Towards reliable and fair probabilistic predictions: field-aware calibration with neural networks

Feiyang Pan[1,2], **Xiang Ao**[1,2], **Pingzhong Tang**[3], **Min Lu**[4], **Dapeng Liu**[4], **Qing He**[1,2]
[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences [3]Tsinghua University [4]Tencent
panfeiyang@ict.ac.cn

Abstract

In machine learning, it is observed that probabilistic predictions sometimes disagree with averaged actual outcomes on certain subsets of data. This is also known as *miscalibration* that is responsible for unreliability and unfairness of practical machine learning systems.

In this paper, we put forward an evaluation metric for calibration, coined *field-level calibration error*, that measures bias in predictions over the input fields that the decision maker concerns. We show that existing calibration methods perform poorly under our new metric. Specifically, after learning a calibration mapping over the validation dataset, existing methods have limited improvements in our error metric and completely fail to improve other non-calibration metrics such as the AUC score. We propose Neural Calibration, a new calibration method, which learns to calibrate by making full use of all input information over the validation set. We test our method on five large-scale real-world datasets. The results show that Neural Calibration significantly improves against uncalibrated predictions in all well-known metrics such as the negative log-likelihood, the Brier score, the AUC score, as well as our proposed field-level calibration error.

1 Introduction

Current decision-making systems are often supported by machine learning agents that are utility maximizers. To make reliable decisions, it is at the heart of machine learning models to make accurate probabilistic predictions. Unfortunately, it has been observed that many existing machine learning methods especially deep learning methods can yield poorly calibrated probabilistic predictions [9, 2, 6], which hurts reliability of the decision-making systems.

For a binary classification problem, a machine learning model is said to be well calibrated if it makes probabilistic predictions that agree with the actual outcomes [14, 6]. That is, when the model makes predictions on an unseen data set, in any subset of the data, if the averaged prediction is p, the actual outcomes will do occur around p fraction of the times.

It has been reported in recent studies [6, 3, 5] that, in the field of computer vision and information retrieval, deep neural networks can make poorly calibrated probabilistic predictions. It is also observed that on several general machine learning and data mining tasks, miscalibration not only affects the overall utility, but also undermines the fairness on certain groups of data. In particular, some common deep models can make predictions desirable with respect to non-calibration performance measures, but suffer from poor results in calibration-related measures.

Let us look at an example, the details of which will be shown in Section 3.2. We train a multi-layer perceptron (MLP) over a public dataset[1] to predict whether an issued loan would be in default. The

[1]The Lending Club loan data: https://www.kaggle.com/wendykan/lending-club-loan-data

Preprint. Under review.

trained neural network achieves a high AUC (area under curve) score on the test set. Unfortunately, we found the predicted probabilities unfair among borrowers in different states in the U.S. In this case, the trained model seems "powerful" for certain utility functions such as the AUC score, but it is unreliable because of the large miscalibration error and unfairness to some specific subsets of data. Therefore, it is crucial to calibrate the predictions so as not to mislead the decision maker.

For this purpose, we first raise the following question.

Q1. *Given probabilistic predictions on a test dataset, how to measure the calibration error?*

Perhaps surprisingly, common metrics are insufficient to evaluate miscalibration errors. In particular, they cannot be used to report biases over subsets of data, such as the aforementioned example of "unfairness" over borrowers in different U.S. states. For example, Negative Log-Likelihood and the Brier score, arguably the two most popular metrics, can solely evaluate the error on instance level which is too fine-grained to measure miscalibration on subsets. On the other hand, the reliability diagram [4] can only visualize the averaged the error on probability intervals, thus is too coarse grained at subset level.

To answer ***Q1***, we put forward a new class of evaluation metrics, coined the *Field-level Calibration Error*. It can evaluate the averaged bias of predictions over specific input fields, which is especially useful on categorical data. Take the loan defaulter prediction task as an example, the new metric can measure the unfairness on the field "address state".

We observe that the proposed field-level calibration error indeed measures the error ignored by previous metrics. That is, a set of predictions can simultaneously get high AUC score, low Log-loss, but large field-level calibration error.

