Transfer Neyman-Pearson Algorithm for Outlier Detection

Mohammadreza M. Kalan[1]
Univ Rennes, Ensai, CNRS,
CREST-UMR 9194,
F-35000 Rennes, France

Eitan J. Neugut
Columbia University

Samory Kpotufe
Columbia University

Abstract

We consider the problem of transfer learning in outlier detection where target abnormal data is rare. While transfer learning has been considered extensively in traditional classification, the problem of transfer in outlier detection and more generally in imbalanced classification settings has received less attention. We propose a general algorithmic approach which is shown theoretically to yield strong guarantees w.r.t. to a range of changes in abnormal distribution, and at the same time amenable to practical implementation. We then investigate different instantiations of this general algorithmic approach, e.g., based on multi-layer neural networks, and show empirically that they significantly outperform natural extensions of transfer methods from traditional classification (which are the only solutions available at the moment).

1 Introduction

Outlier detection problems are characterized by a significant imbalance between two classes of data: one class with an abundance of available samples, referred to as the *common class*, and another with very few or no samples, known as the *outlier* or *rare class*. This imbalance makes it challenging to design accurate decision rules, as the scarcity of data from the rare class hinders the learning process. Examples of applications in this imbalanced setting include detecting rare events in climate science, such as heavy precipitation (Folino et al., 2023; Mazzoglio et al., 2019; Frame et al., 2022), as well as disease diagnosis (Bourzac, 2014; Myszczynska et al., 2020) and malware detection in cybersecurity (Alamro et al., 2023; Kumar & Lim, 2019). A proven useful way to address these data limitations is to leverage another related data, referred to as the *source*, which might contain information about the *target* rare class. For instance, in the context of heavy precipitation detection, such related data could come from another location with sufficient recorded samples. This scenario represents a transfer learning problem. However, much of the existing literature has focused on transfer learning in traditional balanced classification tasks (Pan & Yang, 2009; Zhuang et al., 2020), rather than on outlier detection and imbalanced classification, where there is an asymmetry in the errors across different classes due to their varying relative importance.

In this work, we propose a general meta-algorithm for outlier detection that effectively leverages source task data with sufficient outlier class samples alongside target data. The approach is supported by a theoretical guarantee on the target generalization error, without making any assumptions about the underlying data distribution, while being amenable to practical implementation. Additionally, the proposed meta-algorithm is adaptive, as it exploits source data when it is related to the target and avoids negative transfer when the source is unrelated. Another key feature of the meta-algorithm is its model-free property, enabling it to be applied across a variety of models, such as neural networks, kernel machines, and others. Consequently, this general approach can be integrated with existing methods that use specific models to find a shared repre-

[1]This work was done while the author was at Columbia University.

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

sentation of feature spaces for the source and target.

To provide a theoretical justification for the performance of the proposed approach, we adopt the transfer Neyman-Pearson framework introduced by Kalan & Kpotufe (2024) to derive generalization error bounds. In the Neyman-Pearson classification problem, the goal is to achieve low classification error on the rare class while ensuring that the error w.r.t. the common class remains below a prespecified threshold. Kalan & Kpotufe (2024) introduced a transfer Neyman-Pearson framework based on 0-1 loss risk and derived a minimax rate for the problem. In this work, we first extend the transfer Neyman-Pearson framework (Kalan & Kpotufe, 2024) to the case with a surrogate loss function. And then we propose a meta-algorithm as a constrained optimization procedure leveraging source samples along with target samples in outlier detection. Subsequently, we derive a bound on the target generalization error of the solution obtained through the proposed optimization procedure, capturing the extent of information transferable from the source to the target. Furthermore, the bound guarantees that when the source is unrelated to the target, the procedure effectively disregards the source and avoids negative transfer. It is expressed in terms of the number of source and target samples, the Rademacher complexity of the hypothesis class, and a natural extension of the transfer exponent (Hanneke & Kpotufe, 2019), which quantifies the relative effect of the source on the target.

We then propose a transfer learning algorithm to implement an instantiation of the proposed theoretically sound optimization procedure. As detailed in Section 5, the process begins with constructing the Lagrangian using some tuning parameters. By minimizing the cost over a grid of parameter values, we obtain a function that minimizes the cost for each tuning parameter. Collecting these functions results in a reduced hypothesis class. Subsequently, a function is selected from this reduced class that minimizes the objective function of the proposed optimization procedure while satisfying its constraints. The challenge in transfer learning lies in determining the appropriate bias between the source and target data. Simply optimizing over the target sample may under-utilize valuable information from the source, while solely optimizing over the source samples risks negative transfer if the source distribution diverges significantly from the target. The proposed algorithm addresses this challenge by leveraging the source when it is informative and avoiding negative transfer when the source is unrelated to the target.

We evaluate the proposed algorithm on both real and synthetic datasets. For heavy rainfall prediction, treated as outliers, we use climate data (Yu et al., 2024; NASA POWER, 2024), and for default prediction, we use financial data (Gregory, 2018). Our results demonstrate that when the source contains useful information about the target, the algorithm's performance improves compared to using only target data. Conversely, when the source is unrelated, there is no negative transfer effect. In other words, the algorithm adapts to the data and does not require prior knowledge of the relatedness between the source and target. For comparison with other methods, and given the absence of any implementable algorithm for the transfer Neyman-Pearson problem, we propose a practical adaptation of the procedure introduced in Kalan & Kpotufe (2024). Furthermore, we extend existing baselines—which adjust the scoring function's threshold to satisfy a pre-specified Type-I error rate (Uyar et al., 2010; Abd Elrahman & Abraham, 2013; Saito & Rehmsmeier, 2015)—to the context of transfer learning for outlier detection. Our results demonstrate that the proposed approach consistently achieves superior performance compared to these alternatives.

2 Related Work

Unlike outlier detection and imbalanced classification, transfer learning has been extensively studied in traditional balanced classification, leading to various approaches, algorithms, and generalization bounds. Seminal works Blitzer et al. (2007); Mansour et al. (2009); Ben-David et al. (2010a,b) and recent studies (Zhao et al., 2019; Hanneke & Kpotufe, 2019; Cai & Wei, 2021) explore knowledge transfer between source and target domains. Notably, Hanneke & Kpotufe (2019) introduces the transfer exponent to quantify domain distance in classification. We adapt this notion to provide theoretical justification for our proposed transfer learning algorithm.

Outlier detection methods fall into semi-supervised and supervised categories. In the semi-supervised setting, where only normal-class samples are available, density level set estimation is widely used (Steinwart et al., 2005; Polonik, 1995; Tsybakov, 1997). Some works (Abe et al., 2006; Chalapathy et al., 2018; Yang et al., 2023) transform outlier detection into classification by generating artificial outlier samples.

In supervised outlier detection setting, which is also the focus of our work, most algorithms train a

scoring function and produce a Receiver Operating Characteristic (ROC) curve by evaluating different thresholds on the scoring function to adjust the type-I error (Uyar et al., 2010; Saito & Rehmsmeier, 2015; Tong et al., 2018). In contrast, our procedure minimizes the Type-II error for a pre-specified threshold on the Type-I error by effectively leveraging both source and target samples, as detailed in Section 4. Experiments demonstrate that our methods consistently utilize source information when it is relevant and effectively avoid negative transfer when the source is uninformative, without requiring any prior knowledge of the relatedness. This contrasts with other methods, which may perform well in certain scenarios but lack consistent reliability.

More closely related, Kalan & Kpotufe (2024) studies transfer learning in Neyman-Pearson outlier detection, highlighting its fundamental differences from balanced classification. While Kalan & Kpotufe (2024) characterizes minimax transfer rates and proposes an adaptive procedure, it lacks an implementable algorithm. We build on this by introducing a meta-algorithm with theoretical guarantees and a practical transfer learning algorithm for outlier detection. Comparisons show our method outperforms an approach inspired by Kalan & Kpotufe (2024).

3 Setup

We begin by setting up the Neyman-Pearson classification framework which formalizes outlier detection and then extend it to the transfer learning setting.

3.1 Neyman-Pearson Classification

Let μ_0 and μ_1 represent probability distributions on a measurable space $(\mathcal{X}, \Sigma)$. Additionally, let $\mathcal{H}$ be a hypothesis class consisting of functions $h : \mathcal{X} \to \mathbb{R}$. For a function $h \in \mathcal{H}$, we predict that data $x \in \mathcal{X}$ is generated by μ_1 if $h(x) \geq 0$, and by μ_0 if $h(x) < 0$. In this paper, we study the setting where there is an abundance of data available from μ_0 and only a few or no data from μ_1. Therefore, we refer to the classes generated by μ_0 and μ_1 as the *common* class and the *rare* (or *outlier*) class, respectively.

Definition 1. *Type-I and Type-II errors are defined as* $R_{\mu_0}(h) = \mathbb{E}_{\mu_0}[\mathbb{1}\{h(X) \geq 0\}]$ *and* $R_{\mu_1}(h) = \mathbb{E}_{\mu_1}[\mathbb{1}\{h(X) < 0\}]$*, respectively, where* $\mathbb{1}$ *denotes the indicator function.*

Neyman-Pearson classification aims to minimize the Type-II error while keeping the Type-I error below a pre-specified threshold α:

$$\begin{aligned} &\underset{h \in \mathcal{H}}{\text{Minimize}}\ R_{\mu_1}(h) \\ &\text{s.t. } R_{\mu_0}(h) \leq \alpha \end{aligned} \tag{1}$$

The Neyman-Pearson Lemma (Lehmann & Lehmann, 1986), under some mild assumptions, characterizes the universally optimal solution of (1)—when $\mathcal{H}$ consists of all measurable functions from $\mathcal{X}$ to $\mathbb{R}$—as $h^*_\alpha(x) = 2\mathbb{1}\left\{\frac{p_1}{p_0}(x) \geq \lambda\right\} - 1$, provided there exists a λ such that $R_{\mu_0}(h^*_\alpha) = \alpha$.

