). Similar to traditional training, the *risk-based training* phase updates the parameters of the deep model by gradient descent as follows

$$\omega_{k+1} = \omega_k - \alpha * \nabla_{\omega_k}\mathcal{L}_{test}^{risk}(\omega). \tag{7}$$

Algorithm 1 Risk-based Adaptive Training

```
**Input:** A task $D$ consisting of $D^s$, $D^v$ and $D^t$, and an ER model, $g(\omega)$;
**Output:** A learned classifier $g(\omega_*)$.
$\omega_0 \leftarrow$Initialize $\omega$ with random values;
**for** $k = 0$ **to** $m - 1$ **do**
    $\omega_{k+1} \leftarrow \omega_k - \alpha * \nabla_{\omega_k}\mathcal{L}_{train}(\omega_k)$;
**end for**
Select the best model, $g(\omega_*)$, based on $D^v$;
$\omega_m \leftarrow \omega_*$;
**for** $k = m$ **to** $m + n - 1$ **do**
    Update the risk model based on $D^v$ and $g(\omega_k)$;
    $\omega_{k+1} \leftarrow \omega_k - \alpha * \nabla_{\omega_k}\mathcal{L}^{risk}_{test}(\omega_k)$;
**end for**
Select the best model, $g(\omega_*)$, based on $D^v$.
**Return** $g(\omega_*)$
```

Note that in each iteration, risk values are estimated based on the classifier predictions of the previous iteration. As a result, they are considered as constant while computing gradient descent .

We have sketched the process of risk-based adaptive training in Algorithm 1. The first phase pre-trains a model based on labeled training data, and selects the best one based on its performance on validation data. Beginning with the pre-trained model, the second phase iteratively fine-tunes its parameters by minimizing the loss of $\mathcal{L}^{risk}_{test}(\omega)$ upon the target workload.

4.3. Theoretical Analysis

Suppose that *LearnRisk* generates totally m risk features, denoted by $\{f_1,\ldots,f_m\}$. Let Z_i be a 0-1 variable indicating whether an instance has the risk feature f_i: $Z_i = 1$ if the instance has f_i, otherwise $Z_i = 0$. Let $\mathbf{Z} = (Z_1, Z_2, ..., Z_m)$ denote a risk feature distribution. We can reasonably expect that LearnRisk is generally effective: if an instance is equivalent (resp. inequivalent), its risk features (excluding its DNN output) can indicate its equivalence (resp. inequivalence) status. As shown in Eq.3, LearnRisk estimates the equivalence probability expectation of an instance by a weighted linear combination of the expectations of its DNN risk feature and rule risk features. Specifically, given an equivalent pair of d_i, (μ_i,σ_i^2), with m rule risk features, we have

$$\mathbb{E}(\frac{\sum_{j=1}^{m} z_j \cdot w_j \cdot \mu_{f_j}}{\sum_{j=1}^{m} z_j \cdot w_j}) > 0.5, \tag{8}$$

in which f_j denotes a rule risk feature, and $\mathbb{E}(*)$ denotes the statistical expectation. Similarly, if d_i is inquivalent, it satisfies

$$\mathbb{E}(\frac{\sum_{j=1}^{m} z_j \cdot w_j \cdot \mu_{f_j}}{\sum_{j=1}^{m} z_j \cdot w_j}) < 0.5. \tag{9}$$

According to Eq. 8 and 9, once a pair is correctly labeled by a classifier, it can be expected that its label would not be flipped by risk-based fine-tuning. Our experiments on real data have confirmed that risk-based fine-tuning rarely flips the labels of true positives and true negatives. Therefore, in the rest of this subsection, we focus on showing that given a mispredicted instance, d_i, risk-based fine-tuning can make its expectation, or the value of μ_i, be consistent with its ground-truth label with a fairly good chance.

For theoretical analysis, since both true positives and false negatives (resp. true negatives and false postives) are equivalent (resp. inequivalent) instances, they are assumed to share the same distribution of risk feature activation. Formally, we state the assumption on risk feature distribution as follows:

Assumption 1. Identicalness of Risk Feature Distributions. *Given an ER workload, the risk feature activation of each equivalent instance d_i^+, denoted by $\mathbf{Z}_i^+$, is supposed to follow the same distribution of $\mathbf{Z}^+$; similarly, the risk feature activation of each inequivalent instance d_i^-, denoted by $\mathbf{Z}_i^-$, is supposed to follow the same distribution of $\mathbf{Z}^-$.*

Based on Assumption 1, we can establish the lower bound of the estimated equivalence probability expectation of a false negative by the following theorem. Due to space limit, the proofs of the lemmas and theorems are provided in the supplemental materials.

Theorem 1. *Given a false negative $\tilde{d}_j^-$, suppose that there are totally n true positives, denoted by d_i^+, ranked after $\tilde{d}_j^-$ by LearnRisk such that each true positive, d_i^+, satisfies*

$$\Delta VaR^- - \Delta C^- > \epsilon, \tag{10}$$

in which $\Delta VaR^- = VaR^-(\tilde{d}_j^-) - VaR^+(d_i^+)$, and $\Delta C^- = \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-})$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$, its expectation of equivalence probability of $\tilde{d}_j^-$, $\mu_{\tilde{d}_j^-}$, estimated by LearnRisk*, satisfies*

$$\mu_{\tilde{d}_j^-} \geq \frac{1}{2} + \frac{\epsilon}{2} - \sqrt{\frac{m+1}{2} ln[\frac{1}{1-(1-\delta^{\frac{1}{n}})^{\frac{1}{2}}}]}, \tag{11}$$

in which μ_* denotes the expectation of equivalence probability and σ_* denotes its standard deviation.

In Theorem 1, m denotes the number of rule risk features, and the value of ΔC^- corresponds to the difference of risk expectation between false negatives being labeled as *matching* and true positives being labeled as *matching*. Note that the total number of rule risk features (m) is usually limited (e.g., dozens or hundreds), while n is usually much larger than m. It can be observed that in Theorem 1, by the exponential effect of n, the 3rd term on the right-hand side tends to become zero as the value of n increases. Therefore, if $\epsilon > 0$ and there are sufficient true positives satisfying the specified condition, risk-based fine-tuning would have a fairly good chance to correctly flip the label of $\tilde{d}_j^-$ from *unmatching* to *matching*. To gain deeper insight into Theorem 1, we analyze the value of $\Delta VaR^- - \Delta C^-$. Since the optimization objective of LearnRisk is to maximize the risk difference between $VaR^-(\tilde{d}_j^-)$ and $VaR^+(d_i^+)$, ΔVaR^- can be expected to large for most true positives. Therefore, we analyze the value of ΔC^-. Based on Assumption 1, we have the following lemma:

Lemma 1.

$$\Delta C^- \leq max\{\mathbb{E}(w_{d_i^+}(\hat{\mu}_{d_i^+} - 2\hat{\sigma}_{d_i^+})) - \mathbb{E}(w_{\tilde{d}_j^-}(\hat{\mu}_{\tilde{d}_j^-} - 2\hat{\sigma}_{\tilde{d}_j^-})), \mathbb{E}(w_{d_i^+}\hat{\mu}_{d_i^+}) - \mathbb{E}(w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-})\}, \tag{12}$$

where the $\hat{\mu}_*$ and $\hat{\sigma}_*$ denote the DNN output probability and its corresponding standard deviation respectively, w_* denotes the learned weight of DNN risk feature.

It is interesting to point out that as shown in Lemma 1, the value of ΔC^- only depends on the distributions of DNN outputs and their weights, but independent of the distributions of rule risk features. It has the simple upper bound of

$$\Delta C^- \leq \mathbb{E}(w_{d_i^+}). \tag{13}$$

Hence, when the learned weight of DNN output becomes smaller, which means that the DNN becomes less accurate, true positives would have a higher chance to satisfy $\Delta VaR^- - \Delta C^- > 0$. In our experiments, it is observed that the expected weight of classifier output is usually between 0.2 and 0.6, or $0.2 \leq \mathbb{E}(w_{d_i^+}) \leq 0.6$. As a result, Theorem 1 shows that a false negative has a fairly good chance to be flipped from *unmatching* to *matching*.

Based on Assumption 1, the theoretical chance of a false positive being flipped from *matching* to *unmatching* can be similarly established. The corresponding theorem and lemma are presented as follows:

Theorem 2. *Given a false positive $\tilde{d}_j^+$, suppose that there are totally n true negatives, denoted by d_i^-, ranked after $\tilde{d}_j^+$ by LearnRisk such that each true negative, d_i^-, satisfies*

$$\Delta VaR^+ - \Delta C^+ > \epsilon, \tag{14}$$

in which $\Delta VaR^+ = VaR^+(\tilde{d}_j^+) - VaR^-(d_i^-)$, and $\Delta C^+ = \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-})$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$, its expectation of equivalence probability of $\tilde{d}_j^+$, $\mu_{\tilde{d}_j^+}$ estimated by LearnRisk*, satisfies*

$$\mu_{\tilde{d}_j^+} \leq \frac{1}{2} - \frac{\epsilon}{2} + \sqrt{\frac{m+1}{2} ln[\frac{1}{1-(1-\delta^{\frac{1}{n}})^{\frac{1}{2}}}]},$$

in which μ_* denotes the mean of equivalence probability and σ_* denotes its standard deviation.

Table 2: Empirical Validation of Theoretical Analysis.

