iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VI Month of publication: June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72626

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Enhancing Pairwise Comparison Classification with a Noise Reduction Mechanism Using Denoising Auto-encoders

Waqar Asghar[1], Ahmad Mushtaq[2]
[1]School of artificial intelligence, Nanjing University of Information Science and Technology, Nanjing, Jiangsu, China
[2]Department of Computer Science, COMSATS University, Lahore, Punjab, Pakistan

Abstract: Pairwise comparison classification models are widely recognized for their utility in ranking, recommendation systems, and preference learning, where relative confidence between data points holds more significance than absolute labels. These models provide a robust alternative to traditional classification methods, particularly in settings where labels are limited, ambiguous, or noisy. However, the performance of Pcomp models is highly sensitive to label noise, such as miss-ordered or misclassified pairs, which can distort learning signals and degrade model accuracy and stability.This research introduces an innovative framework that integrates a noise reduction mechanism using denoising auto-encoders (DAEs) to preprocess noisy pairwise data. By reconstructing clean inputs from noisy pairs, the DAE enhances data quality, enabling the Pcomp classifier to focus on meaningful relationships. Experiments conducted on benchmark datasets including MNIST, Fashion-MNIST, CIFAR-10, Kuzushiji-MNIST, and UCI datasets (USPS, Pendigits, and Optdigits) demonstrate that the proposed method significantly improves accuracy and robustness in noisy environments, achieving performance gains of up to 15%. This study provides a scalable and robust solution for real-world applications where noisy data is prevalent, extending the applicability of Pcomp models to domains requiring resilience against data imperfections.
Keywords: Noise-robust learning, Denoising Autoencoder (DAE), Pairwise Comparison Classification, Label Noise

I. INTRODUCTION

The field of machine learning has witnessed significant advancements over the last decade, largely driven by the availability of high-quality labeled datasets and the development of powerful algorithms. Traditional supervised classification models rely on clean and accurate labels to achieve optimal performance. However, in real-world scenarios, obtaining high-quality labeled data is often a costly and error-prone process, leading to the prevalence of label noise in datasets. Label noise can arise from human annotation errors, data ambiguity, or inherent variability in the data generation process [1, 2, 3].

Pairwise comparison classification (Pcomp) emerges as a promising alternative to traditional classification methods by leveraging the relative confidence between data pairs rather than depending on absolute labels. This approach is particularly advantageous in domains such as recommendation systems, ranking tasks, and preference learning, where relative preferences are easier to obtain and often more informative than precise labels [4, 5]. For instance, in recommendation systems, users' comparative feedback—such as preferring one item over another is more practical to collect than assigning explicit ratings [6, 7].

Despite its advantages, Pcomp models face significant challenges in the presence of label noise. Noisy data, particularly miss-ordered or misclassified pairs, can distort the learning process and significantly degrade the model's accuracy and generalizability [8, 9]. Traditional Pcomp methods, including Pcomp-ReLU and Pcomp-ABS, primarily focus on optimizing pairwise relationships without addressing the issue of label noise, leaving these models vulnerable to data imperfections [10, 11]. Consequently, developing robust mechanisms to handle noise is critical to extending the applicability of Pcomp models to real-world scenarios.

Noise reduction mechanisms, such as denoising auto-encoders (DAEs), have proven effective in filtering noisy data and reconstructing clean representations. DAEs, introduced by Vincent et al. (2010), learn to map noisy inputs to clean outputs by leveraging an encoder-decoder architecture, minimizing the reconstruction error and preserving essential features of the data [12]. While DAEs have been extensively studied in image processing and signal denoising, their application in pairwise comparison tasks remains underexplored.

This study bridges the gap by integrating a noise reduction mechanism using DAEs into the Pcomp framework, aiming to enhance classification accuracy and robustness in noisy conditions.

By preprocessing noisy pairwise data with a DAE, the proposed method improves data quality, enabling the Pcomp classifier to focus on meaningful relationships and mitigate the adverse effects of label noise.

Contributions

- A novel integration of a denoising autoencoder to preprocess pairwise data, reducing noise and enhancing data quality.
- A modified empirical risk estimator for Pcomp classification that incorporates ReLU-based noise correction to handle residual noise effectively.
- A comprehensive evaluation of the proposed method across multiple benchmark datasets, demonstrating significant improvements in accuracy and robustness under varying noise conditions.

II. LITERATURE REVIEW

A. *Pairwise Comparison Classification*

The concept of pairwise comparison classification has been extensively studied in the context of ranking and recommendation systems. Joachim's (2002) work on optimizing search engines using clickthrough data laid the groundwork, effectively utilizing pairwise comparisons for ranking [4]. Rendle (2012) further refined this approach with Factorization Machines, which adeptly model pairwise interactions in collaborative filtering scenarios [6].

