Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 3140301, 11 pages
https://doi.org/10.1155/2022/3140301

*Research Article*

# Meta-IP: An Imbalanced Processing Model Based on Meta-Learning for IT Project Extension Forecasts

**Min Li, Yumeng Zhang, Delong Han, and Mingle Zhou**

*Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250013, China*

Correspondence should be addressed to Delong Han; handl@sdas.org

Received 23 June 2022; Revised 9 August 2022; Accepted 18 August 2022; Published 20 September 2022

Academic Editor: Kuei-Hu Chang



With increasing developments in information technology, IT projects have received widespread attention. However, the success rate of large information technology projects is extremely low. Most current extension forecast models are designed based on a balanced number of samples and require a large amount of training data to achieve an acceptable prediction result. Constructing an effective extension forecast model with a small number of actual training samples and imbalanced data remains a challenge. This paper proposes a Meta-IP model based on transferable knowledge bases with few-shot learning and a model-agnostic meta-learning improvement algorithm to solve the problems of sample scarcity and data imbalance. The experimental results show that Meta-IP not only outperforms many current imbalance processing strategies but also resolves the problem of having too few samples. This provides a new direction for IT project extension forecasts.

## 1. Introduction

Information technology (IT) project schedule management is an important part of IT project construction. According to a report published by the Standish Group [1], small-scale projects are much more likely to succeed than larger ones, with large software projects having a success rate of only approximately 2%. Improving the on-time completion rates of a project while ensuring construction quality and meeting cost budgets poses a significant challenge to project managers. We investigated the performance of extension forecast models in imbalanced datasets by analyzing two key concepts: the degree of imbalance and the size of data. The results show that the imbalanced data and scarcity of data have a significant impact on the extension forecast performance of the IT project.

Although several models have been proposed for project extension forecasts, most of them use classical datasets with paired samples. The IT projects are mostly large-scale projects, and the successful completion of such projects is extremely rare. Therefore, researchers are faced with a dataset in which the number of uncompleted projects is significantly higher than the number of completed projects. We explored the predictive ability of several current extension forecast models in imbalanced datasets and found that methods designed for balanced datasets often fail to fit with imbalanced datasets. When the training data is heavily skewed, the training models tend to remember a small number of samples from minority classes.

Motivated by these limitations, this paper proposes a model called Meta-IP. This meta-learning-based model was designed to solve the data imbalance problem in the extension forecast of IT projects. In this paper, we propose a meta-learning-based imbalance classification model for small-sample IT projects, which combines transfer learning and meta-learning. We use transfer learning to solve the problem of sparse raw data samples and (model-agnostic meta-learning) MAML to process the data for imbalance. Not only does the model solve the problem of sample sparsity, but it also reduces overfitting in terms of imbalanced processing. Experimental results show the superiority of this algorithm compared to existing algorithms.

The main contributions of Meta-IP are as follows:

1. In the proposed model, transfer learning can avoid the limitations of sample scarcity and improve forecast accuracy and generalization performance in few-shot

conditions. Which provides the geometrical and algebraic basis for IT project extension forecasts.

2. Meta-IP makes full use of the excellent performance of MAML to solve data imbalances to reduce overfitting and greatly decrease the number of training samples required.
3. A meta-learning-based IT extension forecast model that combines transfer learning with meta-learning. As far as we know, this work is the first to incorporate transfer learning and meta-learning into IT project management, providing a rapid generalization of performance.

The rest of this paper is organized as follows: Section 2 introduces the imbalanced data processing methods and defines meta-learning and its related trends. The solutions for sample scarcity and imbalanced datasets and the proposed architecture are presented in Section 3. Section 4 discusses the dataset and provides an overview of selected classification models. Evaluation and analysis of the experimental results are discussed in Section 5. Section 6 offers a summary of the paper and a discussion of future work.

## 2. Related Work

*2.1. Imbalanced Data Processing Methods.* The current widely used methods for dealing with imbalanced data can be broadly classified into oversampling, undersampling, and classifier methods.

*2.1.1. Oversampling.* Oversampling mechanically replicates representatives from the minority class. However, this does not create new information about the minority class, and oversampling can result in overfitting. The Synthetic Minority Oversampling Technique (SMOTE) method addresses severe overfitting by developing individual pieces to create more diversity in a few classes of data. The generation of unique samples is achieved by linear interpolation of a few classes of existing observations. SMOTE is widely used, and some improvements have been made for generating additional training data to produce better decision bounds after training [2–4]. For example, support vector machine (SVM) SMOTE generates new minority examples along the bounds found by the support vector machine. However, SMOTE and its derived methods apply only to tabular data and not to high-dimensional data such as images.