(a) on True Positives and True Negatives

	#	# Flipped
True Positives	1143	12
True Negatives	5987	27

(b) on False Negatives and False Positives

#($\Delta VaR^- > \Delta C^-$	False Negatives		False Positives	
or $\Delta VaR^+ > \Delta C^+$)	#	# Flipped	#	# Flipped
#< 100	1	0	0	0
#≥ 100	188	179	100	91
Total	189	179	100	91

In Theorem 2, the value of ΔC^+ corresponds to the difference of risk expectation between false positives being labeled as *unmatching* and true negatives being labeled as *unmatching*. Similar to the case of Theorem 1, it can be observed that in Theorem 2, by the exponential effect of n, the 3rd term on the right-hand side tends to become zero as the value of n increases. Therefore, if $\epsilon > 0$ and there are sufficient true negatives satisfying the specified condition, risk-based fine-tuning would have a fairly good chance to correctly flip the label of $\tilde{d}_j^+$ from *matching* to *unmatching*. To gain a deeper insight into Theorem 2, we also analyze the value of ΔC^+ by the following lemma:

Lemma 2.

$$\Delta C^+ \leq max\{\mathbb{E}(w_{\tilde{d}_j^+}(\hat{\mu}_{\tilde{d}_j^+} + 2\hat{\sigma}_{\tilde{d}_j^+})) - \mathbb{E}(w_{d_i^-}(\hat{\mu}_{d_i^-} + 2\hat{\sigma}_{d_i^-})), \mathbb{E}(w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+}) - \mathbb{E}(w_{d_i^-}\hat{\mu}_{d_i^-})\}, \tag{15}$$

where the $\hat{\mu}_*$ and $\hat{\sigma}_*$ denote the DNN output probability and its corresponding standard deviation respectively, w_* denotes the learned weight of DNN risk feature.

Similarly, as shown in Lemma 2, the value of ΔC^+ has an upper bound constrained by the weights of DNN outputs. The true negatives would tend to satisfy $\Delta VaR^+ - \Delta C^+ > 0$ in the case that the trained DNN model becomes less accurate. Hence, Theorem 2 shows that a false positive has a fairly good chance to be flipped form *matching* to *unmatching*.

Empirical Validation. We have illustrated the efficacy of theoretical analysis on the real literature dataset of DBLP-ACM[1] The results on the first iteration of risk-based fine-tuning are presented in Table 2, in which false negatives (resp. false positives) are clustered according to the size of true positives (resp. true negatives) that meet the specified condition. It can be observed: 1) risk-based fine-tuning rarely flips the labels of true positives and true negatives; 2) the majority of false negatives (resp. false positives) have a large number (e.g. ≥ 100) of corresponding true positives (resp. true negatives), and most of them are correctly flipped.

5. Empirical Study

In this section, we empirically evaluate the proposed approach on real benchmark datasets by a comparative study. We first describe the experimental setting, then present the comparative evaluation results, and finally evaluate robustness of the proposed approach w.r.t the size of validation data.

[1] https://github.com/anhaidgroup/deepmatcher/.

Table 3: The statistics of datasets.

Dataset	Size	# Matches	# Attributes
DS	28,707	5,347	4
DA	12,363	2,220	4
Cora	12,674	3,268	12
AB	9,575	1,028	3
IA	539	132	8
SG	19,633	6,108	7

5.1. Experimental Setup

We have used six real datasets from three domains in our empirical study:

- **Publication**. The datasets in this domain contain bibliographic data from different sources, i.e. DBLP, Google Scholar and ACM. As in [3], we use DBLP-Scholar[2] (denoted by DS) and DBLP-ACM [2] (denoted by DA). Additionally, we use the Cora dataset[3], which contains the citation data obtained from the Cora search engine;
- **Music**. In this domain, we use the Itunes-Amazon dataset (denoted by IA) provided by [3]. The size of IA is relatively small, containing only 539 pairs. Additionally, we use the Songs dataset[4] (denoted by SG), which contains song records. On SG, the experiments match the entries within the same table;
- **Product**. In this domain, we use the dataset containing the electronics product pairs extracted from Abt.com and Buy.com [2]. We denote this dataset by AB.

As usual, on all the datasets, we use the blocking technique to filter the pairs deemed unlikely to match. The datasets of DS, DA, AB and IA have been made online available at [2]. On both Cora and SG, we first filter the pairs and then randomly select a proportion of the resulting candidates to generate the workloads. We have provided the statistics of the test datasets in Table 3.

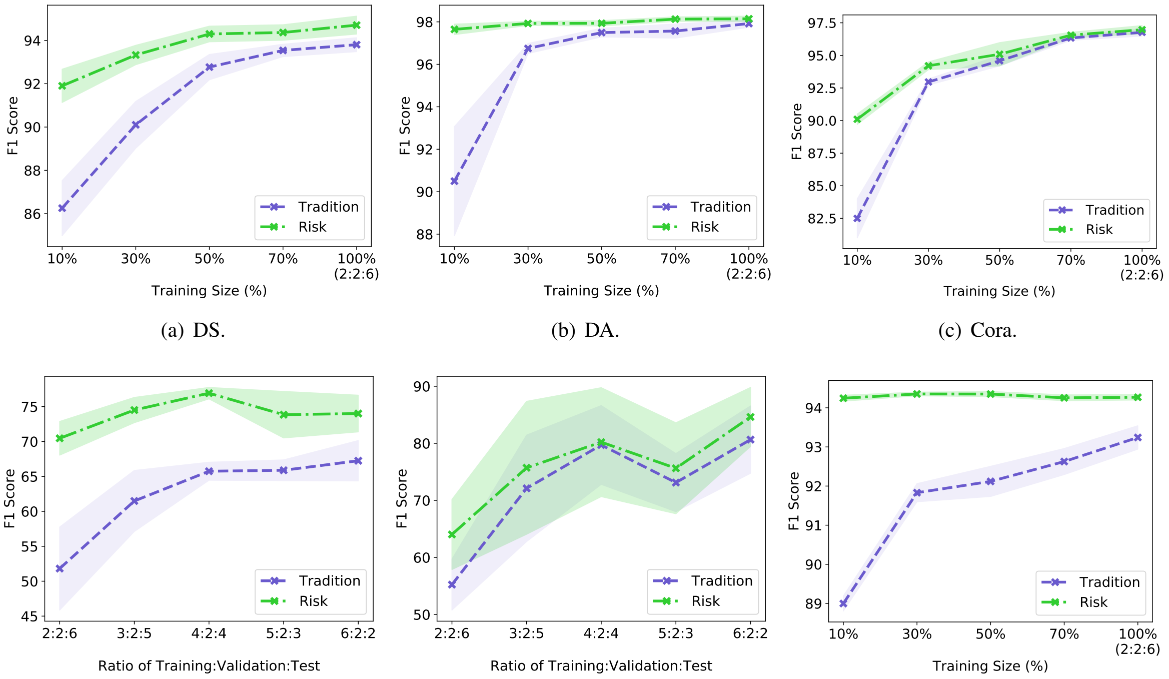
We evaluate the proposed approach in both scenarios where training and test data come from the same source and they come from different sources, thus resulting in more distribution misalignment. In the scenario where training and test data come from the same source, we randomly split each dataset into three parts by the ratio of 2:2:6 as in [3], which specifies the proportions of training, validation and test set respectively. On DA, DS, Cora and SG, deep models perform very well with the 20% split training data; therefore, we randomly select 10%, 30%, 50%, 70% and 100% of the split set of training data to simulate different sufficiency levels. On both AB and IA, we instead fix the proportion of validation data at 20% and vary the proportions of training and test data, resulting in totally 5 sufficiency levels of (50%,30%), (40%,40%), (30%,50%), (20%,60%), and (10%,70%). In this scenario, since training and target data are randomly selected from the same source, we compare the *Risk* approach with the original DeepMatcher, which is denoted by *Tradition*.

In the scenario of distribution misalignment, we use the three datasets in the domain of *publication* (i.e., DS, DA and Cora) to generate six pairwise workloads. For instance, DA2DS denotes the workload where training data come from DA while validation data and test data come from DS. On all the workloads, validation and test data are randomly selected from the original target dataset with both percentages set at 20%. In this scenario, besides *Tradition*, we also compare *Risk* with the technique of transfer learning for ER proposed in [37]. We denote this approach by *Transfer*. It inserts a dataset classifier into the DeepMatcher structure, which can force a deep model to focus on the parameters shared by both training and test data.

[2] https://github.com/anhaidgroup/deepmatcher/

[3] http://www.cs.utexas.edu/users/ml/riddle/data/cora.tar.gz

[4] http://pages.cs.wisc.edu/ãnhai/data/falcon_data/songs/



(a) DS.

(b) DA.

(c) Cora.

(d) AB.

(e) IA.

(f) SG.

Figure 3: Comparative Evaluation with Deepmatcher: the Same-Source Scenario.

We have implemented the proposed solution based on the baseline deep model, *DeepMatcher*, by replacing the original loss function with the risk-based loss function. To overcome the randomness caused by model initialization and training data shuffling, on each experiment, we perform 5 training sessions and report their mean F1-score on test data. In *Tradition* and *Transfer*, each training session consists of 20 iterations; in *Risk*, the traditional training phase consists of 20 iterations and the risk-based training phase consists of additional 10 iterations. Our experiments show that further increasing the number of iterations in each session has only marginal impact on performance. In our implementation, we use the default parameter setting of *DeepMatcher*. Specifically, we set the batch size at 32 and the learning rate at 0.001. The learning rate decays at a rate of 0.8 when the model stops improving. As the original *DeepMatcher*, we use a soft version of negative log likelihood loss and set the label smoothing parameter at 0.05.