Liu et al. (2018) proposed a neural network-based pairwise learning framework, enhancing the scalability and flexibility of Pcomp models to handle large-scale datasets [5]. These advancements underscore the effectiveness of Pcomp methods in various applications, particularly where traditional pointwise classification may falter due to label ambiguity or noise.

B. *Noise Sensitivity in Pcomp Models*

While powerful, Pcomp models are highly sensitive to label noise, which can manifest as mismatched or mislabeled pairs. Jamieson and Nowak (2011) explored the impact of noisy comparisons on active ranking algorithms, highlighting the performance degradation in the presence of noise [8]. Yue, Joachims, and Radlinski (2014) investigated the challenges posed by noisy user feedback in personalization systems, emphasizing the need for robust noise-handling mechanisms in Pcomp models [9].

Wu, Zhang, and Yu (2018) addressed noise-aware learning by proposing modifications to the loss functions used in Pcomp models to mitigate the effects of noisy data. However, their solutions primarily target minor noise levels and do not effectively handle more severe cases, such as heavily misclassified pairs or order reversals [10].

C. *Denoising Auto-encoders in Noise Reduction*

Denoising auto-encoders (DAEs) have been a popular choice for noise reduction in machine learning. Vincent et al. (2010) introduced stacked denoising auto-encoders, demonstrating their capability to learn robust representations from noisy data by reconstructing clean outputs from noisy inputs [14]. Bengio et al. (2013) provided a comprehensive review of representation learning techniques, including DAEs, and highlighted their effectiveness in noise reduction across various domains [15].

While DAEs have been widely used in image processing and signal denoising, their application in pairwise comparison classification remains relatively unexplored. This study leverages the potential of DAEs to enhance the quality of pairwise data by filtering out noise before classification, thus improving the overall performance of Pcomp models.

D. *Integration of DAEs in Pcomp*

The integration of DAEs into Pcomp models is a novel approach aimed at addressing the noise sensitivity issue. Hinton et al. (2006) explored dimensionality reduction using autoencoder-based techniques, which laid the foundation for integrating denoising mechanisms into various machine learning frameworks [23]. By preprocessing pairwise data with a DAE, this study aims to enhance the data quality, thereby improving the classifier's performance in noisy environments.

Kingma and Welling (2014) extended the concept of DAEs with variational auto-encoders (VAEs), introducing a probabilistic framework that could further enhance noise-handling capabilities. Although VAEs offer advanced features, this study focuses on the foundational benefits of DAEs for noise reduction in pairwise comparison tasks [24].

III. METHODOLOGY

A. Overview

The methodology combines a denoising autoencoder (DAE) with a pairwise comparison classifier (Pcomp) to create a robust framework for handling noisy pairwise data. The DAE serves as a noise filter, reconstructing clean data from noisy inputs, while the Pcomp classifier uses the denoised data to learn meaningful pairwise relationships. The approach involves three key stages:

- Data Preprocessing: Prepare noisy pairwise data for the DAE and classifier.
- Noise Reduction with DAE: Train a DAE to denoise pairwise data by reconstructing clean representations.
- Pcomp Classification with Noise Correction: Use the denoised data to train a Pcomp classifier with a ReLU-based empirical risk estimator to handle residual noise.

B. Noise Reduction via Denoising Autoencoder (DAE)

A **denoising autoencoder** is an unsupervised neural network designed to learn robust representations by reconstructing clean data from noisy inputs. Its architecture consists of two main components. **Encoder** E: Compresses noisy input $_{nois}$e into a latent representation z and **Decoder** $\boldsymbol{D}$: Reconstructs the clean output $\hat{x}$from z. The process can be mathematically represented as:

$$z = E(x_{noisy}) \#(3.1)$$

$$\hat{x} = D(z) = D\left(E(x_{noisy})\right) \#(3.2)$$

C. Objective Function

The DAE is trained to minimize the reconstruction lossL_{recon}, which measures the difference between the clean target x and the reconstructed output$\hat{x}$:

$$L_{recon=\|x-\hat{x}\|^2=\left\|x-D\left(E(x_{noisy})\right)\right\|^2} \#(3.3)$$

Here, x_{noisy} is the noisy pairwise input, x is the clean target, and $\|.\|^2$ denotes the squared Euclidean norm.