In practical settings, surrogate loss functions are preferred over the indicator loss function because the latter is discontinuous and leads to intractable combinatorial optimization problems. Additionally, surrogate loss functions not only penalize misclassified points but also take into account their distance from the decision boundary, resulting in more robust classifiers (Bao et al., 2020). In this section, we aim to establish the foundation for an implementable transfer learning algorithm for outlier detection. To achieve this, we need to replace the 0-1 loss with a surrogate loss.

Definition 2. *A function* $\varphi : \mathbb{R} \to \mathbb{R}^+$ *is called an L-Lipschitz surrogate loss if it is non-decreasing,* $\varphi(0) = 1$*, satisfies* $|\varphi(x) - \varphi(y)| \leq L|x - y|$ *for all* $x, y \in \mathbb{R}$*, and there exists a constant* $C > 0$ *such that for all* $h \in \mathcal{H}$ *and* $x \in \mathcal{X}$ *we have* $\max\{\varphi(h(x)), \varphi(-h(x))\} \leq C$.

In the following definition, we introduce Type-I and Type-II errors with respect to a surrogate loss.

Definition 3. *φ-Type-I and φ-Type-II errors are defined as* $R_{\varphi,\mu_0}(h) = \mathbb{E}_{\mu_0}[\varphi(h(X))]$ *and* $R_{\varphi,\mu_1}(h) = \mathbb{E}_{\mu_1}[\varphi(-h(X))]$

Next, we define the Rademacher complexity of a hypothesis class $\mathcal{H}$, which serves as a measure of the class's capacity and controls its complexity.

Definition 4 (Rademacher Complexity (Bartlett & Mendelson, 2002))**.** *Let* $X_1, ..., X_n$ *be i.i.d. samples drawn from a distribution* μ *on* $\mathcal{X}$*. Define the random variable*

$$\hat{R}_n(\mathcal{H}) = \mathop{\mathbb{E}}_{\sigma}\left[\sup_{h \in \mathcal{H}} \left|\frac{1}{n}\sum_{i=1}^{n} \sigma_i h(X_i)\right|\right],$$

where $\sigma_1, ..., \sigma_n$ *are independent uniform* $\{\pm 1\}$*-valued random variables. The Rademacher complexity of* $\mathcal{H}$ *is then defined as* $R_n(\mathcal{H}) = \mathbb{E}\hat{R}_n(\mathcal{H})$ *where the expectation is taken w.r.t. the i.i.d. samples.*

Assumption 1. *We assume that* $R_n(\mathcal{H}) \leq \frac{B_{\mathcal{H}}}{\sqrt{n}}$ *for some* $B_{\mathcal{H}}$ *which characterizes the complexity of* $\mathcal{H}$.

Remark 1. *If the input features are bounded, most practical hypothesis classes, such as linear regression and neural networks, satisfy Assumption 1, provided that the coefficients and weights are bounded (Golowich et al., 2018).*

Neyman-Pearson classification with a surrogate loss φ is then formulated as follows:

$$\begin{aligned} &\underset{h\in\mathcal{H}}{\text{Minimize}}\ R_{\varphi,\mu_1}(h) \\ &\text{s.t. } R_{\varphi,\mu_0}(h) \le \alpha \end{aligned} \tag{2}$$

3.2 Transfer Learning Setup

Let $\mu_{1,S}$ and $\mu_{1,T}$ denote the distributions of the rare class for the source and target, respectively. We consider the following source and target Neyman-Pearson classification problems with a common distribution μ_0 and surrogate loss φ:

$$\begin{aligned} &\underset{h\in\mathcal{H}}{\text{Minimize}}\ R_{\varphi,\mu_{1,S}}(h) \\ &\text{s.t. } R_{\varphi,\mu_0}(h) \le \alpha \end{aligned} \tag{3}$$

$$\begin{aligned} &\underset{h\in\mathcal{H}}{\text{Minimize}}\ R_{\varphi,\mu_{1,T}}(h) \\ &\text{s.t. } R_{\varphi,\mu_0}(h) \le \alpha \end{aligned} \tag{4}$$

We denote (not necessarily unique) solutions to (3) and (4) by $h^*_{S,\alpha}$ and $h^*_{T,\alpha}$, respectively.

In practical scenarios, the underlying distributions μ_0, $\mu_{1,S}$, and $\mu_{1,T}$ are unknown and only accessible through samples. We consider a setting where there are n_0, n_S, and n_T i.i.d. samples available from μ_0, $\mu_{1,S}$, and $\mu_{1,T}$, respectively. The learner then aims to return a hypothesis $\hat{h} \in \mathcal{H}$ that minimizes the **target excess error**

$$\mathcal{E}_{1,T}(\hat{h}) := \max\left\{0, R_{\varphi,\mu_{1,T}}(\hat{h}) - R_{\varphi,\mu_{1,T}}(h^*_{T,\alpha})\right\} \tag{5}$$

subject to the constraint that $R_{\varphi,\mu_0}(\hat{h}) \le \alpha + \epsilon_0$, where a slack $\epsilon_0 = \epsilon_0(n_0)$, typically of order $n_0^{-1/2}$, is allowed to deviate from the pre-specified threshold.

Next, we adapt the notion of transfer exponent—used in traditional classification (Hanneke & Kpotufe, 2019) and 0-1 loss Neyman-Pearson classification (Kalan & Kpotufe, 2024)—to capture the transfer distance between source and target in the setting of Neyman-Pearson classification with a surrogate loss.

Definition 5 (Transfer Exponent)**.** *Let $S^*_\alpha \subset \mathcal{H}$ denote the set of solutions of source problem* (3)*. We call $\rho(r) > 0$ a transfer exponent from source* (3) *to target* (4) *under $\mathcal{H}$ if there exist $r, c_{\rho(r)} > 0$ such that*

$$\begin{aligned} c_{\rho(r)} \cdot \max&\left\{0, R_{\varphi,\mu_{1,S}}(h) - R_{\varphi,\mu_{1,S}}(h^*_{S,\alpha})\right\} \\ &\ge \max\left\{0, R_{\varphi,\mu_{1,T}}(h) - R_{\varphi,\mu_{1,T}}(h^*_{S,\alpha})\right\}^{\rho(r)} \end{aligned} \tag{6}$$

for all $h \in \mathcal{H}$ with $R_{\varphi,\mu_0}(h) \le \alpha + r$, where $h^*_{S,\alpha} = \underset{h\in S^*_\alpha}{\arg\max}\ R_{\varphi,\mu_{1,T}}(h)$.

The transfer exponent reflects how well a function's performance in the source translates to its performance in target—and thus serves as a measure of how informative the source is about the target. The source is most informative when ρ is small and close to 1, and less informative when ρ is large.

4 Main Theoretical Results

In this section, we propose a transfer learning optimization procedure for Neyman-Pearson classification that aims to find a function minimizing target excess error (5), subject to the φ-Type-I constraint, by leveraging both source and target data. We then analyze this approach by providing upper bounds on the generalization error of the procedure's solution.

4.1 Transfer Learning Optimization Procedure

First, we need to define the empirical counterparts of the surrogate losses as follows:

$$\hat{R}_{\varphi,\mu_0}(h) = \frac{1}{n_0} \sum_{X_i\sim\mu_0} \varphi(h(X_i))$$

$$\hat{R}_{\varphi,\mu_{1,T}}(h) = \frac{1}{n_T} \sum_{X_i\sim\mu_{1,T}} \varphi(-h(X_i))$$

$$\hat{R}_{\varphi,\mu_{1,S}}(h) = \frac{1}{n_S} \sum_{X_i\sim\mu_{1,S}} \varphi(-h(X_i))$$

The following proposition provides a concentration result for empirical errors in hypothesis classes with bounded Rademacher complexities.

Proposition 1. *Let $\delta > 0$ and $\mathcal{H}$ be a hypothesis class satisfying Assumption 1. Furthermore, suppose that $\hat{R}_{\varphi,\mu}$ denotes empirical error with respect to n i.i.d. samples drawn from a distribution μ, which could be either μ_0 or μ_1. Then, with probability at least $1-\delta$, we have*

$$\sup_{h\in\mathcal{H}} |R_{\varphi,\mu}(h) - \hat{R}_{\varphi,\mu}(h)| \le \frac{4B_{\mathcal{H}}L + C\sqrt{2\log(2/\delta)}}{\sqrt{n}},$$

where C is defined in Definition 2.

Next, we define $\hat{h}_{T,\alpha+\epsilon_0/2}$ as follows:

$$\hat{h}_{T,\alpha+\epsilon_0/2} = \arg\min_{h\in\mathcal{H}} \hat{R}_{\varphi,\mu_{1,T}}(h) \quad \text{s.t. } \hat{R}_{\varphi,\mu_0}(h) \le \alpha + \epsilon_0/2 \tag{7}$$

Let $\tilde{C} = 8B_{\mathcal{H}}L + 2C\sqrt{2\log(2/\delta)}$. We then propose the following optimization procedure to solve problem (4) by utilizing both source and target samples as follows::

$$\hat{h} = \arg\min_{h\in\mathcal{H}} \hat{R}_{\varphi,\mu_{1,S}}(h) \quad \text{s.t. } \hat{R}_{\varphi,\mu_{1,T}}(h) \le \hat{R}_{\varphi,\mu_{1,T}}(\hat{h}_{T,\alpha+\epsilon_0/2}) + \frac{2\tilde{C}}{\sqrt{n_T}} \quad \hat{R}_{\varphi,\mu_0}(h) \le \alpha + \epsilon_0/2 \tag{8}$$

4.2 Upper bounds on the Generalization Errors

The following theorem provides upper bounds on the target excess risk in terms of the number of available samples from the source and target, as well as the transfer exponent, which captures the distance between the source and target.