*2.1.2. Undersampling.* Undersampling can reduce the number of the majority class samples, and it can prevent users from employing multiple classes of data to learn. Unlike oversampling, undersampling balances data by removing data samples randomly from the majority class to achieve class balance in the dataset. However, this entails the risk of losing critical data in most categories. Therefore, several researchers have proposed methods to select the majority of samples that can be removed without losing essential information in most categories. Wilson [5] proposed an Edit nearest neighbor (ENN) algorithm in which class data points that do not agree with the predictions of the K-nearest neighbor (K-NN) algorithm are removed. However, as with oversampling, undersampling is not suitable for high-dimensional data because the data being processed is often not informative enough. The Classifier model focuses on minority class samples during the training process. There are three types of classifier models: cost-sensitive learning [6], regularizers [7], and rescaling the classifier scores [8]. For example, cost-sensitive learning can change the loss in the number of minority class samples by changing the learning rate or by applying different weights to the training samples. Regularizers can either increase the number of generalization errors of many classes on a small course of data or impose constraints on the "equilibrium performance" measured on small balanced datasets.

However, none of these methods prevent fast overfitting of minority-class data and are suboptimal when applied to highly overparameterized models with short memory data [9].

*2.2. Meta-Learning.* Meta-learning is a method proposed to address the characteristics of traditional neural network models with poor generalization performance and poor adaptation to new kinds of tasks [10]. Meta-learning techniques use the meta-knowledge accumulated from historical tasks as a priori knowledge, then learn a small number of target samples to quickly master the new task. This technique improves the training method and training time and is highly adaptive to, and robust in, unknown scenarios. The corresponding machine meta-learning research is now broadly divided into five methods: metric-based learning methods, generalization-based learning initialization methods, optimizer-based methods, additional external storage methods, and data augmentation-based methods. Of these, the most progress has been made in learning generalization-based initialization, which has gradually become the backbone of the meta-learning field.

Meta-learning is currently widely used in natural language processing [11] and in computer vision fields [12] such as image recognition [13, 14], image classification [15, 16], object detection [17, 18], and recommender systems [19, 20]. In these scenarios, samples may be inherently scarce or difficult to collect, and annotated labels may be difficult to obtain. Moreover, the actual situation is often much more complex than the experimental setting. The best experimental accuracy results in the area of learning initialization based on strong generalizability come from the MAML [21] proposed by Finn et al. Subsequent improvements based on MAML include the relation network [22], residual network [23], and Bayesian experiments [24]. Due to the openness and flexibility of MAML, it can also be used for a series of gradient descent-based training models, including classification [25], regression [26], and reinforcement learning [27]. Several outstanding meta-learning algorithms have been derived from MAML in combination with other techniques, such as avoiding the problem of tricky gradient descent of high-dimensional parameters [28], realistic robot short-time learning motion [22], and face replacement techniques [29].

The benefits of MAML for imbalanced data processing stem from the fact that users can choose different sampling strategies for sampling support and querying data in the inner and outer loops. By complementing the two losses of the inner and outer loop, MAML can significantly improve performance beyond the baseline and reduce overfitting [30].

## 3. Meta-IP Model

To address the problems of sample scarcity and data imbalance in IT projects, this paper proposes Meta-IP, a data imbalance extension forecast model for few-shot IT projects based on meta-learning. The model solves the problems of sample scarcity and data imbalance through two modules: a transfer learning module with transferable knowledge, and a data imbalance processing module based on the MAML algorithm.

*3.1. Few-Shot Processing.* By sharing learning parameters with the new model, the transfer learning method with learning ability can transfer knowledge from the source task to the target task [31, 32]. This optimization method of transfer learning can greatly improve the generalization ability of the original model and the speed of new task modeling [33]. Due to the difficulty of collecting data from the small number of IT projects, most of the current deep learning-based IT project extension forecasts rely on laboratory simulation data rather than real-world situations. To address this issue, this paper will train on a source domain dataset and test on a target domain dataset. This will not only alleviate the problem of the small amount of real-world data but will also improve the accuracy of the model's conversion to real-world situations. In the module dealing with sample sparsity, we apply two datasets (the specific description of the dataset is presented in paper 4.1.): the source domain dataset from the prior knowledge base ($D_{\text{meta}}$) and the target domain dataset for a specific task ($D_{\text{novel}}$, IT project extension forecast) is given as follows:

$$\begin{aligned} D_{\text{meta}} &= \left\{\left(X_i, Y_i\right), Y_i \epsilon\, C_{\text{metal}}\right\}_{i=1}^{N_{\text{meta}}} \\ D_{\text{novel}} &= \left\{\left(X_i, Y_i\right), Y_i \epsilon\, C_{\text{novel}}\right\}_{i=1}^{N_{\text{novel}}}. \end{aligned} \tag{1}$$

$X_i$ in ($X_i$, $Y_i$) denotes the original feature vector of the i-th item, where $Y_i$ is the class label. $N_{\text{meta}}$ and $N_{\text{novel}}$ indicate the total number of observations of $D_{\text{meta}}$ and $D_{\text{novel}}$, respectively. The two-class labels $C_{\text{meta}}$ and $C_{\text{novel}}$ are disjointed.