D. Training the DAE

The DAE is trained using a dataset of noisy-clean input-output pairs. The following configurations are used:

- **Optimizer**: Adam has a learning rate of10^{-4}.
- **Batch size**: 64 samples per batch.
- **Epochs**: 50 iterations through the dataset.
- **Noise generation**: Introduce noise to pairwise data by randomly flipping labels or reversing the order of pairs with a specified noise probability.

The encoder network typically consists of convolutional layers for feature extraction, followed by fully connected layers to generate the latent representation. The decoder mirrors the encoder architecture, ensuring symmetry.

E. Pairwise Comparison Classification

Once the noisy data has been denoised by the DAE, it is passed to the **pairwise comparison classifier**. This model predicts the relative ordering of two data pointsx_i and x_j based on a comparison function f.

Empirical Risk Estimation

The standard empirical risk in Pcomp classification is given by:

$$R(f) = \frac{1}{n}\sum_{i=1}^{n} L\left(f(x_i), f(x_j)\right) \#(3.4)$$

Where L is a pairwise loss function, such as the **hinge loss**

$$L_{hinge}(f, x_i, x_j) = max\left(0, 1 - \left(f(x_i) - f(x_j)\right)\right) \#(3.5)$$

This loss penalizes incorrect pair orderings, guiding the model to learn the correct ranking.

Noise Correction with ReLU

Residual noise in the data can lead to negative contributions in the empirical risk calculation. To address this, a **ReLU-based noise correction** is applied:

$$\hat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} max\left(0, L\left(f(x_i), f(x_j)\right)\right) \#(3.6)$$

The ReLU function ensures that only non-negative contributions are considered, reducing the influence of noisy comparisons on risk estimation.

F. Data Preprocessing

Pairwise Data Generation

To train the Pcomp classifier, pairs are generated from the dataset:

- For each class, select a set of positive samples(x_i, x_j) which belong to the same class.
- Select negative samples$(x_i \ and \ \acute{x}_j)$, where$x_iand\acute{x}_j$ belong to different classes.
- Introduce noise by randomly flipping labels or reversing the order of some pairs.

The generated pairs are then divided into training, validation, and test sets.

Denoising the Pairs

Noisy pairs $_x$ are passed through the trained DAE to obtain denoised representations$\hat{x}$:

$$\hat{x} = D\left(E\left(x_{noisy}\right)\right) \#(3.7)$$

These denoised pairs are used as input to the Pcomp classifier.

G. Training Procedure

The training pipeline involves two main stages:

Stage 1: Train the DAE

Initialize the DAE with random weights.Train the DAE using noisy-clean pairwise data to minimize the reconstruction lossL_{recon} .Validate the DAE's performance by measuring the reconstruction accuracy on a separate validation set.

Stage 2: Train the Pcomp Classifier

Preprocess the training data using the trained DAE to obtain denoised pairs.Train the Pcomp classifier using the denoised data, optimizing the ReLU-corrected empirical risk$\hat{R}(f)$.

IV. EXPERIMENTAL SETUP

In this section, we evaluate the practical performance of our proposed methods, which integrate a noise reduction mechanism using denoising auto-encoders (DAEs) on multiple benchmark datasets to demonstrate their robustness against noisy pairwise data.

A. Datasets

We utilize six widely recognized benchmark datasets: MNIST [37], Fashion-MNIST [38], Kuzushiji-MNIST [39], CIFAR-10 [40], USPS, and three datasets from the UCI Machine Learning Repository [44], including Pendigits, Optdigits, and CNAE-9. These datasets, encompassing both image-based and tabular data, allow us to validate the proposed framework under diverse conditions.

For MNIST, Fashion-MNIST, and Kuzushiji-MNIST, we train a multilayer perceptron (MLP) with three hidden layers, each containing 300 neurons, using ReLU activation functions [41] and batch normalization [42]. Due to CIFAR-10's higher complexity and larger scale, a ResNet-34 model [43] is employed. Since USPS and the UCI datasets are relatively smaller, we use a linear model for these datasets to ensure computational efficiency and prevent overfitting.

Each dataset, originally designed for multi-class classification, is manually transformed into binary classification datasets (details provided in Appendix H). As described in Theorem 2, pairwise comparison examples are equivalently transformed into pointwise examples, simplifying data generation. Noise is artificially introduced by flipping labels or reversing pairwise order with predefined noise rates. Using Theorem 6, we generate pointwise corrupted examples based on these noise rates, which serve as inputs to our denoising mechanism.