Additionally, we have also implemented and evaluated the proposed solution based on *Ditto* [6], which is the state-of-the-art DNN for ER based on pre-trained Transformer-based language models. Note that compared with *DeepMatcher*, *Ditto* generally performs better and can perform well with less training data. Therefore, on AB and IA, our experiments begin with the ratio of training data at 10%. Similarly, we perform 5 training sessions and report their mean F1-score on test data to overcome the randomness. We use the default parameters of *Ditto* for both traditional training and adaptive training, except that the learning rate of risk-based training is set to be $3 * 10^{-6}$, instead of the default $3 * 10^{-5}$. Specifically, we set the batch size at 32 and the number of training epochs at 15. As the case of *DeepMatcher*, the risk-based tuning phase consists of 10 iterations while the traditional training phase has 20 iterations. Our implementations based on *DeepMatcher* and *Ditto* have been made open-source at our website [5].

5.2. Comparative Evaluation on DeepMatcher

5.2.1. Same-source Scenario

The comparative results are presented in Figure 3, in which we report both the mean of F1 score and its standard deviation (represented by the shadow in the figure). It can be observed that *Risk* achieves consistently better perfor-

[5] https://chenbenben.org/adaptive-training.html

Table 4: Comparative Evaluation with Deepmatcher: Distribution Misalignment.

Dataset	F1 Score (Mean ± Standard deviation)		
	Tradition	Transfer	Risk
DA2DS	19.86 ± 5.12	43.81 ± 11.88	**91.67**±0.56
DA2Cora	76.47 ± 4.59	74.86 ± 3.70	**89.08**±0.81
Cora2DS	55.81 ± 5.90	62.08 ± 6.65	**86.55**±1.34
Cora2DA	71.28 ± 5.23	72.92 ± 7.57	**96.99**±0.34
DS2DA	93.08 ± 1.71	93.50 ± 1.49	**94.18**±0.97
DS2Cora	83.11 ± 1.81	82.48 ± 3.52	**84.88**±0.29

mance than *Tradition*. On the workloads where *Tradition* performs unsatisfactorily (e.g. AB and IA), the performance margins between *Risk* and *Tradition* are very considerable. For instance, on AB, with the ratio of (2, 2, 6), *Risk* outperforms *Tradition* by around 20% in terms of F1 (70% vs 51%).

It can also be observed that on the workloads where *Tradition* can perform well (e.g. DS, DA, Cora and SG), the performance margins between *Risk* and *Tradition* are similarly considerable when training data are insufficient. For instance, on DS, with 10 percent of the training data, *Risk* outperforms *Tradition* by more than 6% and achieves the F1 score of more than 92%. In particular, on DA and SG, with only 10% of the training data, *Risk* achieves the performance very similar to what is achieved by using 100% of the training data. As the size of training data increases, the margins between *Risk* and *Tradition* tend to decrease. This trend can be expected, because when training and test data are randomly selected from the same source, more training data mean less improvement potential for risk-based fine-tuning. Due to the small size of IA, its comparative results have higher randomness compared with other datasets.

5.2.2. *Scenario of Distribution Misalignment*

The comparative results are presented in Table 4, in which the best results have been highlighted. It can be observed that: 1) the performance of *Tradition* deteriorates significantly on most testbeds; 2) the performance of *Transfer* fluctuates wildly across the test workloads. As shown on DS2DA and DS2CORA, where *Tradition* performs well, its impact becomes very marginal or even negative; 3) *Risk* consistently outperforms both *Tradition* and *Transfer*, and the margins are very considerable in most cases.

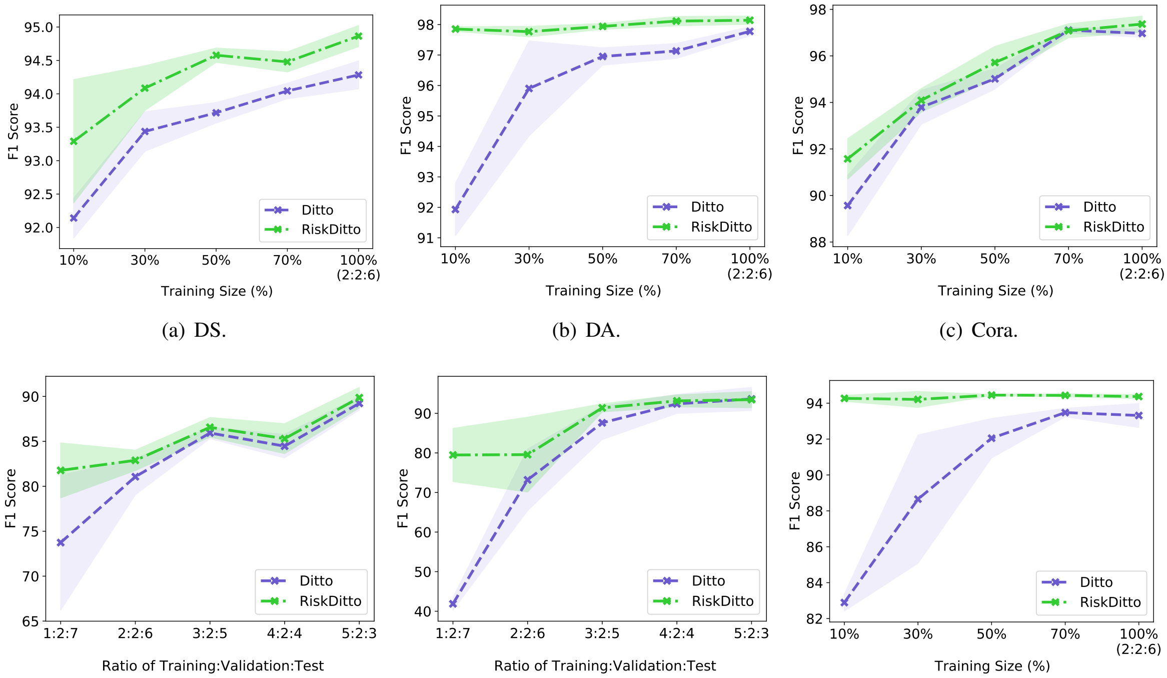
We explain the efficacy of the risk-based approach by illustrative examples. On DA2DS, we observe that the model trained on DA performs very poorly (only around 20%) on the target workload of DS. This is mainly due to the fact that DS is more challenging than DA, and the data distribution of DA to a large extent fails to reflect the more complicated distribution of DS. In contrast, LearnRisk can reliably identify the mispredictions of the pre-trained model on DS. We observe that in the first iteration of risk-based fine-tuning, it correctly identifies totally 877 mispredictions among the top 1000 risky pairs, most of which are later correctly flipped. However, on the workloads (e.g. DS2DA and DS2CORA) where *Tradition* performs well, the advantage of *Risk* over *Tradition* becomes less considerable. This result should be no surprise because in such circumstances, risk analysis becomes more challenging.

Furthermore, we have visualized the learned embeddings from the *attribute similarity representation* layer of *DeepMatcher* on the DA2DS workload in Figure 5. The blue crosses represent the training data and the red points represent the test data. It can be observed that, after tuning the *DeepMatcher* model based on risk analysis, feature representations become more compact and have more overlaps between training and test data as well, which mean training and target data are more aligned.

5.3. *Comparative Evaluation on Ditto*

5.3.1. *Same-source Scenario*

The comparative results on *Ditto* are presented in Figure 4. It can be observed that *Ditto* generally performs better than *Deepmatcher*. For instance, on AB, with the ratio setting of (2, 2, 6), the F1 score of *Ditto* is 81% while *Deepmatcher* can only achieve 51%. Similar to what have been observed on *DeepMatcher*, risk-based fine-tuning effectively improves the performance of *Ditto* even though it is a better baseline. For instance, on SG, with



(a) DS.

(b) DA.

(c) Cora.

(d) AB.

(e) IA.

(f) SG.

Figure 4: Comparative Evaluation with Ditto: the Same-Source Scenario.

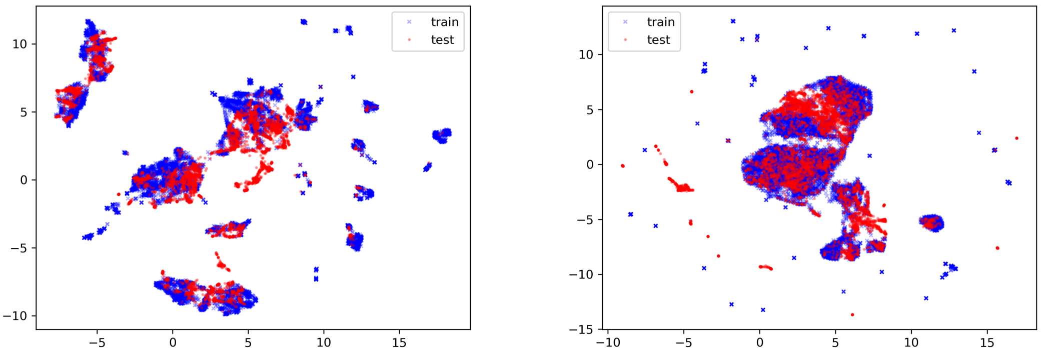
the percentage of training data at 10% and 30%, the performance improvements in terms of F1 are 11% and 6% respectively. The evaluation results on *Ditto* demonstrate clearly that the proposed approach of risk-based adaptive training is generally applicable to various DNN models.