Various calibration methods have been proposed to fix miscalibration, e.g., [1, 17, 18, 16, 14, 6].. A standard pipeline for calibration builds a mapping function on a validation (development) dataset that transforms raw model outputs into calibrated probabilities. By using the mapping function, the error on the hold-out data can then be reduced. However, such methods might be insufficient in practice: when directly training a model over the joint of the training set and the validation set, we observe that it can reach a much higher AUC score comparing to conventional calibration methods. Therefore, a practical question arises:

Q2. *Can we simultaneously reduce the calibration error and improve other non-calibration metrics such as the AUC score?*

Our answer is "Yes". To achieve this, we propose a neural network based method, coined Neural Calibration. Rather than learning a mapping from the raw model output to a calibrated probability like previous work, Neural Calibration trains a neural network over the validation set by taking both the raw model output and all other features as inputs. This method is simple yet powerful to use.

It naturally follows a learning pipeline for general machine learning and data mining tasks: first train a base model on the training set, and then train a Neural Calibration model over a validation dataset. We conducted experiments over five large scale real-world datasets to verify the effectiveness. We show that by using our learning pipeline, the resulted predictions can not only achieve lower calibration error than previous calibration methods, but also reach a comparable or better performance on non-calibration metrics compared with the joint training pipeline.

Our contribution can be summarized as follows:

- We put forward Field-level Calibration Error, a new type of metric to measure miscalibration. It focuses on detecting the bias on specific subset of data. We observe that the new metric indeed reports errors that are overlooked by existing metrics.
- We propose Neural Calibration, which takes the uncalibrated model output along with other input features as input and outputs calibrated probabilistic predictions.
- It follows a pipeline for practitioners in machine learning and data mining, which can achieve strong results in both calibration metrics and non-calibration metrics.

What we do not study: This paper is not related to the literature of fairness-aware classification [7, 19, 13] which aims to give absolute fair predictions for sensitive features, e.g., to predict the same acceptance rate for female and male applicants. Also, we do not study the reason why miscalibration occurs as in [4, 6], we study the metrics to evaluate it and a practical method to fix it.

2 Background

This paper focuses on calibrating probabilistic predictions for binary classification, i.e., to predict $\Pr(y = 1 \mid \boldsymbol{x})$, where $\boldsymbol{x}$ is the input and $y \in \{0, 1\}$ is the binary outcome. Consider we obtain the probabilistic prediction by a base model $\hat{p}(\boldsymbol{x}) = \sigma(l) = \sigma(f(\boldsymbol{x}))$ where $\sigma(\cdot)$ is the sigmoid function and $l = f(\boldsymbol{x})$ is the non-probabilistic output (also known as the *logit*) of the discriminative model. We denote a labeled dataset as $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{|\mathcal{D}|}$. The training / validation / test set are denoted by $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{valid}}, \mathcal{D}_{\text{test}}$, respectively. For the simplicity of notations, we will use $\hat{p}_i$ to denote $\hat{p}(\boldsymbol{x}_i)$.

2.1 Existing metrics for probabilistic predictions

Instance-level calibration error

A straight-forward way to measure miscalibration is to average the error on every single instance. For example, Negative Log-Likelihood (NLL), also known as the Log-loss, is formulated as

$$\text{LogLoss} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \big[-y_i \log \hat{p}_i - (1 - y_i) \log(1 - \hat{p}_i) \big]. \tag{1}$$

Similarly, the Brier score is the mean squared error over instances

$$\text{BrierScore} = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \big[y_i - \hat{p}_i \big]^2. \tag{2}$$

A drawback for these two metrics is that they cannot measure the bias on groups of instances. Thus by optimizing these objectives, the model can still give unfair predictions.

Probability-level calibration error

In many previous studies [4, 14, 6], the calibration error is formulated by partitioning the predictions into bins and summing up the errors over the bins. Formally, if we partition the $[0, 1)$ interval into K partitions where the k^{th} interval is $[a_k, b_k)$, the error is

$$\text{Prob-ECE} = \frac{1}{|\mathcal{D}|} \sum_{k=1}^{K} \Big| \sum_{i=1}^{|\mathcal{D}|} (y_i - \hat{p}_i) \mathcal{I}_{[\hat{p}_i \in [a_k, b_k)]} \Big|, \tag{3}$$

By minimizing this objective, the goal can be understood as "for every subset of data where the prediction is p, the actual averaged outcome should be around p".

However, we argue that this metric is too rough for evaluating predictions, and can be misleading for real-world applications. For example, one can get zero Prob-ECE by predicting a constant for all the samples. Therefore, this paper does not include Prob-ECE as an evaluation metric.