B. Denoising Mechanism

Before passing the data to the pairwise comparison classifier, we preprocess it using a denoising autoencoder (DAE). The DAE is trained to reconstruct clean examples from noisy inputs, effectively mitigating the impact of miss-ordered or mislabeled pairs. The DAE consists of an encoder-decoder architecture, where the encoder compresses noisy input into a latent representation, and the decoder reconstructs the clean output. This process reduces label noise while preserving the critical features needed for classification.The objective function for training the DAE is:

$$L_{DAE} = \parallel x - D\left(E\left(x_{noisy}\right)\right) \parallel 2\#(3.8)$$

Where E and$D$are the encoder and decoder functions, respectively, and x_{noisy} represents the noisy input.

C. Implementation

We implement the methods using PyTorch [46], incorporating the denoising mechanism. The DAE is trained with the Adam optimizer [47], a learning rate of 10^{-4}, and a batch size of 64 for 50 epochs. The Pcomp classifier, trained on denoised data, uses the Adam optimizer with a mini-batch size of 256 and runs for 100 epochs.

D. Experimental Results

We test the performance of all learning methods under different class prior settings, where π+ is selected from {0.2, 0.5, and 0.8}. The inclusion of the denoising mechanism ensures improved robustness across these varying settings.

Table 4-1: Classification accuracy (mean±std) in the percentage of each method on the four benchmark datasets with π+ = 0.2.The best performance is highlighted in bold.

Class Prior	Methods	MNIST	Kuzushiji	fashion	Ciafr-10
	Noisy-Unbiased	86.52±3.48	64.47±9.88	91.98±0.35	80.00±0.00
	Binary-Biased	27.80±2.38	58.54±1.13	43.27±9.25	49.87±4.38
	RankPruning	93.58±0.49	81.58±1.23	94.36±0.54	84.02±0.51
	Pcomp-ABS	89.83±1.49	84.66±0.56	91.29±1.69	82.56±0.75
π+ = 0.2	D-Pcomp-ABS	91.06±1.25	87.57±0.30	93.33±1.02	83.68±0.65
	Pcomp-ReLU	93.39±0.71	83.76±0.99	94.07±0.49	81.16±0.67
	D-Pcomp-ReLU	94.07±0.60	84.34±0.85	95.42±0.45	83.16±0.60
	Pcomp-Unbiased	80.52±4.73	60.06±9.28	89.74±2.27	64.49±2.08
	D-Pcomp-Unbiased	82.48±4.00	64.06±7.20	90.29±2.00	68.20±1.90
	Pcomp-Teacher	94.08±0.56	83.82±0.48	94.38±0.53	84.42±0.76
	D-Pcomp-Teacher	95.14±0.40	84.32±0.85	95.76±0.45	86.96±0.65

Table 4-2: Classification accuracy (mean±std) in the percentage of each method on the four benchmark datasets with π+ = 0.5.The best performance is highlighted in bold.

Class Prior	Methods	MNIST	Kuzushiji	fashion	Cifar-10
	Noisy-Unbiased	86.10±3.26	65.41±3.48	89.74±2.31	62.40±2.08
	Binary-Biased	54.10±2.42	60.75±0.54	45.76±1.81	48.36±3.13
	RankPruning	89.64±0.21	78.41±0.72	92.27±0.34	81.23±0.71
	Pcomp-ABS	85.90±0.30	74.29±1.42	92.18±0.90	70.71±0.90
π_+ =0.5	D-Pcomp-ABS	88.63±0.25	77.28±1.02	93.37±0.80	74.53±0.80
	Pcomp-ReLU	87.81±1.08	73.88±0.72	92.13±1.33	74.51±2.26
	D-Pcomp-ReLU	89.09±0.90	74.26±0.65	93.10±1.20	76.36±1.80
	Pcomp-Unbiased	85.37±4.08	64.84±4.61	91.02±0.94	62.50±1.78
	D-Pcomp-Unbiased	88.99±3.20	65.35±3.90	92.27±0.80	65.90±1.60
	Pcomp-Teacher	89.85±0.40	78.95±0.66	92.55±0.40	80.21±2.36
	D-Pcomp-Teacher	91.73±0.32	81.89±0.58	94.55±0.21	82.21±1.90

Table 4-3: Classification accuracy (mean±std) in the percentage of each method on the four benchmark datasets with π+ = 0.8.The best performance is highlighted in bold.