5.3.2. Scenario of Distribution Misalignment

In the scenario of distribution misalignment, besides the *Tradition* and *Transfer*, we also compare *Risk* with the state-of-the-art domain adaptation approach named *Invgan+KD* presented in [39], which has been shown to perform better than its alternatives. The detailed comparative results are presented in Table 5, where *Tradition* represents the *Ditto*-based model. It can be observed that *Ditto* significantly outperforms *Deepmatcher* in the scenario of distribution misalignment. For example, on DA2DS, the F1 score of *Ditto* is more than 90% without employing any adaptation technique, while the F1 score of *Deepmatcher* is only 20%. It is noteworthy that *Risk* achieves the best performance on 4 out of totally 6 workloads, and its performance is very close to the best one on the other two workloads, DS2DA and DA2DS. In comparison, both *Transfer* and *Invgan+KD* achieves the best performance on only one, DS2DA and DA2DS respectively. It can be observed that on both DS2DA and DA2DS, the performance of the baseline *Ditto* is already very good, or more than 90%, and the improvement margins of *Transfer*, *Invgan+KD* and *Risk* are all very small. This observation is consistent with the evaluation results on *DeepMatcher*, which show that the advantage of *Risk* over a baseline deep model tends to decrease as the performance of the deep model improves.

5.4. Robustness w.r.t Size of Validation Data

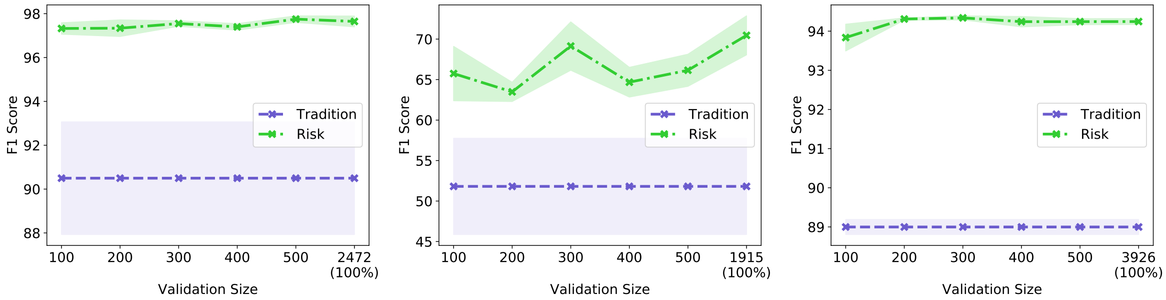
In real scenarios, validation data are necessary for hyperparameter tuning and model selection to ensure that a trained model can generalize well. However, due to labeling cost, it is usually desirable to reduce the size of validation data. Since risk analysis leverages validation data for risk model learning, we evaluate the performance robustness of the proposed approach w.r.t the size of validation data.



(a) Before Risk-tuning.

(b) After Risk-tuning.

Figure 5: Visualization of feature representations on DA2DS dataset.



(a) DA.

(b) AB.

(c) SG.

Figure 6: Robustness Evaluation.

To this end, we fix the sets of training and test data at 20% and 60% respectively, and vary the size of validation data by randomly selecting a proportion of instances from the split set of validation data. The results on the DA, AB and SG workloads are presented in Figure 6, in which the performance of *Tradition* and *Risk* with the whole set of validation data are also included for reference. The evaluation for DA and SG is based on the setting that 10% of the split set of training data is used. It can be observed that with as few as 100 validation instances, *Risk* is able to improve classifier performance by considerable margins. Our evaluation results are consistent with those reported in [21], which showed that the performance of LearnRisk is very robust w.r.t the size of validation data. These experimental results bode well for the application of the proposed approach in real scenarios.

6. Conclusion

In this paper, we have proposed a risk-based approach to enable adaptive deep learning for ER. It can effectively tune a deep model towards its target workload by the workload's particular characteristics. Both theoretical analysis and empirical study have validated its efficacy. For future work, it is worthy to point out that the proposed approach is generally applicable to other classification tasks; their technical solutions however need further investigation.

On the other hand, there are still two limitations w.r.t the proposed solution worthy of future investigations. First, our empirical study shows that the proposed solution's advantage over a baseline deep model tends to decrease as the performance of the deep model improves. This phenomenon is mainly due to the limited capability of the proposed

Table 5: Comparative Evaluation with Ditto: Distribution Misalignment.

Dataset	F1 Score (Mean ± Standard deviation)			
	Tradition	Transfer	Invgan+KD	Risk
DA2DS	90.69±1.15	89.77±0.88	**92.19±0.48**	91.35±0.76
DA2Cora	88.76±0.35	87.73±1.66	88.71±0.31	**89.84**± 0.53
Cora2DS	81.45±5.89	86.37±0.69	88.84±0.60	**88.91**±0.54
Cora2DA	88.87±3.89	94.43±1.01	93.48±1.33	**95.25**± 1.77
DS2DA	95 .28±1.54	**96.16**±0.40	95.98±0.13	95.87± 0.62
DS2Cora	84.63±0.06	84.73±0.32	84.86±0.77	**85.01**± 0.10

solution to generate knowledge beyond what can be discovered by deep models in the circumstances where labeled training data are sufficient. However, even a well-trained deep model may still make some obvious mistakes, which are easily detectable by common sense knowledge. Therefore, it is worthy to investigate how to incorporate common sense knowledge into the process of risk feature generation for the purpose of improving risk analysis, finally risk-based adaptive learning as well, in future work. Second, the current solution supposes that a target workload is readily available for risk fine-tuning. However, in some applications (e.g., network intrusion detection and automatic drive), target data may only become available incrementally. How to adapt risk fine-tuning for these scenarios also deserves further investigation.

Acknowledgments

This work was supported by the NSF of China (62172335, 61732014 and 61672432), the Fundamental Research Funds for the Central Universities (3102019DX1004).

References

References

[1] P. Christen, Automatic record linkage using seeded nearest neighbour and support vector machine classification, in: Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2008, pp. 151–159.

[2] M. Ebraheem, S. Thirumuruganathan, S. R. Joty, M. Ouzzani, N. Tang, Distributed representations of tuples for entity resolution, PVLDB 11 (11) (2018) 1454–1467.

[3] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, V. Raghavendra, Deep learning for entity matching: A design space exploration, in: Proceedings of the ACM International Conference on Management of Data (SIGMOD), 2018, pp. 19–34.

[4] H. Nie, X. Han, B. He, L. Sun, B. Chen, W. Zhang, S. Wu, H. Kong, Deep sequence-to-sequence entity matching for heterogeneous entity resolution, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 629–638.

[5] C. Zhao, Y. He, Auto-em: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning, in: The World Wide Web Conference, 2019, pp. 2413–2424.

[6] Y. Li, J. Li, Y. Suhara, A. Doan, W.-C. Tan, Deep entity matching with pre-trained language models, PVLDB 14 (1) (2021) 50–60.

[7] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering (TKDE) 22 (10) (2010) 1345–1359.

[8] Y. Wei, Y. Zhang, J. Huang, Q. Yang, Transfer learning via learning to transfer, in: Proceedings of the 35th International Conference on Machine Learning (ICML), Vol. 80, 2018, pp. 5072–5081.

[9] N. Houlsby, A. Giurgiu, S. Jastrzkebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for NLP, in: Proceedings of the 36th International Conference on Machine Learning (ICML), Vol. 97, 2019, pp. 2790–2799.

[10] M. Long, G. Ding, J. Wang, J. Sun, Y. Guo, P. S. Yu, Transfer sparse coding for robust image representation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 407–414.

[11] M. Long, Y. Cao, J. Wang, M. I. Jordan, Learning transferable features with deep adaptation networks, in: Proceedings of the 32nd International Conference on Machine Learning (ICML), Vol. 37, 2015, pp. 97–105.

[12] H. Zhao, R. T. des Combes, K. Zhang, G. J. Gordon, On learning invariant representations for domain adaptation, in: Proceedings of the 36th International Conference on Machine Learning (ICML), Vol. 97, 2019, pp. 7523–7532.

[13] Z. Wu, X. Wang, J. E. Gonzalez, T. Goldstein, L. S. Davis, Ace: Adapting to changing environments for semantic segmentation, in: Proceedings of the IEEE International Conference on Computer Vision (CVPR), 2019, pp. 2121–2130.

[14] T. Kim, M. Jeong, S. Kim, S. Choi, C. Kim, Diversify and match: A domain adaptive representation learning paradigm for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12456–12465.

[15] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: Proceedings of the 3rd International Conference on Learning Representations (ICLR), 2015, pp. 1–10.

[16] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, D. Mané, Concrete problems in ai safety, in: arXiv:1606.06565, 2016, pp. 1–29.

[17] D. Hendrycks, K. Gimpel, A baseline for detecting misclassified and out-of-distribution examples in neural networks, in: Proceedings of the 5th International Conference on Learning Representations (ICLR), 2017, pp. 1–12.

[18] Z. Chen, Q. Chen, B. Hou, M. Ahmed, Z. Li, Improving machine-based entity resolution with limited human effort: A risk perspective, in: Proceedings of the International Workshop on Real-Time Business Intelligence and Analytics, 2018, pp. 1–5.

[19] H. Jiang, B. Kim, M. Guan, M. Gupta, To trust or not to trust a classifier, in: Advances in Neural Information Processing Systems, 2018, pp. 5546–5557.

[20] D. Hendrycks, M. Mazeika, T. G. Dietterich, Deep anomaly detection with outlier exposure, in: Proceedings of the 7th International Conference on Learning Representations (ICLR), 2019, pp. 1–18.

[21] Z. Chen, Q. Chen, B. Hou, Z. Li, G. Li, Towards interpretable and learnable risk analysis for entity resolution, in: Proceedings of the ACM International Conference on Management of Data (SIGMOD), 2020, pp. 1165–1180.