In transfer learning, the task-specific dataset can be represented as $T = S \cup Q$. $T$ consists of a supported dataset $S$ and a small set of labeled query sets $Q$ from the same set of classes, so that the classifier can correctly distinguish the query set $Q$ depending on the support set. The meta-learner trains the data with a large amount of labeled data from the underlying dataset $D_{\text{meta}}$, which is then modified in the task-specific dataset $D_{\text{novel}}$ for a specific category. Only the classifier is trained, and the weight of the feature extractor is fixed. In this paper, all supporting datasets are represented by $S^{(k)}$, and, similarly, the query set is represented by $Q^{(k)}$. Items come from the same set of classes $C^{(k)}$, and there are three cases of subsets of $C$: $C_{\text{meta}}$, $C_{\text{novel}}$, and $C_{\text{meta}} \cup C_{\text{novel}}$.

The process of generating tasks from $D_{\text{novel}}$: $\left\{\widetilde{T} = \widetilde{S} \cup \widetilde{Q}\right\}_{k=1}^{T_{\text{novel}}}$ is carried out by performing novel by random sampling ($T_{\text{novel}}$ is the total number of $C_{\text{novel}}$ on the set of labels $\left\{\widetilde{C}^{(k)}\right\}_{k=1}^{T_{\text{novel}}}$ of sampling tasks on the dataset $D_{\text{novel}}$), followed by sampling instances in these classes. Compute the loss $L$ on the query set $Q$, conditional on $S$ and $D_{\text{meta}}$ for the classifier $f_\theta(X^{(k)}|\widetilde{S}(k), D_{\text{base}})$, and record.

$$\widetilde{Y}^{(k)} = f_\theta\left(X^{(k)}|\widetilde{S}(k), D_{\text{base}}\right). \tag{2}$$

Then, update model's parameters $\theta$:

$$a = \frac{1}{|\widetilde{Q}|}\sum_{k=1}^{|\widetilde{Q}|}\delta\left(Y^{(k)} = \widetilde{Y}^{(k)}\right). \tag{3}$$

The model trained by $D_{\text{novel}}$ now needs to learn to classify the task-specific query set $\widetilde{Q}^{(k)}$ and then adapt to $\widetilde{Q}^{(k)}$ using its support set $\widetilde{S}^{(k)}$. The support set in this module is used to calculate the prototype of data features, and the query set is used to train and improve the model's performance. The process is shown in Figure 1.

Because $C_{\text{meta}}$ and $C_{\text{novel}}$ are disjoint, the tasks of datasets $D_{\text{meta}}$ and $D_{\text{novel}}$ are not directly related and are linked together by some transferable knowledge so as to resolve the problem of sample scarcity.

*3.2. Imbalanced Data Processing.* The module dealing with data imbalance in this paper is based on the MAML algorithm and uses small loops of meta-learning to deal with data imbalance. The inner and outer loops of standard MAML algorithms have their own potential unique loss functions. In the inner loop, the model is fine-tuned by minimizing loss functions defined by a set of supporting data. The outer loop evaluates the fine-tuned model against a batch of queries from a query set from the same task.

The initial values of the weights are crucial for the speed of convergence because, for the same local minimum, different weights require different numbers of iterations to converge. The proposed Meta-IP method exploits the fact that meta-learning decouples the inner and outer loop loss functions, allowing different class balancing strategies for each position. We use the progress of the IT project to synthesize the training data, train, and draw X meta-training tasks $T = \{T_1, T_2 \ldots, T_X\}$ from the train dataset. Each training task contains a support set and a query set. Therefore, the support set $S_i = \{\nu_{i1}, \nu_{i2} \ldots, \nu_{is} = (x_{i1}, y_{i1}), (x_{i2}, y_{i2}) \ldots, (x_{is}, y_{is})\}$. The key steps in Meta-IP, including task sampling, meta-training, and meta-testing, are then implemented, as shown in Figures 2 and 3.

At the same time, we can use rebalanced data in the query set to guide the algorithm in the training process to achieve high accuracy for class-balanced data. This allows us to combine the benefits of training with imbalanced data