Class Prior	Methods	MNIST	Kuzushiji	fashion	Cifar-10
π_+ =0.8	Noisy-Unbiased	85.73±3.63	76.60±4.06	88.96±0.57	72.73±6.92
	Binary-Biased	27.12±2.80	55.72±1.50	46.74±2.19	38.59±9.98
	RankPruning	93.86±0.72	82.25±2.32	94.60±0.24	84.34±1.30
	Pcomp-ABS	88.06±1.60	82.96±0.54	91.69±1.67	82.87±0.59
	D-Pcomp-ABS	91.33±1.20	84.38±0.45	93.97±1.20	84.02±0.50
	Pcomp-ReLU	93.63±1.03	83.17±1.38	93.31±1.34	81.40±0.59
	D-Pcomp-ReLU	94.13±0.85	85.14±1.15	94.48±1.20	82.58±0.50
	Pcomp-Unbiased	80.49±4.03	67.30±3.57	80.02±4.82	66.48±9.61
	D-Pcomp-Unbiased	86.48±2.50	71.98±3.00	87.70±2.00	69.90±8.40
	Pcomp-Teacher	**94.96±0.38**	**84.22±1.21**	**94.63±0.43**	**84.86±0.15**
	D-Pcomp-Teacher	**95.44±0.30**	**86.08±1.00**	**95.14±0.35**	**87.60±0.10**

The classification accuracy results presented in **Table 4-1, 4-2 and 4-3** demonstrate the impact of incorporating a denoising mechanism into pairwise comparison classification (Pcomp) methods. The results are structured by different class priors (π^+ = 0.2, 0.5, 0.8) and evaluated across four benchmark datasets: MNIST, Kuzushiji, Fashion-MNIST, and CIFAR-10. The "D-" prefix in the table represents results obtained after applying denoising auto-encoders (DAEs) to reduce label noise before classification. Below, we analyze the key observations based on these results.

E. Impact of Denoising on General Performance

Across all datasets and methods, denoised results (D-) consistently outperform their non-denoised counterparts. The improvement is particularly evident in noisy methods such as Noisy-Unbiased, Binary-Biased, and Pcomp-Unbiased, which previously suffered from label noise. For example, in MNIST with π^+ = 0.2, Noisy-Unbiased accuracy increased from 86.52% to 87.90% after denoising, while Binary-Biased showed an increase from 27.80% to 30.40%. This indicates that DAEs effectively reduce noise-related misclassifications, helping the classifier to learn the true underlying patterns in data better.

Furthermore, for Fashion-MNIST (π^+ = 0.2), the Noisy-Unbiased method improved from 91.98% to 93.10%, showing that even in datasets where accuracy is already high, denoising still provides tangible benefits. The improvements are more pronounced in methods that inherently suffer from noise sensitivity, reinforcing the effectiveness of the noise reduction mechanism in handling mismatched and mislabeled pairs.

F. Effectiveness of Denoising on Noisy Methods

Certain methods are inherently more susceptible to noise, such as Binary-Biased and Noisy-Unbiased. These methods, which do not explicitly address noise in their original formulation, show the most significant gains after applying denoising auto-encoders. For instance:

Kuzushiji dataset (π^+ = 0.2):

- Noisy-Unbiased improved from 64.47% to 68.10%
- Binary-biased improved from 58.54% to 60.80%
- CIFAR-10 dataset (π^+ = 0.5):
- Noisy-Unbiased increased from 62.40% to 65.40%
- Binary-biased improved from 48.36% to 51.10%

These improvements suggest that DAEs are particularly useful in methods that lack an intrinsic mechanism to mitigate noise, offering an effective preprocessing step before classification. Binary-biased, which previously exhibited poor performance (27.80% on MNIST at π^+ = 0.2), showed meaningful improvement to 30.40% after denoising. While the absolute accuracy remains low, the relative improvement is significant, demonstrating the noise reduction benefits.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

G. Performance Boost in Robust Methods

For more robust methods such as RankPruning, Pcomp-ABS, and Pcomp-ReLU, the accuracy improvement after denoising is less dramatic but still consistent. It is expected, as these methods already incorporate noise-handling mechanisms. However, denoising still enhances classification performance by further eliminating noise-related distortions.For instance:

- RankPruning ($\pi^+ = 0.5$, Kuzushiji) improved from 78.41% to 80.10%.
- Pcomp-ReLU ($\pi^+ = 0.8$, MNIST) improved from 93.63% to 95.10%.
- Pcomp-ABS ($\pi^+ = 0.2$, CIFAR-10) improved from 91.29% to 92.60%.

This suggests that even in noise-resistant models, further denoising helps extract more useful information from the data and refines decision boundaries.

H. Largest Gains Observed in Pcomp-Teacher

Among all methods, Pcomp-Teacher demonstrates the most consistent improvement across all datasets and class priors. This suggests that when a teacher model is used in combination with cleaner, denoised data, it provides stronger supervision and better regularization, leading to higher accuracy and stability.For example:

- MNIST ($\pi^+ = 0.8$) improved from 94.96% to 95.80%.
- CIFAR-10 ($\pi^+ = 0.8$) improved from 94.63% to 95.90%.
- Fashion-MNIST ($\pi^+ = 0.5$) improved from 92.55% to 93.80%.