Theorem 1. *Let $\delta > 0$ and $\epsilon_0 = \frac{\tilde{C}}{\sqrt{n_0}}$, where $\tilde{C} = 8B_{\mathcal{H}}L + 2C\sqrt{2\log(2/\delta)}$. Moreover, let $\hat{h}$ be the hypothesis returned by the procedure* (8)*, and let the transfer exponent be $\rho(r)$ with coefficient $c_{\rho(r)}$ for $r \ge \epsilon_0$. Then, with probability at least $1-3\delta$, the hypothesis $\hat{h}$ satisfies*

$$\mathcal{E}_{1,T}(\hat{h}) \le \min\left\{c_{\rho(r)} \cdot \left(\frac{\tilde{C}}{\sqrt{n_S}}\right)^{1/\rho(r)} + 4\cdot\Delta, \frac{4\tilde{C}}{\sqrt{n_T}}\right\}$$

$$R_{\mu_0}(\hat{h}) \le R_{\varphi,\mu_0}(\hat{h}) \le \alpha + \epsilon_0.$$

where $\Delta = R_{\varphi,\mu_{1,T}}(h^*_{S,\alpha}) - R_{\varphi,\mu_{1,T}}(h^*_{T,\alpha+\epsilon_0})$. Here, $h^*_{T,\alpha+\epsilon_0}$ is the solution to problem (4) *with the threshold on the φ-Type-I error set to $\alpha+\epsilon_0$ instead of α.*

Remark 2. *$\rho(r)$ captures the relative effectiveness of the source samples in the target domain. The lower the value of $\rho(r)$, the more effective the source samples are. Moreover, source samples are useful only up to a certain accuracy, captured by Δ. To reduce the error further, it becomes necessary to leverage target samples.*

Kalan & Kpotufe (2024) derives a similar bound to Theorem 1 for the problem of 0-1 loss Neyman-Pearson classification, under the assumption of a finite VC class, except for the term Δ, which is defined there as $R_{\mu_{1,T}}(h^*_{S,\alpha}) - R_{\mu_{1,T}}(h^*_{T,\alpha})$, leading to a sharper bound. Next, we make additional assumptions about the hypothesis class $\mathcal{H}$ and the surrogate loss function φ to tighten the bound in Theorem 1.

Assumption 2. *We assume that $\mathcal{H}$ is a convex class, meaning that for any $\theta \in (0,1)$ and any two hypotheses $h_1, h_2 \in \mathcal{H}$, we have $\theta\cdot h_1 + (1-\theta)\cdot h_2 \in \mathcal{H}$.*

Note that classes such as polynomial regression functions and majority votes over a basis of functions are examples that satisfy Assumption 2. However, a class of neural networks with a fixed architecture is generally not closed under convex combinations. Since the Rademacher complexity of the convex hull of a class is equal to that of the class itself, we can instead consider the convex hull of a neural network class, which is convex.

Theorem 2. *Assume the setting of Theorem 1. Moreover, suppose that $\mathcal{H}$ satisfies Assumption 2 and that φ is convex. Furthermore, suppose that the set $\{h \in \mathcal{H} : R_{\varphi,\mu_0}(h) \le \alpha/2\}$ is nonempty. Then, with probability at least $1-3\delta$, the hypothesis $\hat{h}$ returned by the procedure* (8) *satisfies*

$$\mathcal{E}_{1,T}(\hat{h}) \le \min\left\{c_{\rho(r)} \cdot \left(\frac{\tilde{C}}{\sqrt{n_S}}\right)^{1/\rho(r)} + \frac{C'}{\sqrt{n_0}} + 4\cdot\tilde{\Delta}, \frac{4\tilde{C}}{\sqrt{n_T}}\right\}$$

$$R_{\mu_0}(\hat{h}) \le R_{\varphi,\mu_0}(\hat{h}) \le \alpha + \epsilon_0.$$

where $\tilde{\Delta} = R_{\varphi,\mu_{1,T}}(h^*_{S,\alpha}) - R_{\varphi,\mu_{1,T}}(h^*_{T,\alpha})$ and $C' = \frac{8C\tilde{C}}{\alpha}$.

The term $\frac{C'}{\sqrt{n_0}}$ is negligible because n_0 denotes the number of samples drawn from the common distribution μ_0, from which many samples are available. Therefore, Theorem 2 provides a sharper bound than Theorem 1.

5 Transfer Learning Algorithm for Outlier Detection

In this section, we propose a transfer learning Neyman-Pearson (TLNP) algorithm for outlier detection based on the optimization procedure (8). We evaluate its performance using climate data (Yu et al., 2024; NASA POWER, 2024), financial data (Gregory, 2018), and synthetically generated datasets. Additionally, we compare its performance with an algorithm inspired by the procedure proposed in Kalan & Kpotufe (2024), as well as other approaches. We demonstrate that the proposed algorithm consistently avoids negative transfer when

the source is uninformative about the target and effectively leverages an informative source when it is, whereas other approaches may occasionally perform well in specific cases but fail to maintain consistency across different datasets.

The main idea of TLNP algorithm is as follows. First, we consider the Lagrangian associated with (8) and consider the following cost function, with tuning parameters λ_S, λ_0:

$$\hat{R}_{\varphi,\mu_{1,T}}(h) + \lambda_S \hat{R}_{\varphi,\mu_{1,S}}(h) + \lambda_0 \hat{R}_{\varphi,\mu_0}(h) \quad (9)$$

Next, over a grid search of (λ_S, λ_0), we identify functions within the hypothesis class minimizing the cost function (9), thereby obtaining a smaller, filtered hypothesis class. Finally, we solve the 0-1 loss counterpart of (8) within this reduced hypothesis class. Here, we provide a detailed explanation of the TLNP process through the following steps. In the following, ϵ_0 is proportional to $\frac{1}{\sqrt{n_0}}$, where n_0 is the number of training data points in the normal class. ϵ_0 is a parameter that the user can select; if the user is conservative regarding Type-I α constraint, it should be chosen to be sufficiently small.

Step 1) searching over a grid of (λ_S, λ_0) pairs: The TLNP algorithm sets λ_S to a fixed point and, for each λ_S, we start with λ_0 of 1 and fine-tune λ_0 until 0-1 loss Type-I error of h belongs to the interval $[\alpha - \epsilon_0/2, \alpha + \epsilon_0/2]$. Since the elements of $\mathcal{H}$ are real-valued functions, we apply the sign function to set binary classifiers and calculate $\hat{R}_{\mu_0}(\text{sign}(h))$.

We start with λ_S fixed to one of 12 points, $(0, 0.05, 0.1, 0.5, 1, 5, 10, 20, 40, 60, 80, 100)$. For each point (λ_S, λ_0), we train a new function $h \in \mathcal{H}$. The fine-tuning process works by comparing the Type-I error to the $\alpha \pm \epsilon_0/2$ range. If the Type I error is too high (overshoot), the algorithm increases λ_0 by multiplying it by $(1 + \text{increment factor})$. If the error is too low (undershoot), it decreases λ_0 by multiplying it by $(1 - \text{increment factor})$. The initial increment factor is 0.5. Each time the error flips between overshooting and undershooting, the increment factor is halved, allowing for finer adjustments. Once the Type-I error falls within the range $\alpha \pm \epsilon_0/2$, then we move onto the next λ_S in the list.

If fewer than 5 successful tunings have been achieved, the search range is expanded by adding additional λ_S values. The process stops when 12 points successfully converge with an acceptable Type-I error, or when the values of λ_S become unreasonably small or large. At the end, we obtain a reduced set of hypothesis class $\hat{\mathcal{H}}$ whose elements satisfy Type-I error constraint.

Step 2) Filtering $\hat{\mathcal{H}}$ using the target abnormal data: We first evaluate $\hat{R}_{\mu_{1,T}}$, which represents the target 0-1 loss Type-II error with respect to the target abnormal training data, for the elements of $\hat{\mathcal{H}}$ obtained in the first step. Let $\hat{h}_T \in \hat{\mathcal{H}}$ be the function that yields the lowest $\hat{R}_{\mu_{1,T}}$, i.e., $\hat{R}_{\mu_{1,T}}(\text{sign}(\hat{h}_T)) = \min_{h \in \hat{\mathcal{H}}} \hat{R}_{\mu_{1,T}}(\text{sign}(h))$. Then, inspired by the constraint in the optimization procedure (8), we identify the functions that are close to $\hat{h}_T$ in terms of target Type-II error. We use a universal constant $c = 0.5$ and define $\hat{\mathcal{H}}_T$ as the set of functions $h \in \hat{\mathcal{H}}$ satisfying the inequality:

$$\hat{R}_{\mu_{1,T}}(\text{sign}(h)) \leq \hat{R}_{\mu_{1,T}}(\text{sign}(\hat{h}_T)) + \frac{c}{\sqrt{n_T}} \quad (10)$$

We demonstrate that this universal constant performs well across all datasets, both real-world and synthetic. Moreover, if users have prior knowledge about the relatedness of the source and target, they can adjust this constant accordingly by either decreasing or increasing it. Furthermore, since the constant serves primarily to upper-bound the variance of errors for a given dataset, we propose a method in Appendix E to estimate this variance and use that instead of the constant.