[22] N. Barlaug, J. A. Gulla, Neural networks for entity matching: A survey, ACM Transactions on Knowledge Discovery from Data (TKDD) 15 (3) (2021) 1–37.

[23] P. Christen, Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection, Springer Science & Business Media, 2012, Ch. 2, pp. 32–34.

[24] V. Christophides, V. Efthymiou, K. Stefanidis, Entity resolution in the web of data, Synthesis Lectures on the Semantic Web 5 (3) (2015) 1–122.

[25] A. K. Elmagarmid, P. G. Ipeirotis, V. S. Verykios, Duplicate record detection: A survey, IEEE Transactions on Knowledge and Data Engineering (TKDE) 19 (1) (2007) 1–16.

[26] W. Fan, X. Jia, J. Li, S. Ma, Reasoning about record matching rules, PVLDB 2 (1) (2009) 407–418.

[27] L. Li, J. Li, H. Gao, Rule-based method for entity resolution, IEEE Transactions on Knowledge and Data Engineering (TKDE) 27 (1) (2015) 250–263.

[28] R. Singh, V. Meduri, A. Elmagarmid, S. Madden, P. Papotti, J.-A. Quiané-Ruiz, A. Solar-Lezama, N. Tang, Generating concise entity matching rules, in: Proceedings of the ACM International Conference on Management of Data (SIGMOD), 2017, pp. 1635–1638.

[29] I. P. Fellegi, A. B. Sunter, A theory for record linkage, Journal of the American Statistical Association 64 (328) (1969) 1183–1210.

[30] P. Singla, P. Domingos, Entity resolution with markov logic, in: Proceedings of the IEEE 6th International Conference on Data Mining (ICDM), 2006, pp. 572–582.

[31] M. Cochinwala, V. Kurien, G. Lalk, D. Shasha, Efficient data reconciliation, Information Sciences 137 (1-4) (2001) 1–15.

[32] P. Kouki, J. Pujara, C. Marcum, L. Koehly, L. Getoor, Collective entity resolution in familial networks, in: Proceedings of the IEEE International Conference on Data Mining (ICDM), 2017, pp. 227–236.

[33] S. Sarawagi, A. Bhamidipaty, Interactive deduplication using active learning, in: Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2002, pp. 269–278.

[34] B. Li, Y. Miao, Y. Wang, Y. Sun, W. Wang, Improving the efficiency and effectiveness for bert-based entity resolution, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 13226–13233.

[35] D. Yao, Y. Gu, G. Cong, H. Jin, X. Lv, Entity resolution with hierarchical graph attention networks, in: Proceedings of the 2022 International Conference on Management of Data, SIGMOD '22, 2022, p. 429–442.

[36] J. Huang, W. Hu, Z. Bao, Q. Chen, Y. Qu, Deep entity matching with adversarial active learning, The VLDB Journal (2022) 1–27.

[37] J. Kasai, K. Qian, S. Gurajada, Y. Li, L. Popa, Low-resource deep entity resolution with transfer and active learning, in: Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL), 2019, pp. 5851–5861.

[38] M. Trabelsi, J. Heflin, J. Cao, Dame: Domain adaptation for matching entities, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pp. 1016–1024.

[39] J. Tu, J. Fan, N. Tang, P. Wang, C. Chai, G. Li, R. Fan, X. Du, Domain adaptation for deep entity resolution, in: Proceedings of the 2022 International Conference on Management of Data, SIGMOD '22, 2022, p. 443–457.

[40] P. Zhang, J. Wang, A. Farhadi, M. Hebert, D. Parikh, Predicting failures of vision systems, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 3566–3573.

[41] Y. Nafa, Q. Chen, Z. Chen, X. Lu, H. He, T. Duan, Z. Li, Active deep learning on entity resolution by risk sampling, Knowledge-Based Systems 236 (2022) 107729.

[42] R. Kohavi, et al., A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI), Vol. 2, 1995, pp. 1137–1145.

[43] T. Evgeniou, M. Pontil, T. Poggio, Regularization networks and support vector machines, Advances in computational mathematics 13 (1) (2000) 1–50.

[44] P. Baldi, P. J. Sadowski, Understanding dropout, in: Advances in Neural Information Processing Systems, 2013, pp. 2814–2822.

[45] B. Neyshabur, S. Bhojanapalli, D. McAllester, N. Srebro, Exploring generalization in deep learning, in: Advances in Neural Information Processing Systems, 2017, pp. 5947–5956.

[46] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning requires rethinking generalization, in: Proceedings of the 5th International Conference on Learning Representations (ICLR), 2017, pp. 1–11.

[47] X. J. Zhu, Semi-supervised learning literature survey, Tech. rep., University of Wisconsin-Madison Department of Computer Sciences (2005).

[48] A. Iscen, G. Tolias, Y. Avrithis, O. Chum, Label propagation for deep semi-supervised learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5070–5079.

[49] B. Settles, Active learning, Synthesis Lectures on Artificial Intelligence and Machine Learning 6 (1) (2012) 1–114.

[50] M. Ducoffe, F. Precioso, Adversarial active learning for deep networks: a margin based approach, in: arXiv:1802.09841, 2018, pp. 1–10.

[51] Z.-H. Zhou, Ensemble learning., Encyclopedia of biometrics 1 (2009) 270–273.

[52] O. Sagi, L. Rokach, Ensemble learning: A survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (4) (2018) e1249.

[53] L. Breiman, Bagging predictors, Machine learning 24 (2) (1996) 123–140.

[54] R. E. Schapire, The strength of weak learnability, Machine learning 5 (2) (1990) 197–227.

[55] G. Tardivo, Value at risk (var): The new benchmark for managing market risk, Journal of Financial Management & Analysis 15 (1) (2002) 16–26.

[56] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations (ICLR), 2015, pp. 1–11.

[57] C. McDiarmid, On the method of bounded differences, London Mathematical Society Lecture Note Series, Cambridge University Press, 1989, p. 148–188.

Appendix

In theoretical analysis, for simplicity of presentation, without loss of generality, we suppose that *LearnRisk* sets the confidence value at $\theta = 0.975$. Hence, given a pair d_i with the equivalence probability distribution of $\mathcal{N}(\mu_i, \sigma_i^2)$, its VaR risk is equal to $1 - (\mu_i - 2\sigma_i)$ if it is labeled as *matching* by a classifier, and its VaR risk is equal to $\mu_i + 2\sigma_i$ if it is labeled as *unmatching*.

Appendix .1. Proof of Theorem 1

Theorem 1 *Given a false negative $\tilde{d}_j^-$, suppose that there are totally n true positives, denoted by d_i^+, ranked after $\tilde{d}_j^-$ by LearnRisk such that each true positive, d_i^+, satisfies*

$$\Delta VaR^- - \Delta C^- > \epsilon, \tag{.1}$$

in which $\Delta VaR^- = VaR^-(\tilde{d}_j^-) - VaR^+(d_i^+)$, and $\Delta C^- = \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-})$. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, its expectation of equivalence probability of $\tilde{d}_j^-$, $\mu_{\tilde{d}_j^-}$, estimated by LearnRisk*, satisfies*

$$\mu_{\tilde{d}_j^-} \geq \frac{1}{2} + \frac{\epsilon}{2} - \sqrt{\frac{m+1}{2} ln[\frac{1}{1-(1-\delta^{\frac{1}{n}})^{\frac{1}{2}}}]},$$

in which μ_* denotes the mean of equivalence probability and σ_* denotes its standard deviation.

Note that in Theorem 1, m denotes the number of rule risk features and ΔC^- denotes the difference of risk expectation between false negatives being labeled as *matching* and true positives being labeled as *matching*. In the rest of this section, we first prove a lemma that states the concentration inequalities of VaR risk functions, and then prove Theorem 1 based on the lemma.

Lemma 3. *Given a randomly selected pair d_i^+ from true positives, whose equivalence probability distribution estimated by* LearnRisk *is denoted by $\mathcal{N}(\mu_{d_i^+}, \sigma_{d_i^+})$, for any $\delta \in (0, 1)$, with probability at least $(1 - \delta)$, the following inequality holds*

$$(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) \leq \varepsilon,$$

where $\varepsilon = \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})}$. Similarly, for a randomly selected false negative $\tilde{d}_j^-$ with equivalence probability distribution of $\mathcal{N}(\mu_{\tilde{d}_j^-}, \sigma_{\tilde{d}_j^-})$, with probability at least $(1 - \delta)$, the following inequality holds

$$\mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) - (\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) \leq \varepsilon.$$

Proof. Consider the randomly selected pair d_i^+ from true positives. The mean of its equivalence probability can be represented by

$$\mu_{d_i^+} = \frac{\sum\limits_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{d_i^+} \hat{\mu}_{d_i^+}}{\sum\limits_{k=1}^{m} w_k Z_k + w_{d_i^+}},$$

where m is the number of rule risk features, w_k is the learned weight of a risk feature f_k, μ_{f_k} is the probability mean of the feature, Z_k is a random variable indicates if a selected pair has this feature, and $\hat{\mu}_{d_i^+}$ is the output probability by a classifier with its weight $w_{d_i^+}$. Note that for a randomly selected true positive, the values of w_k and μ_{f_k} for each rule risk feature are fixed, while Z_k, $\hat{\mu}_{d_i^+}$ and $w_{d_i^+}$ are random variables. Note that according to LearnRisk, the value of $w_{d_i^+}$ totally depends on $\hat{\mu}_{d_i^+}$.