Non-calibration metrics

A number of non-calibration metrics can evaluate probabilistic predictions, such as the classification accuracy, the F-scores, and the Area under Curve (AUC). We will use AUC as the major non-calibration metric in this paper.

2.2 Existing calibration methods

Generally, data scientists observe miscalibration when testing the model on the validation set. Therefore, it is necessary to fix the error by learning a calibration function using the validation set.

Existing calibration methods can be categorized into non-parametric and parametric methods based on the mapping function. Non-parametric methods includes binning methods [18, 14] and Isotonic Regression [1, 16]. The idea is to partition the raw prediction into bins. Each bin is assigned with the averaged outcomes of instances in this bin using the validation set. If the partitions are predefined, the method is known as Histogram Binning [18]. However, the mapping function cannot keep the order of predictions, thus cannot be used for applications including advertising. Another common non-parametric method is Isotonic Regression [1], which requires the mapping function to be non-decreasing and is widely used in real-world industry [12, 3]. Parametric methods, on the other hand, use parameterized functions as the mapping function. The most common choice, Platt

Scaling [17, 16], is equivalent to a univariate logistic regression that transforms the model output (the logit) into calibrated probabilities. Because of the simple form, Platt scaling can be extended to multi-class classification for image and text classification [6]. However, the oversimplified mapping tends to under-fit the data and might be sub-optimal.

Other related works includes calibration with more detailed mapping functions or in different problem settings. To name some, [14] extends Histogram Binning to a Bayes ensemble; [6] extends Platt scaling to Temperature scaling; [15] extends Isotonic Regression to Bayesian, [10] generalizes it in an online setting, [3] uses it for calibrating click models; and [11] uses model ensemble to reduce the bias of predictions of deep learning models.

3 Measuring unfairness with field-level calibration error

We put forward the field-level calibration error as a new metric to measure the bias of probabilistic predictions in different subset of the dataset, which reflects the "unfairness" of the model.

3.1 Formulation of field-level calibration errors

Suppose that the model input is a $d+1$ dimensional vector $\boldsymbol{x} = (z, x_1, \ldots, x_d)$ including one specific categorical field $z \in \mathcal{Z}$ that the decision-maker especially cares about. Given that z is a categorical feature, we can partition the input space into $|\mathcal{Z}|$ disjoint subsets. For example, in the loan defaulter prediction task mentioned previously, this particular field is the "address state" feature with 51 levels, i.e., $z \in \mathcal{Z} = \{1, \ldots, 51\}$. Thus the data can be partitioned into 51 disjoint subsets.

Now we use these subsets to formulate field-level calibration errors. In particular, we formulate the field-level expected calibration error (Field-ECE) as

$$\text{Field-ECE} = \frac{1}{|\mathcal{D}|} \sum_{z=1}^{|\mathcal{Z}|} \Big| \sum_{i=1}^{|D|} (y_i - \hat{p}_i) \mathcal{I}_{[z_i=z]} \Big|, \tag{4}$$

which is straight-forward to understand: "for every subset of data categoried by the field z, the averaged prediction should agree with the averaged outcome". Therefore, if a set of predictions gets a large Field-ECE, it indicates that the model is biased on some part of the data.

Although this formulation has a similar form to Prob-ECE, there is a key difference. In Prob-ECE, the partition is determined by the prediction $\hat{p}$ itself, so the result can be misleading, e.g., it can get zero Prob-ECE by predicting a constant. But in our Field-ECE, the partition is determined by the input feature z, so the resultant metric can be consistent without being affected by the predictions.

Further, we can have the field-level relative calibration error formulated as the averaged rate of errors divided by the true outcomes,

$$\text{Field-RCE} = \frac{1}{|\mathcal{D}|} \sum_{z=1}^{|\mathcal{Z}|} N_z \frac{\big| \sum_{i=1}^{|\mathcal{D}|} (y_i - \hat{p}_i) \mathcal{I}_{[z_i=z]} \big|}{\sum_{i=1}^{|\mathcal{D}|} (y_i + \epsilon) \mathcal{I}_{[z_i=z]}}, \tag{5}$$

where N_z is the number of instance in each subset, i.e., $\sum_{z=1}^{|\mathcal{Z}|} N_z = |\mathcal{D}|$, and ϵ is a positive small number to prevent division by zero, e.g., $\epsilon = 0.01$.

Note that although our field-level calibration errors are formulated upon a categorical input field, they can be easily extended to non-categorical fields by discretizing them into disjoint subsets.