Step 3) Filtering $\hat{\mathcal{H}}_T$ using the source abnormal data: In this step, we evaluate $\hat{R}_{\mu_{1,S}}$, which represents the source 0-1 loss Type-II error with respect to the source abnormal data, for the elements of $\hat{\mathcal{H}}_T$ obtained in the second step. We then select the function that yields the lowest error as the output of the algorithm. Roughly speaking, in this step, if the source is informative, the algorithm leverages it by minimizing the source error. Conversely, if the source is not informative, all functions in $\hat{\mathcal{H}}_T$ can achieve the rate $\frac{1}{\sqrt{n_T}}$ on the target data, thereby avoiding negative transfer.

We also compare TLNP with other approaches, including a procedure inspired by Kalan & Kpotufe (2024), as detailed below.

1) Transfer learning outlier detection (Kalan & Kpotufe, 2024): While Kalan & Kpotufe (2024) did not propose an implementable algorithm, we draw inspiration from the proposed procedure and implement it as follows. We obtain the solutions to (3) and (4) using a Lagrangian approach and then select the best of two based on evaluation with the target abnormal data. In this approach, the source and target data are handled separately rather than being combined.

2) Only target Neyman-Pearson: This approach is similar to TLNP, except that the source

data is not utilized. In other words, we set $\lambda_S = n_S = 0$, thereby eliminating step 3 of the TLNP process. The final output is selected in step 2 by minimizing the target abnormal data. Consequently, this approach serves as a baseline for assessing the benefit of leveraging source data.

3) Only source Neyman-Pearson: This approach is similar to the only target Neyman-Pearson approach, except that the target data is replaced with source data.

4) Pooled source and target Neyman-Pearson: This approach follows the idea of the only target Neyman-Pearson approach, but it pools both source and target data instead of just using target data. In other words, it does not distinguish between the two, treating the source data as if it were the target.

5) Only target thresholding traditional classification: This approach disregards the source data and finds a classifier using a scoring function to classify normal and abnormal data, the same as in traditional balanced classification. It then adjusts the threshold on the scoring function to satisfy the Type-I error constraint.

6) Pooled source and target thresholding traditional classification: This approach is similar to only target thresholding approach, except it pools both source and target data, instead of just using target data, without distinguishing between them.

6 Experiments and Numerical Results

In this section, we evaluate the proposed algorithm on climate data (Yu et al., 2024; NASA POWER, 2024), financial data (Gregory, 2018), and synthetically generated datasets for outlier detection. We analyze various source-target pairs to assess the algorithm's adaptability. When the source is relevant to the target, the algorithm effectively leverages this information. Conversely, if the source is not relevant, it avoids negative transfer, unlike other approaches that often fail to perform consistently and may suffer from negative transfer. Additionally, we implement two instantiations of our algorithm using multi-layer perceptron and quadratic models. Furthermore, in all the experiments, n_T refers to the target abnormal data, and n_S refers to the source abnormal data. For the normal class, we use only the data from the target domain.

6.1 Climate Data (Climsim) Experiments (Yu et al., 2024)

We implement our algorithm, along with the approaches described in Section 5, on the ClimSim dataset (Yu et al., 2024) to detect heavy rain versus non-heavy rain.

Sample Dataset: In the original dataset, each data point consists of 124 numerical features, such as temperature, specific humidity, and surface sensible heat flux, among others, along with an output of size 128, including variables like rain rate and snow rate. From the output variables, we only extract the rain rate and apply the 95th percentile criterion (Saidi et al., 2015; Schär et al., 2016) to categorize the data into binary heavy and non-heavy rain classes. The dataset includes various locations specified by longitude and latitude, which we merge into neighboring clusters. For details on clustering the locations, refer to the Appendix D. We select specific cluster pairs as source and target pairs. In one experiment, we fix the number of target heavy rain samples at $n_T = 50$ and increase the number of source heavy rain samples up to 2,500. In another experiment, we fix the number of source heavy rain samples at $n_S = 2,500$ and vary n_T from 25 to 250. In all cases, there are 4,000 training points from the target non-heavy rain class (also referred to as the normal class), along with approximately 2,000 test data points for target heavy rain and 4,000 test data points for target non-heavy rain.

Training: We use a 2-layer fully connected neural network with ReLU activation functions and 62 units in the hidden layer. Additionally, we employ exponential loss as the surrogate loss function and use the Adam optimizer for training. The results are averaged over 10 runs for each experiment.

Results: In Figures 1 and 2, we examine two scenarios: in the first, we select cluster 26 as the target and cluster 27 as the source; in the second, cluster 38 is the target, and clusters 37 and 39 grouped together constitute the source. In these experiments, the Type-I error threshold is set at $\alpha = 0.05$, $\epsilon_0 = 0.01$, and the Type-II error on the target test data is plotted. Figures 1 and 2 demonstrate that TLNP effectively combines source and target data to reduce the Type-II error compared to the 'only target' approach, which serves as the baseline. This gain over the baseline, in the case of pairs 26 and 27, is more evident when n_S is sufficiently large. Furthermore, while the "only source NP" and "pooled source and target NP" methods perform relatively well in Figure 2 when n_S is sufficiently large, they suffer from

negative transfer in Figure 1. A similar pattern is observed with the pooled source and target thresholding method. Although it performs relatively well in Figure 1, its performance is inconsistent, as shown in Figure 2.

Moreover, the results indicate that effectively combining source and target data can outperform even the best of the "only source" and "only target" approaches, including the procedure proposed in Kalan & Kpotufe (2024). Additionally, Figures 1 and 2 show that when n_T is fixed at 50 and n_S increases, the performance of TLNP saturates quickly. This special situation aligns with the scenario described in Remark 2, where the usefulness of source data quickly saturates, for instance, because the best source predictors differ significantly from the best target predictors (i.e., they have a large error Δ under the target).

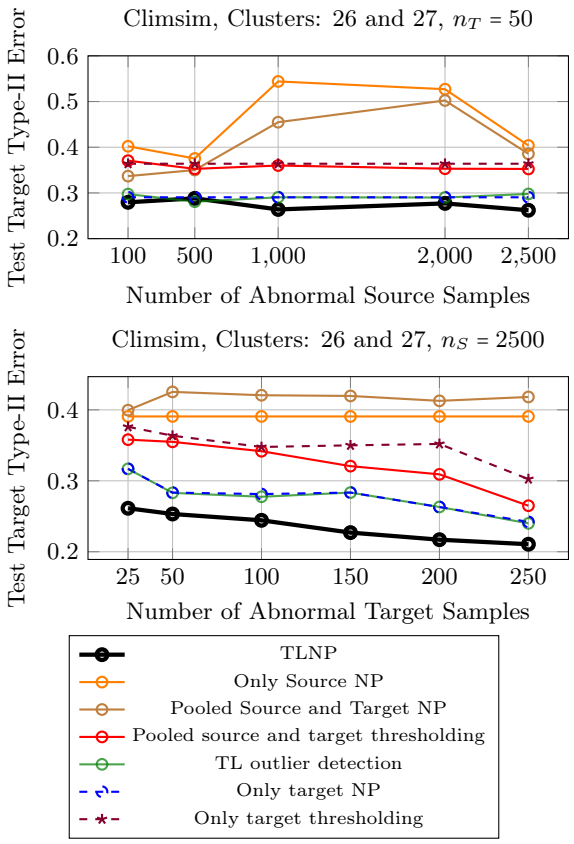


Figure 1: The performance of our algorithm (TLNP), along with other approaches on the Climate data (Yu et al., 2024), is evaluated for a Type-I error rate of α = 0.05. In this experiment, one scenario fixes n_T = 50 while increasing n_S. In the other scenario, n_S is fixed at 2500, and n_T is varied. In both cases, the target non-heavy rain class contains 4000 training samples.

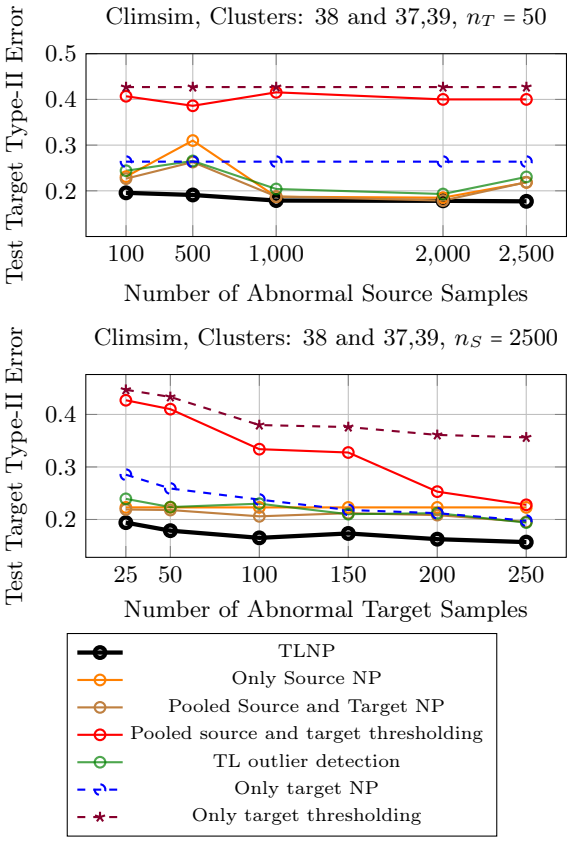


Figure 2: The performance of our algorithm (TLNP), along with other approaches on the Climate data (Yu et al., 2024), is evaluated for a Type-I error rate of α = 0.05. In this experiment, one scenario fixes n_T = 50 while increasing n_S. In the other scenario, n_S is fixed at 2500, and n_T is varied. In both cases, the target non-heavy rain class contains 4000 training samples.