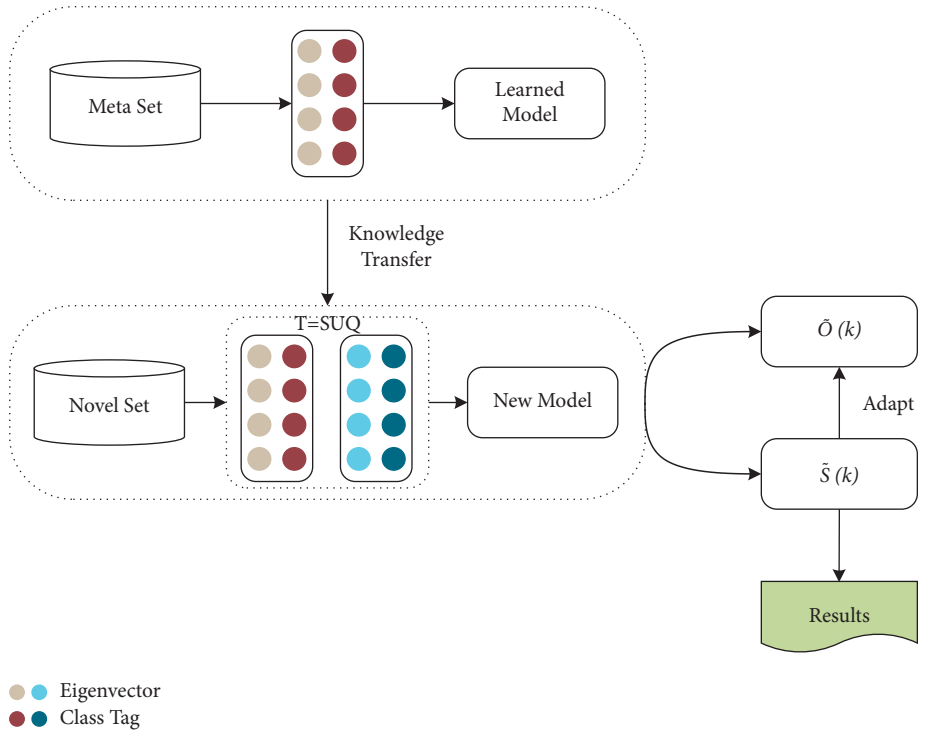


Figure 1: Few-shot processing flow chart.

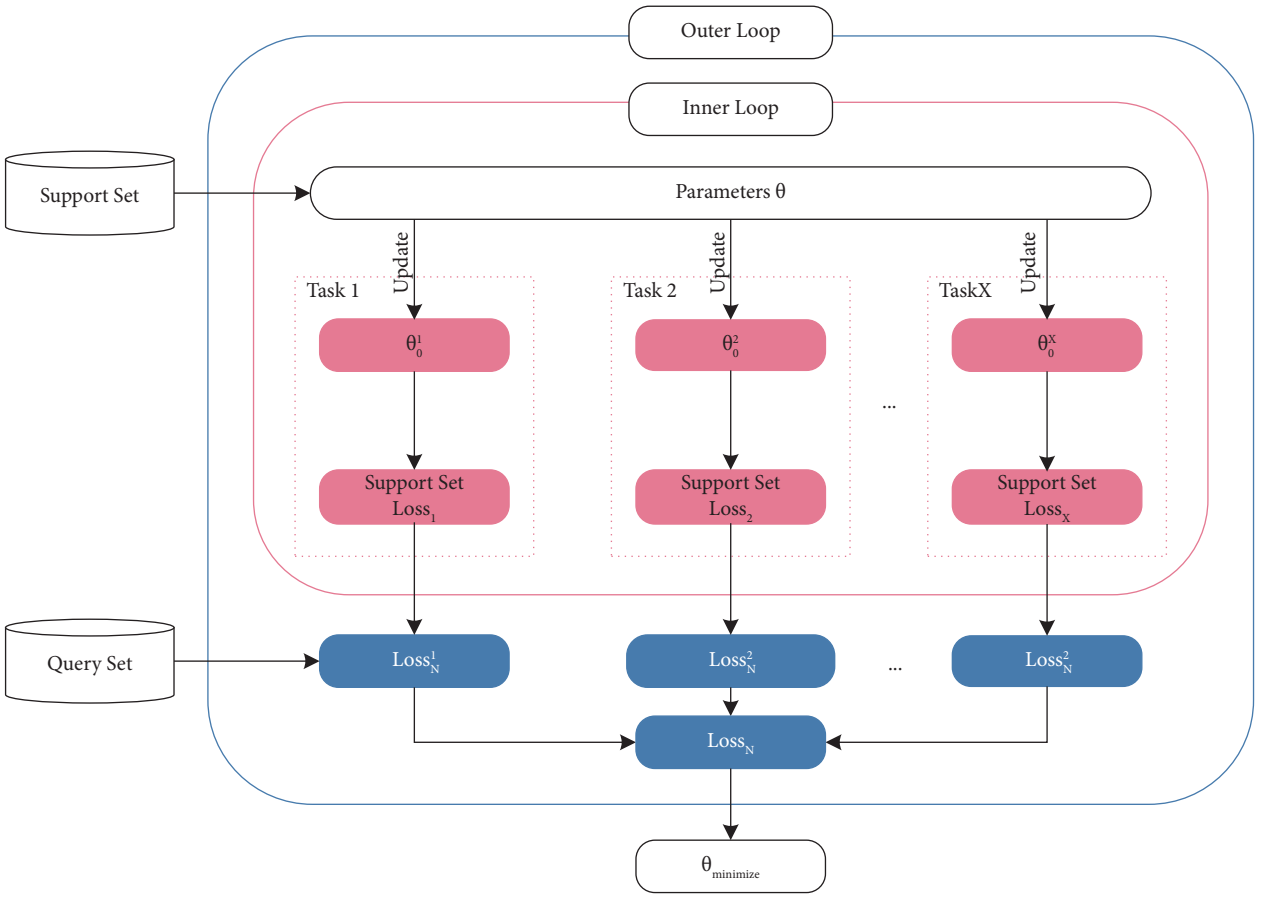


Figure 2: Training pipeline of the Meta-IP.