3.2 Observing miscalibration

Here we would like to show some observations to demonstrate the issue of miscalibration, especially field-level miscalibration. For all the tested tasks, we split the datasets into three parts: 60% for training, 20% for validation, and the other 20% for testing. We trained the base model, a two layered MLP, in two versions: Model-1 is trained on the training data $\mathcal{D}_{\text{train}}$, and Model-2 is updated incrementally over the validation set. For Model-1, since we can observe miscalibration on the validation set, we tested two existing calibration methods, Isotonic Regression and Platt Scaling. Such calibration pipelines are denoted by $\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$.

Observation 1: Neither a larger training set nor a higher AUC could indicate a smaller calibration error. Table 1 shows some results on the mentioned loan defaulter prediction task. From the results,

we see that Model-2 outperforms Model-1 in AUC significantly, which is easy to understand because it is trained over more data. Meanwhile, however, Model-2 suffered from higher calibration errors than Model-1, thus is not reliable to use.

Observation 2: Lower instance-level calibration error does not indicate lower field-level calibration error. Table 2 shows another example. It reports that Model-2 gets lower Log-loss and Brier score than the baseline calibration methods Isotonic Regression and Platt scaling. However, the Field-ECE and Field-RCE of Model-2 are larger, which means it is more biased than the calibrated ones.

Observation 3: Previous calibration methods are sometimes better than Model-2 in calibration metrics, but they are always worse than Model-2 in AUC. It can be observed from Table 1, 2, 3. Particuarlly, Isotonic Regression and Platt scaling did help calibration on the first two datasets, but failed on the third one. Moreover, we can see these calibration methods cannot help improving the AUC score over the base model, thus are always worse than Model-2 which learns from more data.

Observation 4: Improvement in Field-level calibration errors is easier to observe and more interpretable than in instance-level metrics. From the results, we see that the calibration methods often significantly reduce the field-level errors of Model-1, e.g., in Table 2, the relative reductions on Field-ECE and Field-RCE are around 20%. However, in the same Table, the relative reduction in Log-loss and Brier score given by calibration methods are no more than 0.2%, which is not significant. So the field-level metrics can be easier to use and to explain.

Table 1: Results on loan defaulter prediction

Method	Training data	Log-loss	Brier score	Field-ECE	Field-RCE	AUC
Base (Model-1)	$\mathcal{D}_{\text{train}}$	2.250	0.111	0.114	73.1%	0.821
Base (Model-2)	$\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{valid}}$	3.704	0.183	0.187	122.1%	0.936
Isotonic Reg.	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.302	0.086	0.025	16.1%	0.821
Platt Scaling	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.324	0.093	0.041	26.3%	0.821
Neural Calibration	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.059	0.013	0.025	16.0%	0.993

Table 2: Results on Criteo click-through rate prediction

Method	Training data	Log-loss	Brier score	Field-ECE	Field-RCE	AUC
Base (Model-1)	$\mathcal{D}_{\text{train}}$	0.4547	0.1474	0.0160	7.46%	0.7967
Base (Model-2)	$\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{valid}}$	0.4516	0.1464	0.0167	7.08%	0.8001
Isotonic Reg.	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.4539	0.1472	0.0134	6.09%	0.7967
Platt Scaling	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.4539	0.1472	0.0135	6.11%	0.7967
Neural Calibration	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.4513	0.1463	0.0094	4.59%	0.7996

Table 3: Results on Avazu click-through rate prediction

Method	Training data	Log-loss	Brier score	Field-ECE	Field-RCE	AUC
Base (Model-1)	$\mathcal{D}_{\text{train}}$	0.3920	0.1215	0.0139	12.88%	0.7442
Base (Model-2)	$\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{valid}}$	0.3875	0.1204	0.0120	11.17%	0.7496
Isotonic Reg.	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.3917	0.1216	0.0199	18.56%	0.7442
Platt Scaling	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.3921	0.1215	0.0165	15.18%	0.7442
Neural Calibration	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.3866	0.1202	0.0121	10.91%	0.7520

4 Neural Calibration

In light of these observations, we are motivated to design a new calibration method that can improve both calibration and non-calibration metrics. Our proposed solution is named Neural Calibration. It consists of two modules: a parametric probabilistic calibration module to transform the original model output into a calibrated one, and an auxiliary neural network to fully exploit the validation set. The basic formulation is written as follows

$$q(l, \boldsymbol{x}) = \sigma\big(\eta(l) + g(\boldsymbol{x})\big) \tag{6}$$

which is a sigmoid function of the sum of two terms: $\eta(l)$ transforms the logit $l = f(\boldsymbol{x})$ given by the original model to a calibrated one, and $g(\cdot)$ is an auxiliary neural network of all the other features.