The standard deviation of its equivalence probability can also be represented by

$$\sigma_{d_i^+} = \frac{1}{\sum\limits_{k=1}^{m} w_k Z_k + w_{d_i^+}} \sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{d_i^+}^2 \hat{\sigma}_{d_i^+}^2},$$

where $\hat{\sigma}_{d_i^+}^2$ denotes the corresponding variance of a classifier's output $\hat{\mu}_{d_i^+}$.

Recall that a function $f : X^n \to \mathbb{R}$ has the *bounded differences property* if for some non-negative constants $c_1, c_2, ..., c_n$,

$$\sup_{x_1,...,x_n,x_k' \in X} |f(x_1, ..., x_{k-1}, x_k, x_{k+1}, ..., x_n) - f(x_1, ..., x_{k-1}, x_k', x_{k+1}, ..., x_n)| \le c_k, 1 \le k \le n$$

The bounded differences property shows that if the ith variable is changed while all the others being fixed, the value of f will not change by more than c_k.

Let $f(Z_1, ..., Z_m, \hat{\mu}_{d_i^+}) = \mu_{d_i^+} - 2\sigma_{d_i^+}$. Now we proceed to consider the bounded differences property of f. Note that a valid equivalence probability should be between 0 and 1. Hence, for all $\mu \ge 0, \sigma \ge 0$, we have $0 \le \mu \pm 2\sigma \le 1$. As a result, by changing the value of Z_k, we have

$$sup|f(z_1, ..., z_k, ..., z_m, \hat{\mu}_{d_i^+}) - f(z_1, ..., z_k', ..., z_m, \hat{\mu}_{d_i^+})| \le 1$$

Similarly, the upper bound of f by changing the value of $\hat{\mu}_{d_i^+}$ is

$$sup|f(z_1, ..., z_m, \hat{\mu}_{d_i^+}) - f(z_1, ..., z_m, \hat{\mu}'_{d_i^+})| \le 1$$

At this point, we have obtained the upper bounds of the function $f(Z_1, ..., Z_m, \hat{\mu}_{d_i^+})$ by changing any one of the variables. Denoting these bounds as $c_1, ..., c_m, c_{m+1}$, where $c_k = 1, 1 \le k \le m + 1$, we have

$$\sum_{k=1}^{m+1} c_k^2 = m + 1$$

Recall that the McDiarmid's inequality [57] states that if a function f satisfies the bounded differences property with constants $c_1, ..., c_n$. Let $Y = f(X_1, ..., X_n)$, where the X_ks are independent random variables. Then, for all $\varepsilon > 0$,

$$\mathbb{P}(Y - \mathbb{E}Y \ge \varepsilon) \le exp(-\frac{2\varepsilon^2}{\sum_{k=1}^{n} c_k^2});$$

$$\mathbb{P}(\mathbb{E}Y - Y \ge \varepsilon) \le exp(-\frac{2\varepsilon^2}{\sum_{k=1}^{n} c_k^2}).$$

Based on the McDiarmid's inequality, for all $\varepsilon > 0$, we have

$$\mathbb{P}(\mu_{d_i^+} - 2\sigma_{d_i^+} - \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) \ge \varepsilon) \le exp(-\frac{2\varepsilon^2}{\sum_{k=1}^{m+1} c_k^2}) = exp(-\frac{2\varepsilon^2}{m+1}).$$

Let $\delta = exp(-\frac{2\varepsilon^2}{m+1})$, we can get that $\varepsilon = \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})}$. Hence, with the probability at least $1 - \delta$, the inequality $\mu_{d_i^+} - 2\sigma_{d_i^+} - \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) \le \varepsilon$ holds.

Similarly, with the probability at least $1 - \delta$, we have $\mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) - (\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) \le \varepsilon$.

In the following, we present the proof of Theorem 1.

Proof. **[Theorem 1]** According to Lemma 3, with the probability at least $(1-\delta)^2$, the following inequalities hold

$$(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) + \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) - (\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) \leq 2\varepsilon;$$
$$(\mu_{d_i^+} - 2\sigma_{d_i^+} - \mu_{\tilde{d}_j^-} + 2\sigma_{\tilde{d}_j^-}) + \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) - \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) \leq 2\varepsilon;$$

Hence, we have

$$\mu_{d_i^+} - 2\sigma_{d_i^+} - \mu_{\tilde{d}_j^-} + 2\sigma_{\tilde{d}_j^-} \leq 2\varepsilon + [\mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-})], \tag{.2}$$

where $\varepsilon = \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})}$. We denote

$$\Delta C^- = \mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}) = \mathbb{E}(1 - (\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-})) - \mathbb{E}(1 - (\mu_{d_i^+} - 2\sigma_{d_i^+})), \tag{.3}$$

where $\mathbb{E}(1-(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}))$ is the risk expectation of false negatives being labeled as *matching* and $\mathbb{E}(1-(\mu_{d_i^+} - 2\sigma_{d_i^+}))$ is the risk expectation of true positives being labeled as *matching*. Based on the definition of VaR, we have $VaR^-(\tilde{d}_j^-) = \mu_{\tilde{d}_j^-} + 2\sigma_{\tilde{d}_j^-}$, and $VaR^+(d_i^+) = 1 - (\mu_{d_i^+} - 2\sigma_{d_i^+})$. Denoting $\Delta VaR^- = VaR^-(\tilde{d}_j^-) - VaR^+(d_i^+)$, we have,

$$\begin{aligned} 1 + \Delta VaR^- =& \mu_{\tilde{d}_j^-} + 2\sigma_{\tilde{d}_j^-} + \mu_{d_i^+} - 2\sigma_{d_i^+} \\ \leq& \mu_{\tilde{d}_j^-} + \mu_{\tilde{d}_j^-} + 2\varepsilon + [\mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-})] \\ =& 2\mu_{\tilde{d}_j^-} + 2\varepsilon + \Delta C^-. \end{aligned} \tag{.4}$$

Hence, for a randomly selected false negative and a randomly selected true positive, with probability at least $(1-\delta)^2$, we have

$$\mu_{\tilde{d}_j^-} \geq \frac{1}{2} + \frac{\Delta VaR^-}{2} - \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})} - \frac{\Delta C^-}{2}. \tag{.5}$$

Note that the probability of the above inequality does not hold is $[1-(1-\delta)^2]$. Suppose that there are totally n true positives d_i^+ ranked after $\tilde{d}_j^-$ by LearnRisk such that each true positive, d_i^+, satisfies $\Delta VaR^- - \Delta C^- > \epsilon$. Then the probability of Inequality .5 fails can be approximated by $[1-(1-\delta)^2]^n$. That is, the probability of at least one of the true positives can support the Inequality .5 is $\{1-[1-(1-\delta)^2]^n\}$. Let $1-[1-(1-\delta)^2]^n = 1-\delta'$, we can get $\delta = 1 - \sqrt{1-\delta'^{\frac{1}{n}}}$. Therefore, with probability at least $(1-\delta)$,

$$\mu_{\tilde{d}_j^-} \geq \frac{1}{2} + \frac{\epsilon}{2} - \sqrt{\frac{m+1}{2} ln[\frac{1}{1-(1-\delta^{\frac{1}{n}})^{\frac{1}{2}}}]}. \tag{.6}$$

Note that the total number of rule risk features (m) is usually limited (e.g., dozens or hundreds), while n is usually much larger than m. By the exponential effect of n, the 3rd term on the right-hand side tends to become zero as the value of n increases.

Appendix .2. Proof of Lemma 1

Lemma 1

$$\Delta C^- \leq max\{\mathbb{E}(w_{d_i^+}(\hat{\mu}_{d_i^+} - 2\hat{\sigma}_{d_i^+})) - \mathbb{E}(w_{\tilde{d}_j^-}(\hat{\mu}_{\tilde{d}_j^-} - 2\hat{\sigma}_{\tilde{d}_j^-})), \mathbb{E}(w_{d_i^+}\hat{\mu}_{d_i^+}) - \mathbb{E}(w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-})\}, \tag{.7}$$

where the $\hat{\mu}_*$ and $\hat{\sigma}_*$ denote the DNN output probability and its corresponding standard deviation respectively, w_* denotes the learned weight of DNN risk feature.