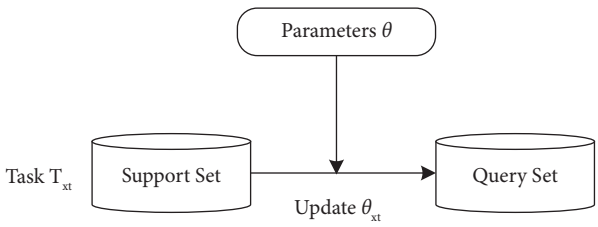


Figure 3: A testing pipeline of the Meta-IP.

(preventing overfitting) and the benefits of training with balanced data (a few classes are not ignored). To formalize the metabolic balancing algorithm, consider a neural network $f$ with parameters $\theta$, which maps samples to predictions. $\theta$ denotes the initial parameter, $\alpha$ denotes the external cyclic learning rate, $\gamma$ denotes the internal cyclic learning rate, and $M$ and $N$ denote the sample imbalance support batches and balanced query batches, respectively. In the inner loop of the metabolic balancing training procedure, through the calculation of adaptive parameters using the gradient descent method, the loss of sample batches $M$ is minimized to obtain the new parameters.

$$\theta_0 = \theta - \alpha \nabla_\theta L_M \left( f_\theta \right). \tag{4}$$

Then, in the outer loop, the model is evaluated on a new balanced sample N based on the new parameter $\theta_0$, and the loss $\text{Loss}_N$ is calculated based on these predictions.

$$\text{Loss}_N = \text{Loss}_N + L_N f\left(\theta'\right) + \beta L_M f\left(\theta\right). \tag{5}$$

We accumulate the external loss over multiple support and query batches, fine-tune each support batch separately, and finally update $\theta_{\text{minimize}}$ to minimize the query loss.

$$\theta_{\text{minimize}} = \theta - \gamma \nabla_\theta \text{Loss}_N. \tag{6}$$

The advantage of Meta-IP is that it allows us to choose different sampling strategies for sampling support and query sets in inner and outer loops. By using complementary sampling strategies for these two losses, the performance beyond the baseline can be significantly improved. The losses for each dataset are unfolded against each other using naive Bayesian, Bagging, Boosting, SMOTE, and SVM.

It is known that balanced datasets are not an efficient way to achieve fairness for applications such as default prediction, extension forecast, and facial recognition [34]. However, despite our training improvements on imbalanced data, our goal is to obtain high accuracy, not fairness.

## 4. Experiment

An extended completion of an IT project is a tabular binary classification problem, in which the positive samples are projects that are extended and the negative samples are projects that are completed on schedule. In this task, we trained a feedforward neural network with five fully connected layers using binary cross-entropy loss.

*4.1. Dataset Description.* In this experiment, there are two types of datasets: the source domain dataset, which contains the simulation data in a simulated environment, and the target domain dataset for task-specific learning, which is derived from all IT projects in Province S, China, from 2015 to 2019, and contains the real management metrics of these IT projects in the actual progress management. The dataset is divided into five parts (each part represents a year) describing the time period from year 1 (2015) to year 5 (2019). The class labels of the dataset ("0" for completed and "1" for extended) were determined based on the completion status of all projects collected in 2020. The dimension of the input characteristics (including class labels) is 14, of which 9 attributes are numeric attributes and 5 attributes are categorical attributes. The details of the dataset can be viewed in Table 1.

There are two main features in the dataset: nondeterministic factors and deterministic factors. Nondeterministic factors include development team maturity, acceptance criteria, process maturity, activity resource requirements, and human resource allocation. Deterministic factors include size, complexity, software development budget, software personnel monthly labor budget, construction management fee budget, consulting fee budget, bidding fee budget, project supervision fee budget, and construction time [35].

*4.2. Dataset Pre-Processing.* Data preprocessing is crucial to the model. In the data preprocessing stage, the numerical features are standardized and normalized using the preprocessing package in the scikit-learn (sklearn) module. The primary data preprocessing techniques such as virtual coding, data normalization, and correlation analysis are used to process the original dataset to obtain valid classification results. Numerical features are normalized by removing the mean and unit variance. The classification features are then optimized using virtual coding and polynomial processing. Virtual coding transforms a continuous input variable into multiple elements, while polynomial processing increases the diversity of features.