Therefore, there are two functions $\eta(\cdot)$ and $g(\cdot)$ to learn, which are parametrized with trainable parameters ϕ and θ, respectively. They need to be trained simultaneously over the validation set.

The objective for training Neural Calibration is to minimize the Log-loss over the validation set, i.e.,

$$\min_{\phi,\theta} \frac{1}{|\mathcal{D}_{\text{valid}}|} \sum_{i=1}^{|\mathcal{D}_{\text{valid}}|} \big[-y_i \log q_i - (1-y_i)\log(1-q_i) \big]. \tag{7}$$

Therefore, Neural Calibration can be trained by stochastic gradient descent just as any other deep learning models. Also, it can extend to multi-class classification without any difficulty.

Now we introduce the detailed function structure of the two modules $\eta(\cdot)$ and $g(\cdot)$.

4.1 Isotonic Line-Plot Scaling (ILPS)

We are interested in finding a stronger parametric function $\eta(\cdot)$ that can achieve high performance. To enhance stronger fitting power, we borrow the spirit of binning from non-parametric methods. We first partition the real axis into several partitions with fixed splits $-M = a_1 < a_2 < \cdots < a_{K+1} = +M$ where M is a large number, and design the function as

$$\eta(l) = w_0 + \sum_{k=1}^{K} w_k (l - a_k) \mathcal{I}_{[a_k \le l < a_{k+1}]}, \tag{8}$$

where w_k are the coefficients to learn. To make it easier to optimize, we further re-parameterize it

$$\eta(l) = \sum_{k=1}^{K} [b_k + (b_{k+1} - b_k)(l - a_k)/(a_{k+1} - a_k)] \mathcal{I}_{[a_k \le l < a_{k+1}]}, \tag{9}$$

where b_k are the parameters. This mapping function looks just like a line-plot, as it connects (a_k, b_k) one by one. In practice, we set $K = 100$ and the splits s.t. $\sigma(a_k) = k/(K+1)$ for $k = 1, \ldots, K$.

Further, we would like to restrict the function to be isotonic (non-decreasing). To achieve this, we put a constraint on the parameters, i.e., $b_k \le b_{k+1}, \forall 1 \le k \le K$. In the actual implementation, the constraint is realized by the Lagrange method, i.e., adding a term on the loss function, so the overall optimization problem can be solved by gradient descent.

Now we get a novel parametric calibration mapping $\eta(\cdot)$ named Isotonic Line-Plot Scaling. We will show in the ablation study (Section 6.2) that this mapping can significantly outperform Platt scaling and be comparable or better than common non-parametric methods.

4.2 The auxiliary neural network for calibration

The previous part designed a univariate mapping from the original model output to a calibrated one. We further learn to use the whole validation data to train an auxiliary neural network.

The neural network $g(\boldsymbol{x})$ learns to fix the "unfairness" by using all necessary features from the validation set. Our intuition is simple: if we observe a field-level calibration error over the validation set on the specific field z, then we should find a way to fix it by learning a function of both the model output and the field z. Since in many cases z is a part of the input $\boldsymbol{x}$, we can directly learn a function of $\boldsymbol{x}$. Here we do not restrict the form of neural network for $g(\boldsymbol{x})$. For example, one can use a multi-layered perceptron for general data mining task, a convolution neural network for image inputs, or a recurrent neural network for sequential data.

5 Learning pipeline

In this section, we would like to describe the actual learning pipeline for real-world application that uses Neural Calibration to improve the performance. Suppose that we want to train a model with labeled binary classification data, and the target is to make reliable and fair probabilistic predictions during inference on the unseen data. We put forward the following pipeline of Neural Calibration:

Step 1. Split the dataset at hand into a training set $\mathcal{D}_{\text{train}}$ and a validation set $\mathcal{D}_{\text{valid}}$.

Step 2. Train a base model $f(\boldsymbol{x})$ over the training set $\mathcal{D}_{\text{train}}$. If necessary, select the model and tune the hyper-parameters by testing on the validation set.