Proof. For simplicity of presentation, let $N_{d_i^+}$ denote the weight normalization factor of d_i^+, or $N_{d_i^+} = \sum_{k=1}^{m} w_k Z_k + w_{d_i^+}$. Similarly, let $N_{\tilde{d}_j^-}$ denote the weight normalization factor of $\tilde{d}_j^-$, or $N_{\tilde{d}_j^-} = \sum_{k=1}^{m} w_k Z_k + w_{\tilde{d}_j^-}$. According to the weight function defined by *LearnRisk* [21], without loss of generality, we suppose that $w_{d_i^+} = w_{\tilde{d}_j^-}$. As a result, $N_{d_i^+} = N_{\tilde{d}_j^-}$. Based on Assumption 1, we have,

$$
\begin{aligned}
&\mathbb{E}(\mu_{d_i^+} - 2\sigma_{d_i^+}) - \mathbb{E}(\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-})\\
=&\mathbb{E}((\mu_{d_i^+} - 2\sigma_{d_i^+}) - (\mu_{\tilde{d}_j^-} - 2\sigma_{\tilde{d}_j^-}))\\
=&\mathbb{E}\Bigg(\frac{1}{N_{d_i^+}}\Bigg(\sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{d_i^+}\hat{\mu}_{d_i^+} - 2\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{d_i^+}^2 \hat{\sigma}_{d_i^+}^2}\Bigg) -\\
&\frac{1}{N_{\tilde{d}_j^-}}\Bigg(\sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-} - 2\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{\tilde{d}_j^-}^2 \hat{\sigma}_{\tilde{d}_j^-}^2}\Bigg)\Bigg)\\
=&\mathbb{E}\Bigg(\frac{1}{N_{d_i^+}}\Bigg(\sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{d_i^+}\hat{\mu}_{d_i^+} - 2\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{d_i^+}^2 \hat{\sigma}_{d_i^+}^2} -\\
&\sum_{k=1}^{m} w_k \mu_{f_k} Z_k - w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-} + 2\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{\tilde{d}_j^-}^2 \hat{\sigma}_{\tilde{d}_j^-}^2}\Bigg)\Bigg)\\
=&\mathbb{E}\Bigg(\frac{1}{N_{d_i^+}}\Bigg(w_{d_i^+}\hat{\mu}_{d_i^+} - w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-} + 2\Bigg[\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{\tilde{d}_j^-}^2 \hat{\sigma}_{\tilde{d}_j^-}^2} - \sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{d_i^+}^2 \hat{\sigma}_{d_i^+}^2}\Bigg]\Bigg)\Bigg) \quad \cdots (S_1)
\end{aligned}
$$

If $w_{\tilde{d}_j^-}\hat{\sigma}_{\tilde{d}_j^-} < w_{d_i^+}\hat{\sigma}_{d_i^+}$, then

$$S_1 \leq \mathbb{E}(\frac{1}{N_{d_i^+}}(w_{d_i^+}\hat{\mu}_{d_i^+} - w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-})).$$

If $w_{\tilde{d}_j^-}\hat{\sigma}_{\tilde{d}_j^-} \geq w_{d_i^+}\hat{\sigma}_{d_i^+}$, then

$$
\begin{aligned}
S_1 &\leq \mathbb{E}\Bigg(\frac{1}{N_{d_i^+}}\Bigg(w_{d_i^+}\hat{\mu}_{d_i^+} - w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-} + 2\Bigg[\sqrt{w_{\tilde{d}_j^-}^2 \hat{\sigma}_{\tilde{d}_j^-}^2} - \sqrt{w_{d_i^+}^2 \hat{\sigma}_{d_i^+}^2}\Bigg]\Bigg)\Bigg) \quad \cdots (S_2)\\
&= \mathbb{E}(\frac{w_{d_i^+}}{N_{d_i^+}}(\hat{\mu}_{d_i^+} - 2\hat{\sigma}_{d_i^+})) - \mathbb{E}(\frac{w_{\tilde{d}_j^-}}{N_{d_i^+}}(\hat{\mu}_{\tilde{d}_j^-} - 2\hat{\sigma}_{\tilde{d}_j^-}))
\end{aligned}
$$

From step S_1 to step S_2, we apply the rule that if $a \geq 0, b \geq 0, c \geq 0$ and $b \geq c$, then $\sqrt{a+b} - \sqrt{a+c} \leq \sqrt{b} - \sqrt{c}$. For simplicity of presentation, we denote the normalization of $\frac{w_{d_i^+}}{N_{d_i^+}}$ by $w_{d_i^+}$, and similarly, the normalized $w_{\tilde{d}_j^-}$.

Hence, we have

$$\Delta C^- \leq max\{\mathbb{E}(w_{d_i^+}(\hat{\mu}_{d_i^+} - 2\hat{\sigma}_{d_i^+})) - \mathbb{E}(w_{\tilde{d}_j^-}(\hat{\mu}_{\tilde{d}_j^-} - 2\hat{\sigma}_{\tilde{d}_j^-})), \mathbb{E}(w_{d_i^+}\hat{\mu}_{d_i^+}) - \mathbb{E}(w_{\tilde{d}_j^-}\hat{\mu}_{\tilde{d}_j^-})\},$$

where the $\hat{\mu}_*$ and $\hat{\sigma}_*$ denote the DNN output probability and its corresponding standard deviation respectively, w_* denotes the learned weight of DNN risk feature.

Appendix .3. Proof of Theorem 2

Similarly, based on Assumption 1, we theoretically analyze the chance of a false positive being flipped from *matching* to *unmatching*. We first prove a lemma, and then prove Theorem 2 based on the lemma.

Lemma 4. *For a randomly selected pair d_i^- from true negatives, we denote the mean of its equivalence probability by $\mu_{d_i^-}$, and the corresponding standard deviation by $\sigma_{d_i^-}$. For any $\delta \in (0, 1)$, with probability at least $(1 - \delta)$, the following inequality holds*

$$\mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-}) - (\mu_{d_i^-} + 2\sigma_{d_i^-}) \leq \varepsilon,$$

where $\varepsilon = \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})}$, m denotes the total number of rule risk features. Similarly, for a randomly selected false positive $\tilde{d}_j^+$ with the equivalence probability mean of $\mu_{\tilde{d}_j^+}$ and the standard deviation of $\sigma_{\tilde{d}_j^+}$, with probability at least $(1 - \delta)$, the following inequality holds

$$(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) \leq \varepsilon.$$

Proof. Consider a randomly selected pair d_i^- from true negatives. The mean of its equivalence probability can be represented by

$$\mu_{d_i^-} = \frac{\sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{d_i^-} \hat{\mu}_{d_i^-}}{\sum_{k=1}^{m} w_k Z_k + w_{d_i^-}},$$

where m denotes the number of rule risk features, w_k denotes the weight of a risk feature f_k, μ_{f_k} is the equivalence probability mean of the feature f_k, Z_k is a random variable indicates if a selected pair has this feature, and $\hat{\mu}_{d_i^-}$ is the output probability by a classifier with its weight $w_{d_i^-}$. Note that for a randomly selected true negative, the values of w_k and μ_{f_k} for each risk feature are fixed, while Z_k, $\hat{\mu}_{d_i^-}$ are random variables. Note that the value of $w_{d_i^-}$ totally depends $\hat{\mu}_{d_i^-}$.

The standard deviation of its equivalence probability can also be represented by

$$\sigma_{d_i^-} = \frac{1}{\sum_{k=1}^{m} w_k Z_k + w_{d_i^-}} \sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{d_i^-}^2 \hat{\sigma}_{d_i^-}^2},$$

where $\hat{\sigma}_{d_i^-}^2$ denotes the corresponding variance of a classifier's output $\hat{\mu}_{d_i^-}$.

Let $f(Z_1, ..., Z_m, \hat{\mu}_{d_i^-}) = \mu_{d_i^-} + 2\sigma_{d_i^-}$. Now we proceed to consider the bounded differences property of f. As in the proof of lemma 1, for all $\mu \geq 0, \sigma \geq 0$, we have $0 \leq \mu \pm 2\sigma \leq 1$. Hence, by changing the value of Z_k, we have

$$sup|f(z_1, ..., z_k, ..., z_m, \hat{\mu}_{d_i^-}) - f(z_1, ..., z_k', ..., z_m, \hat{\mu}_{d_i^-})| \leq 1$$

Similarly, the upper bound of f by changing the value of $\hat{\mu}_{d_i^-}$ is,

$$sup|f(z_1, ..., z_m, \hat{\mu}_{d_i^-}) - f(z_1, ..., z_m, \hat{\mu}'_{d_i^-})| \leq 1$$

At this point, we have obtained the upper bounds of function $f(Z_1, ..., Z_m, \hat{\mu}_{d_i^-})$ by changing any one of the variables. Denoting these bounds by $c_1, ..., c_m, c_{m+1}$, where $c_k = 1, 1 \leq k \leq m + 1$, we have

$$\sum_{k=1}^{m+1} c_k^2 = m + 1.$$

By applying the McDiarmid's inequality, for all $\varepsilon > 0$, we have

$$\mathbb{P}(\mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-}) - (\mu_{d_i^-} + 2\sigma_{d_i^-}) \geq \varepsilon) \leq exp(-\frac{2\varepsilon^2}{\sum_{k=1}^{m+1} c_k^2}) = exp(-\frac{2\varepsilon^2}{m+1}).$$

Let $\delta = exp(-\frac{2\varepsilon^2}{m+1})$, we can get that $\varepsilon = \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})}$. Hence, with the probability at least $1 - \delta$, the inequality $\mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-}) - (\mu_{d_i^-} + 2\sigma_{d_i^-}) \leq \varepsilon$ holds.

Similarly, with the probability at least $1 - \delta$, we have $(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) \leq \varepsilon$.

Theorem 2 *Given a false positive $\tilde{d}_j^+$, suppose that there are totally n true negatives, denoted by d_i^-, ranked after $\tilde{d}_j^+$ by LearnRisk such that each true negative, d_i^-, satisfies*

$$\Delta VaR^+ - \Delta C^+ > \epsilon, \tag{.8}$$

in which $\Delta VaR^+ = VaR^+(\tilde{d}_j^+) - VaR^-(d_i^-)$, and $\Delta C^+ = \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-})$. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, its expectation of equivalence probability of $\tilde{d}_j^+$, $\mu_{\tilde{d}_j^+}$ estimated by LearnRisk*, satisfies*

$$\mu_{\tilde{d}_j^+} \leq \frac{1}{2} - \frac{\epsilon}{2} + \sqrt{\frac{m+1}{2} ln[\frac{1}{1-(1-\delta^{\frac{1}{n}})^{\frac{1}{2}}}]},$$

in which μ_* denotes the mean of equivalence probability and σ_* denotes its standard deviation.