*4.3. Evaluation Metrics.* Three evaluation metrics were used in this study, the area under curve (AUC) [36], the geometric mean (G-mean) [37], and the balance of accuracy (BACC) [38]. Each of these metrics reflects the performance of the model with a different focus. The rules for calculating these metrics are given as follows:

Because we usually care about positive instances of classification results, we used the F indicator to measure the classifier's performance. TP means true samples correctly identified, or true positives; FN means false samples incorrectly identified as true samples, or false negatives; TN means false samples correctly identified, or true negatives; and FP means true samples incorrectly identified as false samples, or false positives.

Table 1: Details of the imbalanced dataset.

| Data-set | Total number of samples | Negative sample | Positive sample | Digital features | Classification features | Total number features |
|---|---|---|---|---|---|---|
| 1st-year | 499 | 425 | 74 | 9 | 5 | 14 |
| 2nd-year | 338 | 313 | 25 | 9 | 5 | 14 |
| 3rd-year | 590 | 554 | 36 | 9 | 5 | 14 |
| 4th-year | 706 | 664 | 42 | 9 | 5 | 14 |
| 5th-year | 442 | 428 | 14 | 9 | 5 | 14 |

$$\text{Sensitivity} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity} = \text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{7}$$

$$F_\beta = \frac{\left(1 + \beta^2\right) \times \text{Sensitivity} \times \text{Specificity}}{\beta \times \text{Sensitivity} + \text{Specificity}}.$$

AUC is a classical evaluation index in classification problems, defined as the area under the receiver operating characteristic (ROC) curve and between that and the coordinate axis. AUC is defined by the following given equation:

$$AUC = \frac{\sum_{ins_i \in \text{positiveclass}} \text{rank}_{ins_i} - M \times (M + 1)/2}{M \times N}, \tag{8}$$

where $\text{rank}_{insi}$ represents the serial number of the i-th sample (the probability score is ranked from smallest to most significant in the rank position). M, N are the number of positive samples and the number of negative examples, $\sum$insi positive class rankins i, respectively. The equation adds only the ordinal numbers of positive samples.

G-mean is a composite metric widely used to measure the accuracy of imbalanced learning models. The G-mean is defined as following:

$$G - \text{Mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}}. \tag{9}$$

Finally, we used BACC rates to evaluate the results. BACC more accurately reflects the actual performance of the classifier in imbalanced learning. Based on the mixed matrix, its definition is

$$BACC = \frac{\text{TPR} + \text{TNR}}{2}. \tag{10}$$

*4.4. Experiment Setup.* To improve the reliability of the experiments, reduce the occurrence of chance events, and offset the randomness of sampling, each group of experiments was trained for 50 epochs and assessed using the optimal results. The training was repeated 20 times.

An IT project extension forecast is a tabular binary classification problem, with positive samples representing the completed projects and negative samples the uncompleted projects. We used a two-stage approach based on data analysis and the concept of the center of gravity [39]. The neural network was trained on a set of unbalanced data of the centers of mass from an a priori knowledge base (the first stage involved finding local minima close to the global minimum, and successful training was indicated when the local minima were relative to the global minimum). Moreover, at this stage, we trained a pretrained model of a five-layer feedforward network with the number of channels per layer as shown in Table 2. In the second training stage, a specific IT project extension forecast was used, in which the classifier was modified to distinguish between correctly labeling the dataset and a few shots. The weights of the initialized neural network were learned using the complete data in the second training stage.

This model has four hidden layers, followed by a binary classification output for predicting item extensions. We used a probability of a 0.5 dropout rate after the second level. Each layer of the neural network (except for the final layer) had a ReLU activation function that was used to increase the speed of gradient descent. Meta-IP, like the other sampling methods, uniformly used the Adam optimizer with a learning rate of 0.001 and trained for 100 epochs. We split the original dataset into a training and a test dataset at a ratio of 8 : 2 and used a batch size of 24. $\theta_0$ was calculated as described in (4), with a $\gamma$ constant of 0.01 and a loss accumulation constant $\beta$ of 0. The accumulated loss reached 80 meta-steps before updating $\theta$. We used a support batch size of 24 and a query batch size of 16. In the training dataset Train, in which the proportion of extension forecast items was equal to or less than 20% of the total number of samples. The prediction performance decreased significantly compared to the balanced data. Although the performance of all classifiers was affected by the imbalanced dataset, the degradation in prediction performance became more pronounced as the dataset imbalance increased.

## 5. Results and Discussion

To address the problems of sample scarcity and data imbalance in IT projects, this paper proposes Meta-IP, a data imbalance extension forecast model for few-shot IT projects based on meta-learning. The model solves the problems of sample scarcity and data imbalance through two modules: a transfer learning module with transferable knowledge and a data imbalance processing module based on the MAML algorithm.