Step 3. Train a Neural Calibration model $q(l, \boldsymbol{x}) = \sigma(\eta(l) + g(\boldsymbol{x}))$ over the validation set.

Step 4. Test on the hold-out data by predicting $\hat{p} = q(f(\boldsymbol{x}), \boldsymbol{x}) = \sigma(\eta(f(\boldsymbol{x})) + g(\boldsymbol{x}))$.

Here we give a brief explanation. To begin with, **Step 1** and **Step 2** are the common processes in machine learning. After having the model $f(\boldsymbol{x})$, one might observe instance-, probability-, or field-level miscalibration by examining the predictions on the validation set. We learn to calibrate the predictions in **Step 3**, which can be viewed as training a model with inputs (x_i, l_i) to fit the label y_i by minimizing the Log-loss as shown in Eq. (7). Finally, during inference on an unseen data sample with input $\boldsymbol{x}$, we can get the final prediction by two steps: first, compute the logit by the original model $l = f(\boldsymbol{x})$, and then compute the calibrated prediction by Neural Calibration $q = \sigma(\eta(l) + g(\boldsymbol{x}))$.

Comparison with existing learning pipelines

Generally, machine learning or data mining pipelines do not consider the miscalibration problem. That is they will directly make inference after **Step 1** and **Step 2**, which results in the Model-1 as mentioned in the previous section. In such case, the validation set is merely used for model selection.

Often, it is preferred to making full use of the labeled data at hand, including the validation set. So after training $f(\boldsymbol{x})$ on the training set, one can further update the model according to the validation set, which results in the Model-2 as mentioned. Such a training pipeline is useful especially when the data is arranged by time, because the validation set contains samples that are closer to the current time. However, this pipeline does not consider the calibration error.

The pipeline of calibration has the same procedure as ours. However, conventional calibration methods solely learn a mapping from uncalibrated outputs to calibrated predictions at **Step 3**. Our Neural Calibration is more flexible and powerful, because it can fully exploit the validation data.

6 Experiments

6.1 Experimental setup

Datasets: We tested the methods on five large scale real-world binary classification datasets.

1. Lending Club loan data[1], to predict whether an issued loan will be in default, with 2.26 million samples. The data is splitted by index. The field z is set as the "`address state`" with 51 levels.

2. Criteo display advertising data[2], to predict the probability that the user will click on a given ad. It consists of 45.8 million samples over 10 days and is splitted by index. The field z is set as an anonymous feature "`C11`" with 5683 levels.

3. Avazu click-through rate prediction data[3]. We used the data of first 10 days with 40.4 million samples, splitted by date. The field z is set as the "`site ID`" with 4737 levels.

4. Porto Seguro's safe driver prediction data[4], to predict if a driver will file an insurance claim next year. It has 0.6 million samples and is splitted by index. The field z is "`ps_ind_03`" with 12 levels.

5. Tencent click-through rate prediction, which is subsampled directly from the Tencent's online advertising stream. It consists of 100 million samples across 10 days, and is splitted by date. The field z is set as the advertisement ID with 0.1 million levels.

Tested models and training details: For all the datasets, the base models $f(\boldsymbol{x})$ and the net for calibration $g(\boldsymbol{x})$ are neural networks with the same structure: the input fields are first transformed into 256-dimensional dense embeddings respectively, which are concatenated together and followed by a multi-layer perceptron with two 200-dimensional ReLU layers. For each task, the base model is trained by the Adam optimizer [8] to minimize the Log-loss with a fixed learning rate of 0.001 for one epoch to get the Model-1. Next, we train the calibration methods on the validation set, or incrementally update Model-1 on the validation set for an epoch to get the Model-2. Specifically, Neural Calibration is also trained on the validation set for the same training steps and learning rates.

[2]https://www.kaggle.com/c/criteo-display-ad-challenge

[3]https://www.kaggle.com/c/avazu-ctr-prediction

[4]https://www.kaggle.com/c/porto-seguro-safe-driver-prediction

Compared baselines: We tested base models trained on the training set (Model-1) and on the joint of training and validation set (Model-2). We also tested common calibration methods, including Histogram Binning, Isotonic Regression and Platt scaling.

6.2 Experimental results

The results are shown in Table 1-5, where we leave the results of Histogram Binning to Table 6 due to the space limitation. From the four columns in the middle of the tables, we found Neural Calibration the best in all calibration metrics, which is significantly better than the tested baselines. From the rightmost column in the tables, we see that Neural Calibration can get significantly higher AUC than conventional calibration methods, but also be better or comparable compared with Model-2.