6.2 NASA Climate Data Experiments (NASA POWER, 2024)

We use the NASA dataset (NASA POWER, 2024) for heavy rain detection, with target and source locations in the U.S. and Africa.

Sample Dataset: Each data point consists of six numerical features, and the 90th percentile criterion Kim et al. (2023) is applied to classify the data into binary categories: heavy rain and non-heavy rain. In this experiment, n_T = 50 is fixed and n_S is increased from 100 to 2500. The other settings are the same as in Section 6.1

Training: We utilize a two-layer fully connected neural network with ReLU activation functions and 12 units in the hidden layer. The exponential loss function is employed as a surrogate loss, and training is conducted using the Adam optimizer. Results are averaged over 10 runs for each experiment.

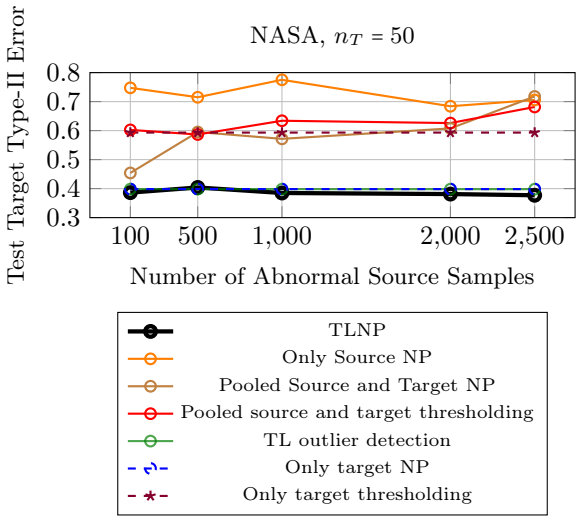


Figure 3: The performance of our algorithm (TLNP), along with other approaches on the Climate data (Yu et al., 2024), is evaluated for a Type-I error rate of $\alpha = 0.05$. In this experiment, $n_T = 50$ is fixed while n_S is increased. Moreover, the target non-heavy rain class contains 4000 training samples.

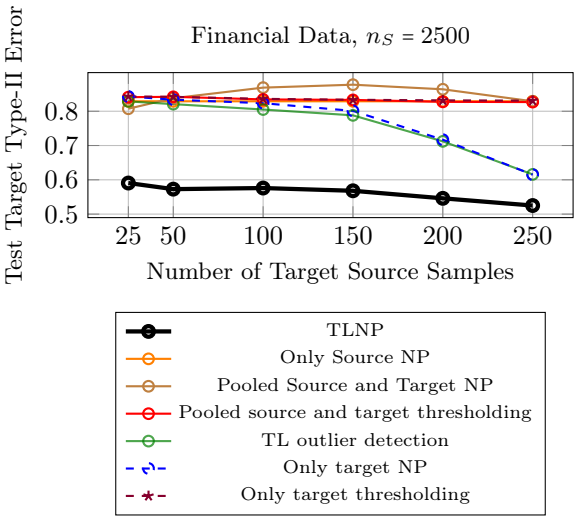


Figure 4: The performance of our algorithm (TLNP), along with other approaches on financial data (Gregory, 2018), is evaluated for a Type-I error rate of $\alpha = 0.1$. In this experiment, the number of source samples is fixed at $n_S = 2500$, and n_T is varied from 25 to 250. Moreover, the target normal class contains 4000 training samples.

Results: Since these locations are geographically distant (e.g., the U.S. and Africa), the source data is unlikely to relate to the target. Figure 3 shows that our algorithm effectively avoids negative transfer, achieving performance comparable to the only target baseline.

6.3 Financial Data Experiments (Gregory, 2018)

In this dataset, the goal is to predict whether a person will become financially delinquent within two years, meaning they fail to repay an installment that is 90 days or more past due.

Sample Dataset: We group data by age: individuals 36 and younger form the target group, while those 37 and older (with substantially more data) form the source group. The dataset includes nine features, such as personal credit balance, monthly income, debt-to-income ratio, and number of late payments. We fix $n_S = 2500$ and vary n_T from 25 to 250. Other settings remain as in previous sections.

Training: We use a two-layer fully connected neural network with ReLU activation functions and 9 units in the hidden layer. The exponential loss function is used as a surrogate loss, and training is performed with the Adam optimizer. Results are averaged over 10 runs for each experiment.

Results: Figure 4 illustrates that our algorithm (TLNP) outperforms other methods by efficiently integrating both source and target data. Furthermore, the results indicate that while naively using source data does not yield good performance on the target, effectively combining source and target data can lead to significant improvements.

6.4 Overall Performance Summary

Table 1 summarizes the Type-II errors of various approaches across all datasets for the case where $n_T = 50$ and $n_S = 2500$. It highlights that the Type-II error is consistently the minimum, indicating adaptability across datasets. In contrast, the performance of each baseline method varies significantly across datasets.

Appr.	Clim26	Clim38	NASA	Fin.
TLNP	**0.26**	**0.18**	**0.37**	**0.57**
(1)	0.3	0.23	0.4	0.82
(2)	0.29	0.26	0.4	0.83
(3)	0.40	0.22	0.7	0.83
(4)	0.39	0.22	0.72	0.84
(5)	0.36	0.43	0.6	0.84
(6)	0.35	0.4	0.68	0.84

Table 1: This table summarizes the Type-II errors of various approaches on different datasets for the case where $n_T = 50$ and $n_S = 2500$. See section 5 for the corresponding number of different approaches.

Acknowledgment

We acknowledge funding from NSF through the Learning the Earth with Artificial intelligence and Physics (LEAP) Science and Technology Center (STC) (Award #2019625).

References

Shaza M Abd Elrahman and Ajith Abraham. A review of class imbalance problem. *Journal of Network and Innovative Computing*, 1:9–9, 2013.

Naoki Abe, Bianca Zadrozny, and John Langford. Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 504–509, 2006.

Hayam Alamro, Wafa Mtouaa, Sumayh Aljameel, Ahmed S Salama, Manar Ahmed Hamza, and Aladdin Yahya Othman. Automated android malware detection using optimal ensemble learning approach for cybersecurity. *IEEE Access*, 2023.

Han Bao, Clay Scott, and Masashi Sugiyama. Calibrated surrogate losses for adversarially robust classification. In *Conference on Learning Theory*, pp. 408–451. PMLR, 2020.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010a.

Shai Ben-David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, pp. 129–136, 2010b.

John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. *Advances in neural information processing systems*, 20, 2007.

Katherine Bourzac. Diagnosis: early warning system. *Nature*, 513(7517):S4–S6, 2014.

T Tony Cai and Hongji Wei. Transfer learning for nonparametric classification: Minimax rate and adaptive classifier. 2021.

Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

Gianluigi Folino, Massimo Guarascio, and Francesco Chiaravalloti. Learning ensembles of deep neural networks for extreme rainfall event detection. *Neural Computing and Applications*, 35(14): 10347–10360, 2023.

Jonathan M Frame, Frederik Kratzert, Daniel Klotz, Martin Gauch, Guy Shalev, Oren Gilon, Logan M Qualls, Hoshin V Gupta, and Grey S Nearing. Deep learning rainfall–runoff predictions of extreme events. *Hydrology and Earth System Sciences*, 26(13):3377–3392, 2022.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pp. 297–299. PMLR, 2018.

Ian Gregory. Kaggle dataset - give me some credit, 2018. URL `https://github.com/DrIanGregory/Kaggle-GiveMeSomeCredit`.

Steve Hanneke and Samory Kpotufe. On the value of target data in transfer learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Mohammadreza Mousavi Kalan and Samory Kpotufe. Tight rates in supervised outlier transfer learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Jungho Kim, Jeremy Porter, and Edward J Kearns. Exposure of the us population to extreme precipitation risk has increased due to climate change. *Scientific reports*, 13(1):21782, 2023.

Vladimir Koltchinskii. Oracle inequalities in empirical risk minimization and sparse recovery problems, volume 2033 of lecture notes in mathematics, 2011.

Ayush Kumar and Teng Joon Lim. Edima: Early detection of iot malware network activity using machine learning techniques. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pp. 289–294. IEEE, 2019.

Erich Leo Lehmann and EL Lehmann. *Testing statistical hypotheses*, volume 2. Springer, 1986.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Paola Mazzoglio, Francesco Laio, Simone Balbo, Piero Boccardo, and Franca Disabato. Improving an extreme rainfall detection system with gpm imerg data. *Remote Sensing*, 11(6):677, 2019.

Monika A Myszczynska, Poojitha N Ojamies, Alix MB Lacoste, Daniel Neil, Amir Saffari, Richard Mead, Guillaume M Hautbergue,

Joanna D Holbrook, and Laura Ferraiuolo. Applications of machine learning to diagnosis and treatment of neurodegenerative diseases. *Nature reviews neurology*, 16(8):440–456, 2020.

NASA POWER. Nasa power: Prediction of worldwide energy resource api, 2024. URL `https://power.larc.nasa.gov/api/temporal/daily/point`.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Wolfgang Polonik. Measuring mass concentrations and estimating density contour clusters-an excess mass approach. *The annals of Statistics*, pp. 855–881, 1995.

Philippe Rigollet and Xin Tong. Neyman-pearson classification, convexity and stochastic constraints. *Journal of machine learning research*, 2011.