*5.1. Validation of Transfer Learning Capability.* The transfer learning capability of Meta-IP performed better than models trained on unprocessed data. The transfer learning process was divided into two stages. Three experiments were conducted to verify the influence of IT project samples on $D_{\text{meta}}$,

Table 2: The number of channels per layer of a feedforward neural network.

| Neural Network | Input layer | First layer | Second layer | Third layer | Fourth layer | Output layer |
|---|---|---|---|---|---|---|
| The number of channels | 14 | 16 | 24 | 20 | 24 | 1 |

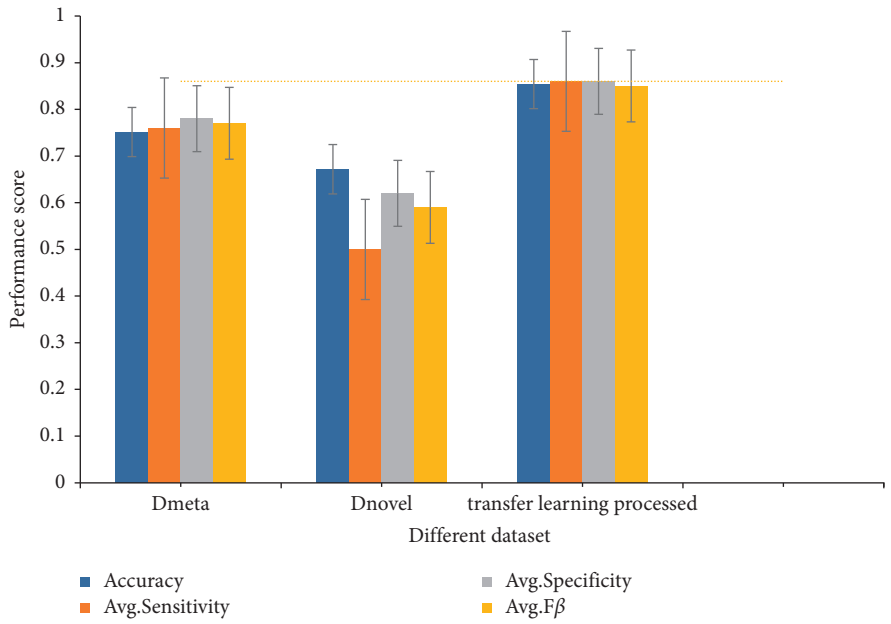


Figure 4: Performance of a transfer learning model under the imbalanced dataset.

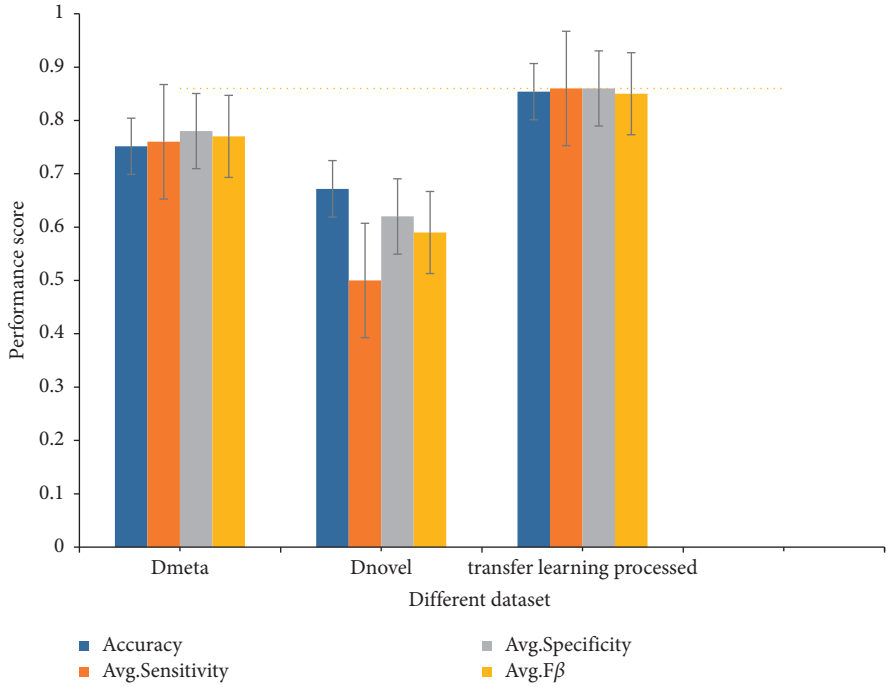


Figure 5: Performance of a transfer learning model under the balanced dataset.

IT project samples on $D_{\text{novel}}$, and the transfer learning model processed dataset on the forecast accuracy of different IT projects under balanced data and imbalanced data. Figures 4 and 5 show the results of the transfer learning in terms of accuracy, on the balanced and imbalanced datasets of IT projects.