Table 4: Results on Porto Seguro's safe driver prediction

Method	Training data	Log-loss	Brier score	Field-ECE	Field-RCE	AUC
Base (Model-1)	$\mathcal{D}_{\text{train}}$	0.1552	0.0351	0.0133	28.55%	0.6244
Base (Model-2)	$\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{valid}}$	0.1538	0.0349	0.0064	13.90%	0.6245
Isotonic Reg.	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.1544	0.0349	0.0021	4.47%	0.6244
Platt Scaling	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.1532	0.0349	0.0020	4.30%	0.6244
Neural Calibration	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.1531	0.0349	0.0018	3.66%	0.6269

Table 5: Results on Tencent click through-rate prediction

Method	Training data	Log-loss	Brier score	Field-ECE	Field-RCE	AUC
Base (Model-1)	$\mathcal{D}_{\text{train}}$	0.1960	0.0522	0.0145	27.12%	0.7885
Base (Model-2)	$\mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{valid}}$	0.1953	0.0521	0.0128	24.58%	0.7908
Isotonic Reg.	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.1958	0.0522	0.0141	25.45%	0.7884
Platt Scaling	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.1958	0.0522	0.0142	25.72%	0.7885
Neural Calibration	$\mathcal{D}_{\text{train}} \to \mathcal{D}_{\text{valid}}$	0.1952	0.0521	0.0124	22.87%	0.7907

Table 6: Ablation study. Field-ECE is reported.

Method	Data-1	Data-2	Data-3	Data-4	Data-5
Histogram Bin.	0.026	0.0134	**0.0146**	0.0019	0.0142
Isotonic Reg.	**0.025**	0.0134	0.0199	0.0021	**0.0141**
Platt Scaling	0.041	0.0135	0.0165	0.0020	0.0142
ILPS	0.028	**0.0133**	**0.0146**	**0.0018**	**0.0141**

Ablation study: We tested solely the Isotonic Line-Plot Scaling to see if it is stronger than previous calibration methods which learn the univariate mapping functions. Table 6 shows that it constantly outperformed Platt scaling on all the datasets, and is comparable or better than non-parametric methods on dataset 2-5.

7 Conclusion

This paper studied the issue of miscalibration for probabilistic predictions of binary classification. We first put forward Field-level Calibration Error as a new class of metrics to measure miscalibration. It can report the bias and "unfairness" on specific subsets of data, which is often overlooked by common metrics. Then we observed that existing calibration methods cannot make full use of the labeled data, and we proposed a new method based on neural networks, named Neural Calibration, to address this issue. It consists of a novel parametric calibration mapping named Isotonic Line-Plot Scaling, and an auxiliary neural network. We tested our method on five large-scale datasets. By using the pipeline of Neural Calibration, we achieved significant improvements over conventional methods on both calibration metrics and non-calibration metrics simultaneously.

References

[1] Richard E Barlow, David J Bartholomew, James M Bremner, and H Daniel Brunk. Statistical inference under order restrictions: The theory and application of isotonic regression. Technical report, Wiley New York, 1972.

[2] Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana. Calibration of machine learning models. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pages 128–146. IGI Global, 2010.

[3] Alexey Borisov, Julia Kiseleva, Ilya Markov, and Maarten de Rijke. Calibration: A simple way to improve click models. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1503–1506. ACM, 2018.

[4] Morris H DeGroot and Stephen E Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22, 1983.

[5] Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. 2018.

[6] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1321–1330. JMLR. org, 2017.

[7] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 643–650. IEEE, 2011.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Kevin B Korb. Calibration and the evaluation of predictive learners. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 73–77, 1999.

[10] Wojciech Kotłowski, Wouter M Koolen, and Alan Malek. Online isotonic regression. In *Conference on Learning Theory*, pages 1165–1189, 2016.

[11] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[12] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2013.

[13] Aditya Krishna Menon and Robert C Williamson. The cost of fairness in binary classification. In *Conference on Fairness, Accountability and Transparency*, pages 107–118, 2018.

[14] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[15] Brian Neelon and David B Dunson. Bayesian isotonic regression and trend analysis. *Biometrics*, 60(2):398–406, 2004.

[16] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632. ACM, 2005.

[17] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[18] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699. ACM, 2002.

[19] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. *arXiv preprint arXiv:1507.05259*, 2015.