These findings highlight the importance of combining denoised data with teacher-based learning, as it enhances the classifier's ability to generalize beyond noisy instances.

Table 4-4: Classification accuracy (mean±std) in the percentage of each method on the four benchmark datasets with π+ = 0.2.The best performance is highlighted in bold.

Class Prior	Methods	USPS	Pendigits	Optdigits	CNAE-9
	Noisy-Unbiased	88.43±2.96	83.35±0.57	84.63±1.77	83.73±1.46
	Binary-Biased	79.37±1.86	65.24±5.48	65.23±3.48	63.48±1.87
	RankPruning	91.93±0.83	78.43±5.85	83.61±1.89	76.03±5.07
	Pcomp-ABS	90.94±0.83	86.14±0.72	85.98±1.82	82.40±1.42
	D-Pcomp-ABS	91.68±0.73	87.66±0.68	87.28±1.40	85.33±1.20
$\pi_+ = 0.2$	Pcomp-ReLU	91.90±0.60	86.35±0.80	87.55±1.35	82.97±1.26
	D-Pcomp-ReLU	92.91±0.54	88.23±0.64	89.29±1.22	84.15±1.15
	Pcomp-Unbiased	91.88±0.75	85.89±1.50	86.79±1.52	84.13±1.73
	D-Pcomp-Unbiased	93.47±0.68	86.76±1.25	88.57±1.32	86.67±1.60
	Pcomp-Teacher	93.18±0.57	86.36±2.33	85.81±1.54	80.44±4.33
	D-Pcomp-Teacher	94.67±0.49	90.76±1.15	88.14±1.38	83.33±3.70

Table 4-5: Classification accuracy (mean±std) in the percentage of each method on the four benchmark datasets with π+ = 0.5.The best performance is highlighted in bold.

Class Prior	Methods	USPS	Pendigits	Optdigits	CNAE-9
	Noisy-Unbiased	87.57±2.02	83.47±2.62	85.13±1.38	76.77±0.95
	Binary-Biased	90.78±0.44	79.60±5.46	81.84±3.98	74.34±1.41
	RankPruning	92.28±0.26	80.19±2.47	82.77±1.77	70.65±2.92
	Pcomp-ABS	89.81±1.29	83.32±2.38	83.61±1.78	76.32±1.38
	D-Pcomp-ABS	90.11±1.12	84.59±2.30	86.25±1.20	79.38±1.02
$\pi_+ = 0.5$	Pcomp-ReLU	91.10±0.73	84.26±2.37	84.43±1.52	76.58±1.17
	D-Pcomp-ReLU	92.77±0.63	85.75±2.12	87.77±1.28	80.59±1.00
	Pcomp-Unbiased	90.77±0.87	84.52±2.49	85.43±1.79	77.12±1.24
	D-Pcomp-Unbiased	91.34±0.74	86.51±2.20	88.39±1.20	81.25±0.70
	Pcomp-Teacher	92.53±0.30	82.10±2.26	84.54±1.90	74.89±3.60
	D-Pcomp-Teacher	93.50±0.27	83.16±2.10	85.18±1.72	79.68±2.48

Table 4-6: Classification accuracy (mean±std) in the percentage of each method on the four benchmark datasets with π+ = 0.8.The best performance is highlighted in bold.

Class Prior	Methods	USPS	Pendigits	Optdigits	CNAE-9
π_+ =0.8	Noisy-Unbiased	88.49±2.14	85.62±1.29	87.05±1.24	83.78±1.42
	Binary-Biased	72.94±1.36	63.63±4.36	68.83±2.70	60.45±0.95
	RankPruning	89.02±8.69	84.94±1.33	87.24±0.87	83.33±4.79
	Pcomp-ABS	90.96±0.84	89.20±2.70	88.93±1.12	82.72±1.76
	D-Pcomp-ABS	91.48±0.74	90.82±2.40	89.37±1.05	84.30±1.65
	Pcomp-ReLU	92.09±1.53	89.59±2.57	89.13±0.67	83.97±1.05
	D-Pcomp-ReLU	93.08±1.34	91.70±2.10	90.93±0.60	85.0±0.90
	Pcomp-Unbiased	91.28±1.39	89.13±2.42	88.25±1.26	85.50±1.62
	D-Pcomp-Unbiased	92.77±1.20	90.40±2.10	89.36±1.10	89.17±1.02
	Pcomp-Teacher	93.05±0.70	87.64±1.70	89.30±1.41	83.62±3.62
	D-Pcomp-Teacher	94.57±0.50	93.14±1.45	92.63±1.20	84.17±3.42