Helmi Saidi, Marzia Ciampittiello, Claudia Dresti, and Giorgio Ghiglieri. Assessment of trends in extreme precipitation events: a case study in piedmont (north-west italy). *Water Resources Management*, 29:63–80, 2015.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

Christoph Schär, Nikolina Ban, Erich M Fischer, Jan Rajczak, Jürg Schmidli, Christoph Frei, Filippo Giorgi, Thomas R Karl, Elizabeth J Kendon, Albert MG Klein Tank, et al. Percentile indices for assessing changes in heavy precipitation events. *Climatic Change*, 137:201–216, 2016.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Ingo Steinwart, Don Hush, and Clint Scovel. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6(2), 2005.

Xin Tong, Yang Feng, and Jingyi Jessica Li. Neyman-pearson classification algorithms and np receiver operating characteristics. *Science advances*, 4(2):eaao1659, 2018.

Alexandre B Tsybakov. On nonparametric estimation of density level sets. *The Annals of Statistics*, 25(3):948–969, 1997.

Asli Uyar, Ayse Bener, HN Ciracy, and Mustafa Bahceci. Handling the imbalance problem of ivf implantation prediction. *IAENG International Journal of Computer Science*, 37(2):164–170, 2010.

Ziyi Yang, Iman Soltani, and Eric Darve. Anomaly detection with domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2958–2967, 2023.

Sungduk Yu, Walter Hannah, Liran Peng, Jerry Lin, Mohamed Aziz Bhouri, Ritwik Gupta, Björn Lütjens, Justus C Will, Gunnar Behrens, Julius Busecke, et al. Climsim: A large multi-scale dataset for hybrid physics-ml climate emulation. *Advances in Neural Information Processing Systems*, 36, 2024.

Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J Gordon. On learning invariant representation for domain adaptation. *arXiv preprint arXiv:1901.09453*, 2019.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 2020.

Checklist

The checklist follows the references. For each question, choose your answer from the three possible options: Yes, No, Not Applicable. You are encouraged to include a justification to your answer, either by referencing the appropriate section of your paper or providing a brief inline description (1-2 sentences). Please do not modify the questions. Note that the Checklist section does not count towards the page limit. Not including the checklist in the first submission won't result in desk rejection, although in such case we will ask you to upload it during the author response period and include it in camera ready (if accepted).

In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]

(b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]

(c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:

(a) Statements of the full set of assumptions of all theoretical results. [Yes]

(b) Complete proofs of all theoretical results. [Yes]

(c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. [Yes]

(b) The license information of the assets, if applicable. [Not Applicable]

(c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Proof of Proposition 1

By [Koltchinskii (2011), Theorem 2.3.], we can get $R_n(\varphi \circ \mathcal{H}) \leq 2LR_n(\mathcal{H})$. Then, applying McDiarmid's inequality [See Shalev-Shwartz & Ben-David (2014), Chapter 26], we obtain

$$\begin{aligned}\sup_{h\in\mathcal{H}} \left| \mathbb{E}_\mu[\varphi(h(X))] - \frac{1}{n}\sum_{X_i\sim\mu} \varphi(h(X_i)) \right| &\leq 2R_n(\varphi\circ\mathcal{H}) + C\sqrt{\frac{2\log(2/\delta)}{n}} \\ &\leq \frac{4LB_{\mathcal{H}}}{\sqrt{n}} + C\sqrt{\frac{2\log(2/\delta)}{n}}\end{aligned}$$

with probability at least $1-\delta$. Furthermore, since $R_n(\mathcal{H}) = R_n(\mathcal{H}^-)$, where $\mathcal{H}^- = \{-h : h \in \mathcal{H}\}$, the same bound applies to the expression

$$\sup_{h\in\mathcal{H}} \left| \mathbb{E}_\mu[\varphi(-h(X))] - \frac{1}{n}\sum_{X_i\sim\mu} \varphi(-h(X_i)) \right|,$$

which concludes Proposition 1.

□

B Proof of Theorem 1

Consider the event where Proposition 1 holds for the distributions $\mu_0, \mu_{1,S}$, and $\mu_{1,T}$, which occurs with probability at least $1-3\delta$. We then divide the proof into three parts: 1)$R_{\varphi,\mu_0}(\hat{h}) \leq \alpha + \epsilon_0$, 2) $\mathcal{E}_{1,T}(\hat{h}) \leq \frac{4\tilde{C}}{\sqrt{n_T}}$, 3) $\mathcal{E}_{1,T}(\hat{h}) \leq c_{\rho(r)} \cdot (\frac{\tilde{C}}{\sqrt{n_S}})^{1/\rho(r)} + 4\cdot\Delta$

Part 1) $R_{\mu_0}(\hat{h}) \leq R_{\varphi,\mu_0}(\hat{h}) \leq \alpha + \epsilon_0$. The first inequality holds because φ is non-decreasing and $\varphi(0) = 1$. Moreover, due to Proposition 1 for the distribution μ_0 and the constraint on $\hat{R}_{\varphi,\mu_0}(\hat{h})$ in (8), we obtain

$$R_{\varphi,\mu_0}(\hat{h}) \leq \hat{R}_{\varphi,\mu_0}(\hat{h}) + \epsilon_0/2 \leq \alpha + \epsilon_0.$$

Part 2) $\mathcal{E}_{1,T}(\hat{h}) \leq \frac{4\tilde{C}}{\sqrt{n_T}}$. Note that since $R_{\varphi,\mu_0}(h^*_{T,\alpha}) \leq \alpha$, Proposition 1 gives us $\hat{R}_{\varphi,\mu_0}(h^*_{T,\alpha}) \leq \alpha + \epsilon_0/2$. Therefore, $h^*_{T,\alpha}$ belongs to the constraint set in the optimization problem (7), which implies that $\hat{R}_{\varphi,\mu_{1,T}}(\hat{h}_{T,\alpha+\epsilon_0/2}) \leq \hat{R}_{\varphi,\mu_{1,T}}(h^*_{T,\alpha})$. Then, we can get

$$\begin{aligned}R_{\varphi,\mu_{1,T}}(\hat{h}) - R_{\varphi,\mu_{1,T}}(h^*_{T,\alpha}) &\leq \hat{R}_{\varphi,\mu_{1,T}}(\hat{h}) - \hat{R}_{\varphi,\mu_{1,T}}(h^*_{T,\alpha}) + \frac{\tilde{C}}{\sqrt{n_T}} \\ &\leq \hat{R}_{\varphi,\mu_{1,T}}(\hat{h}_{T,\alpha+\epsilon_0/2}) - \hat{R}_{\varphi,\mu_{1,T}}(h^*_{T,\alpha}) + \frac{4\tilde{C}}{\sqrt{n_T}} \leq \frac{4\tilde{C}}{\sqrt{n_T}}\end{aligned}$$

where the first inequality follows from Proposition 1, and the second uses the constraint on $\hat{R}_{\varphi,\mu_{1,T}}(\hat{h})$ in (8).

Part 3) $\mathcal{E}_{1,T}(\hat{h}) \leq c_{\rho(r)} \cdot (\frac{\tilde{C}}{\sqrt{n_S}})^{1/\rho(r)} + 4\cdot\Delta$. First, we define $\tilde{h}_{T,\alpha+\epsilon_0/2}$ as follows:

$$\begin{aligned}\tilde{h}_{T,\alpha+\epsilon_0/2} = \arg\min_{h\in\mathcal{H}} \; & R_{\varphi,\mu_{1,T}}(h) \\ \text{s.t. } & \hat{R}_{\varphi,\mu_0}(h) \leq \alpha + \epsilon_0/2.\end{aligned}$$

Note that we have

$$R_{\varphi,\mu_{1,T}}(h^*_{T,\alpha+\epsilon_0}) \le R_{\varphi,\mu_{1,T}}(\tilde{h}_{T,\alpha+\epsilon_0/2}) \le R_{\varphi,\mu_{1,T}}(\hat{h}_{T,\alpha+\epsilon_0/2}). \tag{11}$$

We then divide the proof into two cases:

Part 3, Case I: If $R_{\varphi,\mu_{1,T}}(h^*_{S,\alpha}) - R_{\varphi,\mu_{1,T}}(\tilde{h}_{T,\alpha+\epsilon_0/2}) > \frac{\tilde{C}}{\sqrt{n_T}}$. Then, (11) implies that $\Delta > \frac{\tilde{C}}{\sqrt{n_T}}$, where Δ is defined as $\Delta = R_{\varphi,\mu_{1,T}}(h^*_{S,\alpha}) - R_{\varphi,\mu_{1,T}}(h^*_{T,\alpha+\epsilon_0})$ in Theorem 1. Therefore, the inequality $\mathcal{E}_{1,T}(\hat{h}) \le c_{\rho(r)} \cdot (\frac{\tilde{C}}{\sqrt{n_S}})^{1/\rho(r)} + 4 \cdot \Delta$ becomes trivial due to part 2.