Proof. With Lemma 4, with probability at least $(1 - \delta)^2$, the following inequalities hold

$$(\mu_{d_i^-} + 2\sigma_{d_i^-}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-}) + \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - (\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) \geq -2\varepsilon;$$
$$(\mu_{d_i^-} + 2\sigma_{d_i^-} - \mu_{\tilde{d}_j^+} - 2\sigma_{\tilde{d}_j^+}) + \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-}) \geq -2\varepsilon;$$

Hence, we have

$$\mu_{d_i^-} + 2\sigma_{d_i^-} - \mu_{\tilde{d}_j^+} - 2\sigma_{\tilde{d}_j^+} \geq -2\varepsilon - [\mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-})], \tag{.9}$$

where $\varepsilon = \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})}$. We denote

$$\Delta C^+ = \mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-}), \tag{.10}$$

where $\mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+})$ is the risk expectation of false positives being labeled as *unmatching* and $\mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-})$ is the risk expectation of true negatives being labeled as *unmatching*. Based on the definition of VaR, we have $VaR^+(\tilde{d}_j^+) = 1 - (\mu_{\tilde{d}_j^+} - 2\sigma_{\tilde{d}_j^+})$, and $VaR^-(d_i^-) = \mu_{d_i^-} + 2\sigma_{d_i^-}$. Denoting $\Delta VaR^+ = VaR^+(\tilde{d}_j^+) - VaR^-(d_i^-)$, we have

$$\begin{aligned} 1 - \Delta VaR^+ =& \mu_{d_i^-} + 2\sigma_{d_i^-} + \mu_{\tilde{d}_j^+} - 2\sigma_{\tilde{d}_j^+} \\ \geq& \mu_{\tilde{d}_j^+} + \mu_{\tilde{d}_j^+} - 2\varepsilon - [\mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-})] \\ =& 2\mu_{\tilde{d}_j^+} - 2\varepsilon - \Delta C^+. \end{aligned} \tag{.11}$$

In Equation .11, the inequality is obtained by applying the Inequality .9. Hence, for a randomly selected false positive and a randomly selected true negative, with probability at least $(1 - \delta)^2$, the following inequality holds

$$\mu_{\tilde{d}_j^+} \leq \frac{1}{2} - \frac{\Delta VaR^+}{2} + \sqrt{\frac{m+1}{2} ln(\frac{1}{\delta})} + \frac{\Delta C^+}{2}. \tag{.12}$$

Note that the probability of the above inequality does not hold is $[1-(1-\delta)^2]$. Suppose that there are totally n true negatives, denoted by d_i^-, ranked after $\tilde{d}_j^+$ by LearnRisk such that each true negative, d_i^-, satisfies $\Delta VaR^+ - \Delta C^+ > \epsilon$. Then the probability of Inequality .12 fails can be approximated by $[1 - (1 - \delta)^2]^n$. That is, the probability of at least one of the true negatives can support the Inequality .12 is $\{1 - [1 - (1 - \delta)^2]^n\}$. Let $1 - [1 - (1 - \delta)^2]^n = 1 - \delta'$, we can get $\delta = 1 - \sqrt{1 - \delta'^{\frac{1}{n}}}$. Therefore, with probability at least $(1 - \delta)$, we have

$$\mu_{\tilde{d}_j^+} \leq \frac{1}{2} - \frac{\epsilon}{2} + \sqrt{\frac{m+1}{2} ln[\frac{1}{1-(1-\delta^{\frac{1}{n}})^{\frac{1}{2}}}]}, \tag{.13}$$

Appendix .4. Proof of Lemma 2

Lemma 2

$$\Delta C^{+} \leq max\{\mathbb{E}(w_{\tilde{d}_j^+}(\hat{\mu}_{\tilde{d}_j^+} + 2\hat{\sigma}_{\tilde{d}_j^+})) - \mathbb{E}(w_{d_i^-}(\hat{\mu}_{d_i^-} + 2\hat{\sigma}_{d_i^-})), \mathbb{E}(w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+}) - \mathbb{E}(w_{d_i^-}\hat{\mu}_{d_i^-})\}, \tag{.14}$$

where the $\hat{\mu}_*$ and $\hat{\sigma}_*$ denote the DNN output probability and its corresponding standard deviation respectively, w_* denotes the learned weight of DNN risk feature.

Proof. For simplicity of presentation, let $N_{\tilde{d}_j^+}$ denote the weight normalization factor of $\tilde{d}_j^+$, $N_{\tilde{d}_j^+} = \sum_{k=1}^{m} w_k Z_k + w_{\tilde{d}_j^+}$. Similarly, $N_{d_i^-}$ denote the weight normalization factor of d_i^-, $N_{d_i^-} = \sum_{k=1}^{m} w_k Z_k + w_{d_i^-}$. As in the proof of Lemma 1, we suppose that $N_{\tilde{d}_j^+} = N_{d_i^-}$. Based on Assumption 1, we have,

$$\begin{aligned}
&\mathbb{E}(\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - \mathbb{E}(\mu_{d_i^-} + 2\sigma_{d_i^-})\\
=&\mathbb{E}((\mu_{\tilde{d}_j^+} + 2\sigma_{\tilde{d}_j^+}) - (\mu_{d_i^-} + 2\sigma_{d_i^-}))\\
=&\mathbb{E}\Bigg(\frac{1}{N_{\tilde{d}_j^+}}\Bigg(\sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+} + 2\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{\tilde{d}_j^+}^2 \hat{\sigma}_{\tilde{d}_j^+}^2}\Bigg) -\\
&\frac{1}{N_{d_i^-}}\Bigg(\sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{d_i^-}\hat{\mu}_{d_i^-} + 2\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{d_i^-}^2 \hat{\sigma}_{d_i^-}^2}\Bigg)\Bigg)\\
=&\mathbb{E}\Bigg(\frac{1}{N_{\tilde{d}_j^+}}\Bigg(\sum_{k=1}^{m} w_k \mu_{f_k} Z_k + w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+} + 2\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{\tilde{d}_j^+}^2 \hat{\sigma}_{\tilde{d}_j^+}^2} -\\
&\sum_{k=1}^{m} w_k \mu_{f_k} Z_k - w_{d_i^-}\hat{\mu}_{d_i^-} - 2\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{d_i^-}^2 \hat{\sigma}_{d_i^-}^2}\Bigg)\Bigg)\\
=&\mathbb{E}\Bigg(\frac{1}{N_{\tilde{d}_j^+}}\Bigg(w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+} - w_{d_i^-}\hat{\mu}_{d_i^-} + 2\Bigg[\sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{\tilde{d}_j^+}^2 \hat{\sigma}_{\tilde{d}_j^+}^2} - \sqrt{\sum_{k=1}^{m} w_k^2 \sigma_{f_k}^2 Z_k + w_{d_i^-}^2 \hat{\sigma}_{d_i^-}^2}\Bigg]\Bigg)\Bigg) \quad \cdots (S_3)
\end{aligned}$$

If $w_{\tilde{d}_j^+}\hat{\sigma}_{\tilde{d}_j^+} < w_{d_i^-}\hat{\sigma}_{d_i^-}$, then

$$S_3 \leq \mathbb{E}(\frac{1}{N_{\tilde{d}_j^+}}(w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+} - w_{d_i^-}\hat{\mu}_{d_i^-})).$$

If $w_{\tilde{d}_j^+}\hat{\sigma}_{\tilde{d}_j^+} \geq w_{d_i^-}\hat{\sigma}_{d_i^-}$, then

$$\begin{aligned}
S_3 \leq& \mathbb{E}\Bigg(\frac{1}{N_{\tilde{d}_j^+}}\Bigg(w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+} - w_{d_i^-}\hat{\mu}_{d_i^-} + 2\Bigg[\sqrt{w_{\tilde{d}_j^+}^2 \hat{\sigma}_{\tilde{d}_j^+}^2} - \sqrt{w_{d_i^-}^2 \hat{\sigma}_{d_i^-}^2}\Bigg]\Bigg)\Bigg) \quad \cdots (S_4)\\
=& \mathbb{E}(\frac{w_{\tilde{d}_j^+}}{N_{\tilde{d}_j^+}}(\hat{\mu}_{\tilde{d}_j^+} + 2\hat{\sigma}_{\tilde{d}_j^+})) - \mathbb{E}(\frac{w_{d_i^-}}{N_{\tilde{d}_j^+}}(\hat{\mu}_{d_i^-} + 2\hat{\sigma}_{d_i^-})))
\end{aligned}$$

From step S_3 to step S_4, we apply the rule that if $a \geq 0, b \geq 0, c \geq 0$ and $b \geq c$, then $\sqrt{a+b} - \sqrt{a+c} \leq \sqrt{b} - \sqrt{c}$. For simplicity of presentation, we denote the normalization of $\frac{w_{\tilde{d}_j^+}}{N_{\tilde{d}_j^+}}$ by $w_{\tilde{d}_j^+}$, and similarly, the normalized $w_{d_i^-}$. Hence, we have

$$\Delta C^{+} \leq max\{\mathbb{E}(w_{\tilde{d}_j^+}(\hat{\mu}_{\tilde{d}_j^+} + 2\hat{\sigma}_{\tilde{d}_j^+})) - \mathbb{E}(w_{d_i^-}(\hat{\mu}_{d_i^-} + 2\hat{\sigma}_{d_i^-})), \mathbb{E}(w_{\tilde{d}_j^+}\hat{\mu}_{\tilde{d}_j^+}) - \mathbb{E}(w_{d_i^-}\hat{\mu}_{d_i^-})\},$$

where the $\hat{\mu}_*$ and $\hat{\sigma}_*$ denote the DNN output probability and its corresponding standard deviation respectively, w_* denotes the learned weight of DNN risk feature.