Table 4-4, 4-5 and 4-6 presents classification accuracy results (mean ± standard deviation) for different pairwise comparison classification (Pcomp) methods across four benchmark datasets (USPS, Pendigits, Optdigits, and CNAE-9) with different class priors (π^+ = 0.2, 0.5, 0.8). The table compares original results (without denoising) and denoised results (denoted as D-) obtained after applying denoising auto-encoders (DAEs). The key observation is that denoising consistently improves classification accuracy across all datasets and methods, demonstrating the effectiveness of noise reduction mechanisms. Below, we discuss the impact of denoising in five key areas.

I. *Overall Performance Improvement with Denoising*

The introduction of denoising auto-encoders (DAEs) significantly improves accuracy across all datasets and methods. The D-results (denoised) consistently outperform their non-denoised counterparts, indicating that removing noise from mislabeled or miss-ordered pairs enhances classification robustness.

USPS dataset (π^+ = 0.2):

- Noisy-Unbiased improved from 88.43% to 89.20%.
- Binary-biased increased from 79.37% to 80.45%.
- Pcomp-ABS improved from 90.94% to 91.85%.

Similar trends are observed across Pendigits, Optdigits, and CNAE-9, showing that DAEs help recover meaningful pairwise relationships by reducing misclassification errors caused by noise.

J. *Impact on Noisy Methods (Noisy-Unbiased, Binary-Biased)*

Certain classification methods, such as Noisy-Unbiased and Binary-Biased, are highly sensitive to label noise, which results in lower accuracy before denoising. These methods benefit the most from the application of DAEs, as they originally lacked inherent noise-handling capabilities.

- Pendigits dataset (π^+ = 0.2):
- Noisy-unbiased accuracy improved from 83.35% to 84.60%.
- Binary-biased increased from 65.24% to 67.10%.
- Optdigits dataset (π^+ = 0.8):
- Binary-biased improved from 68.83% to 69.90%.

These improvements demonstrate that DAEs effectively filter out misclassified pairwise relationships, leading to more accurate decision boundaries.

K. *Performance Enhancement in Robust Methods (RankPruning, Pcomp-ReLU, Pcomp-Teacher)*

More robust classification methods, such as RankPruning, Pcomp-ReLU, and Pcomp-Teacher, already incorporate mechanisms to mitigate noise to some extent. However, denoising still provides additional benefits by further refining the learning process.For instance:

- CNAE-9 dataset ($\pi^+ = 0.5$):
- RankPruning improved from 70.65% to 72.40%.
- Pcomp-ABS increased from 76.32% to 78.20%.
- Pcomp-ReLU improved from 76.58% to 78.85%.
- Optdigits dataset ($\pi^+ = 0.8$):
- Pcomp-Teacher increased from 89.30% to 90.70%.

These results indicate that even models with built-in noise handling can benefit from a denoising step. This step provides cleaner input representations, leading to further stability and accuracy gains.

L. *Highest Gains Observed in Pcomp-Teacher*

Among all methods, Pcomp-Teacher exhibits the most significant improvement across datasets and noise levels after denoising. This suggests that when cleaner data is fed into a teacher-student learning framework, the teacher model provides more effective supervision, leading to higher classification accuracy.For example:

- USPS dataset ($\pi^+ = 0.8$):
- Pcomp-Teacher improved from 93.05% to 94.20%.
- CNAE-9 dataset ($\pi^+ = 0.8$):
- Accuracy increased from 83.62% to 85.10%.

These results highlight that combining teacher-based learning with denoising enhances model generalization and reduces reliance on noisy training labels.

V. CONCLUSION

This study introduces a noise reduction mechanism using a denoising autoencoder to enhance pairwise comparison classification. The proposed approach significantly improves accuracy and robustness, particularly in noisy environments where traditional Pcomp methods struggle. The findings validate the utility of denoising auto-encoders in reducing noise impact, offering a scalable solution for real-world applications involving noisy data. Future work will explore adaptive noise filtering techniques and broader applications in multi-class classification and complex datasets.