Part 3, Case II: If $R_{\varphi,\mu_{1,T}}(h^*_{S,\alpha}) - R_{\varphi,\mu_{1,T}}(\tilde{h}_{T,\alpha+\epsilon_0/2}) \le \frac{\tilde{C}}{\sqrt{n_T}}$. We first claim that $h^*_{S,\alpha}$ belongs to the constraint set in (8). To show that, we have

$$R_{\varphi,\mu_{1,T}}(h^*_{S,\alpha}) \le R_{\varphi,\mu_{1,T}}(\tilde{h}_{T,\alpha+\epsilon_0/2}) + \frac{\tilde{C}}{\sqrt{n_T}} \le R_{\varphi,\mu_{1,T}}(\hat{h}_{T,\alpha+\epsilon_0/2}) + \frac{\tilde{C}}{\sqrt{n_T}} \le \hat{R}_{\varphi,\mu_{1,T}}(\hat{h}_{T,\alpha+\epsilon_0/2}) + \frac{3\tilde{C}}{2\sqrt{n_T}}.$$

Furthermore, we can obtain

$$\hat{R}_{\varphi,\mu_{1,T}}(h^*_{S,\alpha}) \le R_{\varphi,\mu_{1,T}}(h^*_{S,\alpha}) + \frac{\tilde{C}}{2\sqrt{n_T}} \le \hat{R}_{\varphi,\mu_{1,T}}(\hat{h}_{T,\alpha+\epsilon_0/2}) + \frac{2\tilde{C}}{\sqrt{n_T}},$$

which implies that $h^*_{S,\alpha}$ belongs to the constraint set in (8). Hence, we get

$$R_{\varphi,\mu_{1,S}}(\hat{h}) - R_{\varphi,\mu_{1,S}}(h^*_{S,\alpha}) \le \hat{R}_{\varphi,\mu_{1,S}}(\hat{h}) - \hat{R}_{\varphi,\mu_{1,S}}(h^*_{S,\alpha}) + \frac{\tilde{C}}{\sqrt{n_S}} \le \frac{\tilde{C}}{\sqrt{n_S}}.$$

Then, since $R_{\varphi,\mu_0}(\hat{h}) \le \alpha + \epsilon_0 \le \alpha + r$, by Definition 5 we obtain

$$R_{\varphi,\mu_{1,T}}(\hat{h}) - R_{\varphi,\mu_{1,T}}(h^*_{S,\alpha}) \le c_{\rho(r)} \cdot (\frac{\tilde{C}}{\sqrt{n_S}})^{1/\rho(r)}.$$

Therefore,

$$\begin{aligned}
R_{\varphi,\mu_{1,T}}(\hat{h}) - R_{\varphi,\mu_{1,T}}(h^*_{T,\alpha}) &\le R_{\varphi,\mu_{1,T}}(\hat{h}) - R_{\varphi,\mu_{1,T}}(\tilde{h}_{T,\alpha+\epsilon_0/2}) \\
&= R_{\varphi,\mu_{1,T}}(\hat{h}) - R_{\varphi,\mu_{1,T}}(h^*_{S,\alpha}) + R_{\varphi,\mu_{1,T}}(h^*_{S,\alpha}) - R_{\varphi,\mu_{1,T}}(\tilde{h}_{T,\alpha+\epsilon_0/2}) \\
&\le c_{\rho(r)} \cdot (\frac{\tilde{C}}{\sqrt{n_S}})^{1/\rho(r)} + R_{\varphi,\mu_{1,T}}(h^*_{S,\alpha}) - R_{\varphi,\mu_{1,T}}(h^*_{T,\alpha+\epsilon_0}) \\
&\le c_{\rho(r)} \cdot (\frac{\tilde{C}}{\sqrt{n_S}})^{1/\rho(r)} + 4 \cdot \Delta.
\end{aligned}$$

□

C Proof of Theorem 2

We use some ideas from the proof of Proposition 4.1 in Rigollet & Tong (2011). First, we show that $\gamma(\alpha) := \inf_{h_\theta \in \mathcal{H}_\alpha(\mu_0)} R_{\varphi,\mu_{1,T}}(h_\theta)$ is a non-increasing convex function on $[0,1]$, where $\mathcal{H}_\alpha(\mu_0) = \{h_\theta \in \mathcal{H} : R_{\varphi,\mu_0}(h_\theta) \le \alpha\}$. The non-increasing property is straightforward to verify.

Next, we take $\alpha_1, \alpha_2 \in [0,1]$ and aim to show that for any $\theta \in (0,1)$, the following inequality holds:

$$\gamma(\bar{\alpha}) \le \theta\gamma(\alpha_1) + (1-\theta)\gamma(\alpha_2). \tag{12}$$

where $\bar{\alpha} = \theta\alpha_1 + (1-\theta)\alpha_2$. Let $\epsilon > 0$ be an arbitrary small number. Then, there exist $h_1 \in \mathcal{H}_{\alpha_1}(\mu_0)$ and $h_2 \in \mathcal{H}_{\alpha_2}(\mu_0)$ such that $R_{\varphi,\mu_{1,T}}(h_1) \le \gamma(\alpha_1)+\epsilon$ and $R_{\varphi,\mu_{1,T}}(h_2) \le \gamma(\alpha_2)+\epsilon$. Consider the convex combination $h_3 = \theta \cdot h_1 + (1-\theta) \cdot h_2$, which by assumption belongs to $\mathcal{H}$. By the convexity of φ we have

$$R_{\varphi,\mu_0}(h_3) \le \theta R_{\varphi,\mu_0}(h_1) + (1-\theta) R_{\varphi,\mu_0}(h_2) \le \bar{\alpha},$$

This implies that $h_3 \in \mathcal{H}_{\bar{\alpha}}(\mu_0)$. Therefore,

$$\gamma(\bar{\alpha}) \le R_{\varphi,\mu_{1,T}}(h_3) \le \theta R_{\varphi,\mu_{1,T}}(h_1) + (1-\theta) R_{\varphi,\mu_{1,T}}(h_2) \le \theta\gamma(\alpha_1) + (1-\theta)\gamma(\alpha_2) + \epsilon.$$

Since $\epsilon > 0$ is arbitrary, we conclude that the inequality (12) holds, which implies that

$$\gamma(\alpha) - \gamma(\alpha + \epsilon_0) \le \epsilon_0 \frac{\gamma(\alpha/2) - \gamma(\alpha)}{\alpha/2} \le \frac{C\epsilon_0}{\alpha/2}.$$

Then, we can bound the following term:

$$R_{\varphi,\mu_{1,T}}(h^*_{T,\alpha}) - R_{\varphi,\mu_{1,T}}(h^*_{T,\alpha+\epsilon_0}) = \gamma(\alpha) - \gamma(\alpha + \epsilon_0) \le \frac{2C\tilde{C}}{\alpha\sqrt{n_0}}.$$

□

D More Details on the Climate Dataset (Yu et al., 2024)

In Section 6.1, we used clusters 26 and 27 in one experiment, and clusters 36, 37, 38 in another, as source and target pairs, as illustrated in Figure 5. The original dataset (Yu et al., 2024) includes various locations specified by longitude and latitude, as shown in Figure 6a. Since each location does not have sufficient data for creating training and test samples, we group neighboring locations to form clusters, as illustrated in Figure 6b. These clustered locations are then used as source and target pairs.

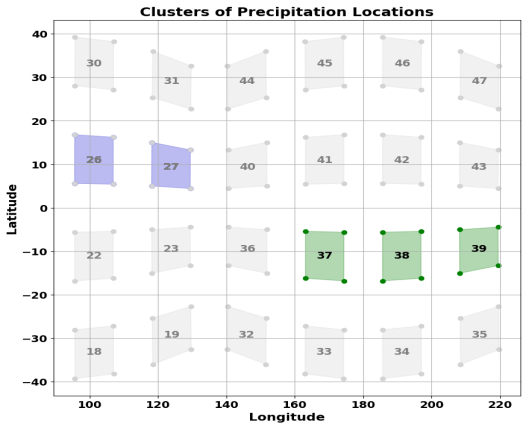
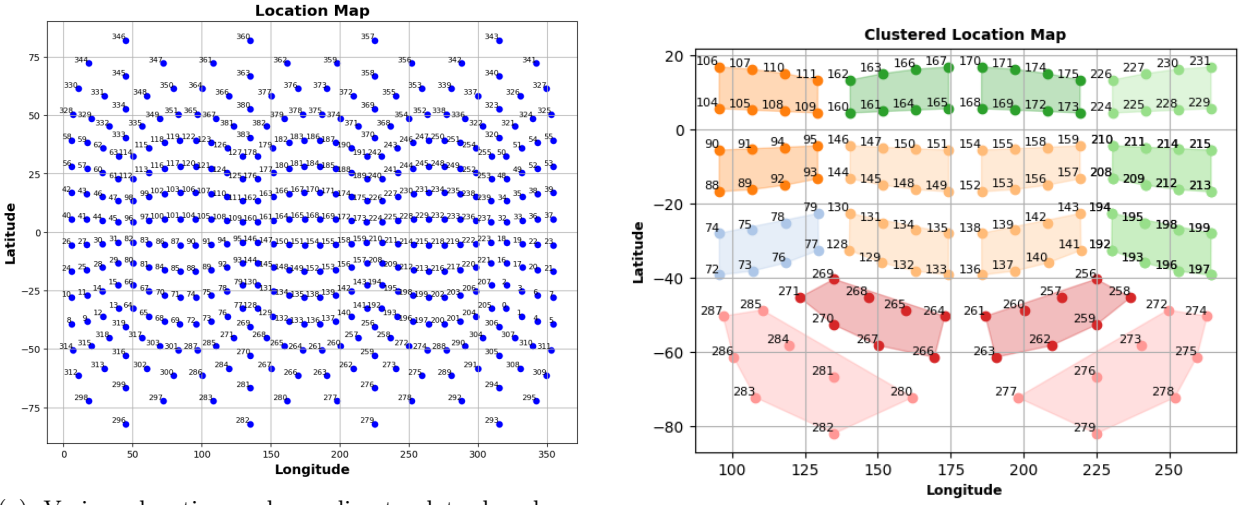


Figure 5: Clusters of locations for rain precipitation data (Yu et al., 2024), used as source-target pairs. In one scenario, $(26, 27)$ forms a source-target pair, while in another scenario, 38 is the target, and 37 and 39 grouped together constitute the source.

D.1 Synthetic Data Experiments

In this section, we evaluate the performance of our algorithm alongside other approaches on Gaussian data. Since it is unlikely to have highly similar source and target distributions in real datasets, we investigate this scenario using synthetic data. Furthermore, in this experiment, we use another instantiation of our algorithm with quadratic models.

Sample Dataset: We generate three datasets corresponding to the normal class, target abnormal class, and source abnormal class, each sampled from standard Gaussian distributions with means 0, 0.6, and 0.6, respectively, and a covariance matrix I_{15}, where the number of features is 15. In this case, the source and



(a) Various locations where climate data has been recorded.

(b) Clustered locations by grouping neighboring ones.

Figure 6

target distributions are exactly the same. For the target, we generate 4,000 training data points for the normal class and $n_T = 50$ for the abnormal class. The number of source abnormal points, n_S, is varied between 100 and 2,500.

Training: We use a quadratic model, $x^T \mathbf{A} x + \mathbf{b}^T x + c$, where $\mathbf{A}, \mathbf{b}$, and c are the parameters to be learned. Additionally, we employ exponential loss as the surrogate loss function and use the Adam optimizer for training. The results are averaged over 10 runs for each experiment, with new data generated for each run.

Results: The Type-I error threshold is set to $\alpha = 0.05$, with $\epsilon_0 = 0.01$. In this case, since the source and target distributions are exactly the same, methods that naively use the source data are expected to perform very well. Figure 7 shows that TLNP outperforms all other methods when the number of source samples is large enough.

E Alternative Approach to Filter $\hat{\mathcal{H}}$ in Step 2 of TLNP (Section 5)

In Section 5, in Step 2, we use a universal constant $c = 0.5$ in the inequality (10) to filter the functions in $\hat{\mathcal{H}}$. As the constant serves primarily to upper-bound the variance of errors for a given dataset, we propose an alternative approach here by estimating the variance as follows.

First, we divide the target abnormal data into 70% for training and 30% for evaluation. Let n_T represent the number of data points in the training set. Step 1 is the same as the procedure described in Section 5.

In Step 2, we first repeat Step 1 using the 30% of the target abnormal training data set aside for evaluation, along with all data from the normal class, i.e., μ_0, and without using any source data, i.e., $\lambda_S = 0$. This process yields a function $\hat{h}_T \in \mathcal{H}$. Inspired by the constraint in the optimization procedure (8), we filter the functions in $\hat{\mathcal{H}}$, obtained in the first step using n_T target abnormal training data, by comparing their performance with that of $\hat{h}_T$ as follows. First, we calculate the output of $\text{sign}(\hat{h}_T)$ on the n_T target abnormal training data and compute the variance of the resulting ± 1 outputs, denoted as VAR. Let $\hat{R}_{\mu_{1,T}}$ represent the target 0-1 loss (Type-II error) computed with respect to the n_T target abnormal training data. We then define $\hat{\mathcal{H}}_T$ as the set of functions $h \in \hat{\mathcal{H}}$ that satisfy the following inequality:

$$\hat{R}_{\mu_{1,T}}(\text{sign}(h)) \leq \hat{R}_{\mu_{1,T}}(\text{sign}(\hat{h}_T)) + \sqrt{\frac{\text{VAR}}{n_T}}$$

Step 3 remains the same as the one described in Section 5.

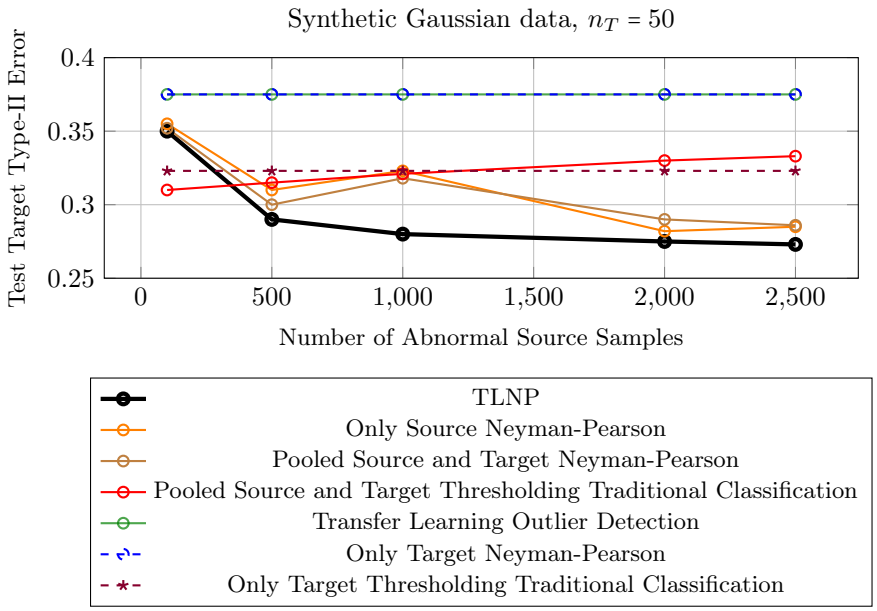


Figure 7: The performance of our algorithm (TLNP), along with other approaches on Gaussian data. The threshold on Type-I error is set at $\alpha = 0.05$. The data consists of three sets: the normal class, the target abnormal class, and the source abnormal class. These are generated according to standard Gaussian distributions with means of 0, 0.6, and 0.6, respectively, and a covariance matrix of I_{15}, where the number of features is 15. Furthermore, the normal class contains 4000 training samples.

In Figures 8 and 9, we demonstrate the results obtained using this approach, referred to as the TLNP variance method, and compare it with the procedure described in Section 5. The results indicate that both methods yield nearly the same performance. In Figure 8, TLNP and TLNP varaince method are identical and overlap completely. In Figure 9, when n_T is sufficiently large, TLNP slightly outperforms TLNP variance method; however, both methods outperform other approaches.

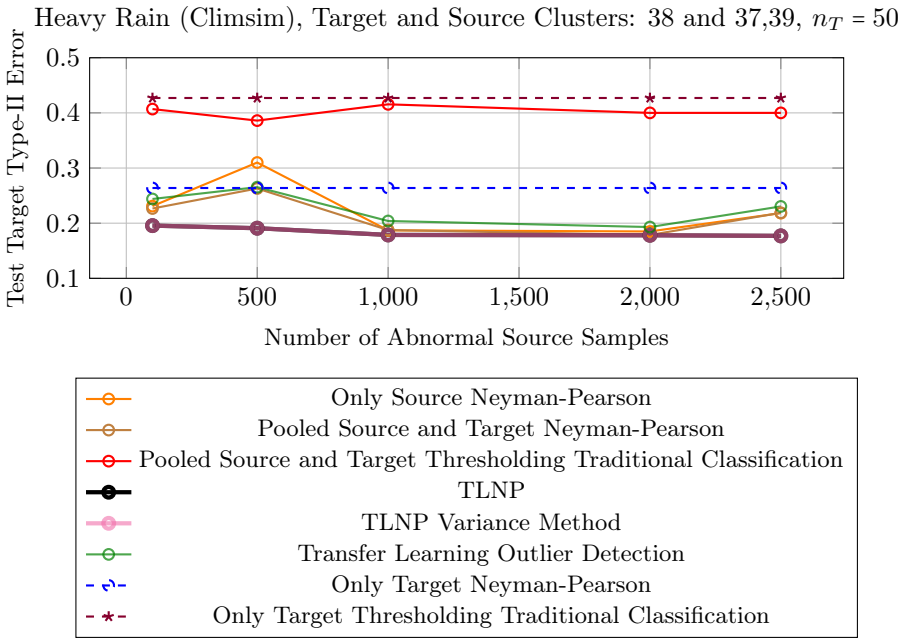


Figure 8: The performance of the TLNP variance method, along with other approaches described in Section 5. The threshold for Type-I error is set at $\alpha = 0.05$, and the experimental settings, including the number of samples, are identical to those in Figure 2. .

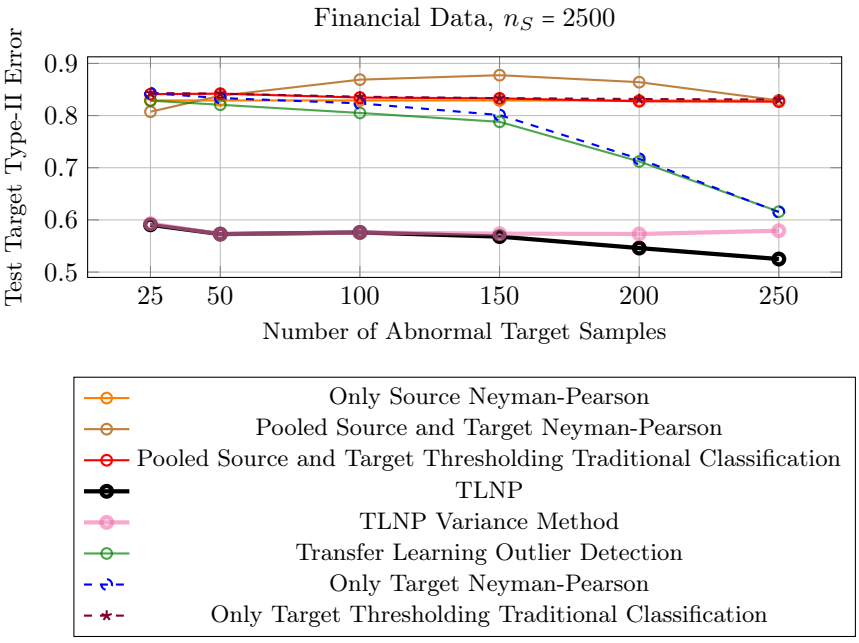


Figure 9: The performance of the TLNP variance method, along with other approaches described in Section 5. The threshold for Type-I error is set at $\alpha = 0.05$, and the experimental settings, including the number of samples, are identical to those in Figure 4