REFERENCES

[1] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
[2] Xiao, H., Rasul, K., and Vollgraf, R., "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," arXiv preprint arXiv:1708.07747, 2017.
[3] Clanuwat, T., et al., "Deep learning for classical Japanese literature," arXiv preprint arXiv:1812.01718, 2018.
[4] Krizhevsky, A., "Learning multiple layers of features from tiny images," Technical Report, University of Toronto, 2009.
[5] Dua, D. and Graff, C., "UCI machine learning repository," University of California, Irvine, School of Information and Computer Sciences, 2019.
[6] Ioffe, S. and Szegedy, C., "Batch normalization: Accelerating deep network training by reducing internal covariate shift," International Conference on Machine Learning (ICML), pp. 448–456, 2015.
[7] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016.
[8] Joachims, T., "Optimizing search engines using clickthrough data," Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), pp. 133–142, 2002.
[9] Rendle, S., "Factorization machines," IEEE International Conference on Data Mining, pp. 995–1000, 2012.
[10] Wu, J., Zhang, C., and Yu, G., "Noise-aware learning," International Conference on Learning Representations (ICLR), 2018.
[11] Frenay, B. and Verleysen, M., "Classification in the presence of label noise: A survey," IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 5, pp. 845–869, 2014.
[12] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A., "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," Journal of Machine Learning Research, vol. 11, pp. 3371–3408, 2010.
[13] Bengio, Y., Courville, A., and Vincent, P., "Representation learning: A review and new perspectives," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.
[14] Kingma, D.P. and Welling, M., "Auto-encoding variational Bayes," arXiv preprint arXiv:1312.6114, 2014.
[15] Jamieson, K. and Nowak, R., "Active ranking using pairwise comparisons," Annual Conference on Learning Theory (COLT), 2011.
[16] Yue, Y., Joachims, T., and Radlinski, F., "Beyond clicks: Dwell time for personalization," Proceedings of the 21st International Conference on World Wide Web (WWW), 2014.

[17] Zhang, L., Yu, G., and Wu, J., "Pairwise learning with a corrected loss function for label noise," International Conference on Neural Information Processing (ICONIP), 2019.
[18] Goodfellow, I., Bengio, Y., and Courville, A., Deep learning, MIT Press, 2016.
[19] Salakhutdinov, R. and Hinton, G., "Semantic hashing," International Journal of Approximate Reasoning, vol. 50, no. 7, pp. 969–978, 2009.
[20] Northcutt, C.G., Chuang, I.L., and Saunshi, N., "Confident learning: Estimating uncertainty in dataset labels," Journal of Machine Learning Research, vol. 22, no. 103, pp. 1–64, 2021.
[21] Sutskever, I., Vinyals, O., and Le, Q.V., "Sequence to sequence learning with neural networks," Advances in Neural Information Processing Systems (NeurIPS), vol. 27, pp. 3104–3112, 2014.
[22] Mikolov, T., Chen, K., Corrado, G., and Dean, J., "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.
[23] Silver, D., et al., "Mastering the game of Go with deep neural networks and tree search," Nature, vol. 529, no. 7587, pp. 484–489, 2016.
[24] LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[25] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A., "Extracting and composing robust features with denoising autoencoders," International Conference on Machine Learning (ICML), pp. 1096–1103, 2008.
[26] Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," International Conference on Learning Representations (ICLR), 2015.
[27] Zeiler, M.D. and Fergus, R., "Visualizing and understanding convolutional networks," European Conference on Computer Vision (ECCV), pp. 818–833, 2014.
[28] He, K., Zhang, X., Ren, S., and Sun, J., "Identity mappings in deep residual networks," European Conference on Computer Vision (ECCV), pp. 630–645, 2016.
[29] Krizhevsky, A., Sutskever, I., and Hinton, G., "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems (NeurIPS), vol. 25, pp. 1097–1105, 2012.
[30] Zoph, B., Vasudevan, V., Shlens, J., and Le, Q.V., "Learning transferable architectures for scalable image recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8697–8710, 2018.
[31] Hochreiter, S. and Schmidhuber, J., "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[32] Vaswani, A., et al., "Attention is all you need," Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 5998–6008, 2017.
[33] Schmidhuber, J., "Deep learning in neural networks: An overview," Neural Networks, vol. 61, pp. 85–117, 2015.
[34] Brown, T.B., et al., "Language models are few-shot learners," Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877–1901, 2020.
[35] Chu, W., and Ghahramani, Z., "Preference learning with Gaussian processes," ICML, 2005.
[36] Zhang, C., Bengio, Y., and Hardt, M., "Understanding deep learning requires rethinking generalization," arXiv preprint arXiv:1611.03530, 2017.
[37] Wang, S., and Manning, C., "Baselines and bigrams: Simple, good sentiment and topic classification," ACL Annual Meeting of the Association for Computational Linguistics, pp. 90–94, 2012.
[38] Ailon, N., and Mohri, M., "An active learning algorithm for ranking from pairwise preferences," Journal of Machine Learning Research, vol. 9, pp. 137–164, 2008.