Figures 4 and 5 clearly show that the accuracy of the extended forecast decreases when the dataset is not subjected to imbalanced treatment. Figure 4 shows that the transfer learning model proposed in this paper has the highest values in terms of accuracy, sensitivity, specificity,

Table 3: Comparison of AUC and BACC across models trained with various sampling methods on the IT project extension forecast tasks. Bold figures reflect the row maximum.

| Model | 1st-year dataset | | 2nd-year dataset | | 3rd-year dataset | | 4th-year dataset | | 5th-year dataset | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AUC | BACC | AUC | BACC | AUC | BACC | AUC | BACC | AUC | BACC |
| Naive Bayesian | 0.967 | 84.3 ± 0.11 | 0.958 | 83.2 ± 0.03 | 0.962 | 85.7 ± 0.10 | 0.74 | 87.2 ± 0.08 | 0.963 | 85.6 ± 0.09 |
| Bagging | 0.943 | 73.2 ± 0.05 | 0.957 | 77.2 ± 0.02 | 0.963 | 73.9 ± 0.04 | 0.962 | 78.3 ± 0.07 | 0.972 | 75.2 ± 0.09 |
| SMOTE | 0.952 | 70.8 ± 1.30 | 0.952 | 66.8 ± 0.90 | 0.979 | 71.2 ± 1.20 | 0.955 | 69.3 ± 0.80 | 0.957 | 69.5 ± 1.10 |
| SVM | 0.921 | 79.2 ± 1.00 | 0.934 | 83.4 ± 0.90 | 0.958 | 81.3 ± 0.80 | 0.963 | 80.2 ± 1.10 | 0.954 | 82.7 ± 1.20 |
| SMEOTE+ | 0.955 | 72.4 ± 0.02 | 0.945 | 74.3 ± 0.03 | 0.935 | 72.7 ± 0.01 | 0.937 | 77.3 ± 0.05 | 0.943 | 74.8 ± 0.07 |
| SMOTE+ | 0.946 | 73.5 ± 0.03 | 0.927 | 72.7 ± 0.01 | 0.947 | 73.5 ± 0.04 | 0.953 | 76.7 ± 0.03 | 0.938 | 72.8 ± 0.05 |
| Meta-IP | **0.975** | **89.7 ± 0.05** | **0.965** | **90.1 ± 0.06** | **0.988** | **91.2 ± 0.02** | **0.985** | **89.3 ± 0.03** | **0.976** | **91.8 ± 0.07** |

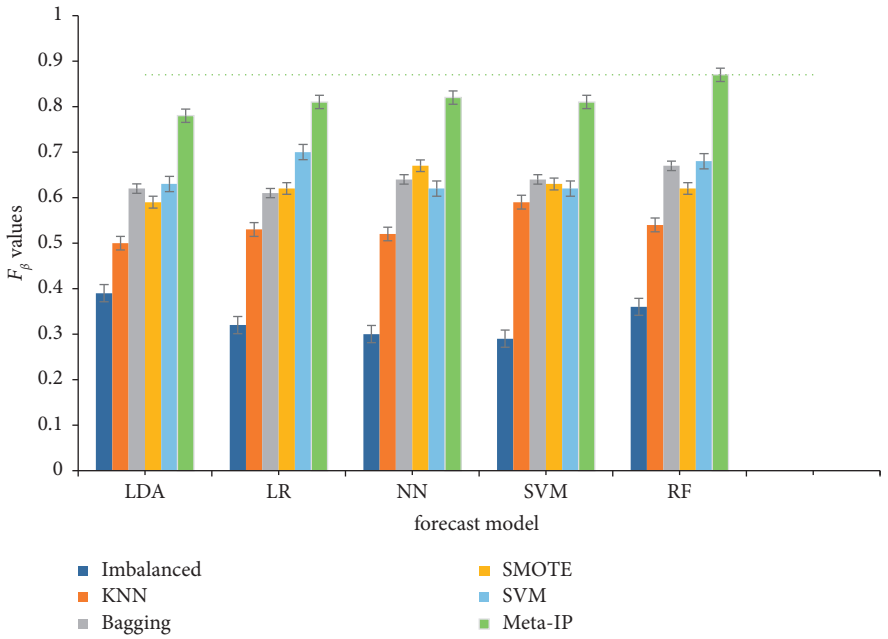


Figure 6: Comparison of $(F)_\beta$ across models trained with various sampling methods on the IT project extension forecast tasks.

Table 4: The G-mean values were achieved with Meta-IP on balanced training sets by sample.

| Dataset | Meta-IP |
|---|---|
| 1st-year | 88.3 |
| 2nd-year | 88.2 |
| 3rd-year | 89.2 |
| 4th-year | 89.8 |
| 5th-year | 89.2 |

Table 5: The G-mean values were achieved with forecast models on balanced and imbalanced training datasets (1st-year) by various sampling methods. Bold figures reflect the row maximum.

| Forecast model | Imbalanced | Naive Bayesian | Bagging | SMOTE | SVM |
|---|---|---|---|---|---|
| LDA | 32.9 | **73.4** | 83.1 | 76.5 | 79.6 |
| LR | **33.6** | 72.1 | 83.5 | 76.3 | 80.1 |
| NN | 32.8 | 71.5 | 84.6 | 77.2 | 81.2 |
| SVM | 32.1 | 72.2 | 82.3 | 75.2 | 79.5 |
| RF | 31.7 | 71.1 | **84.8** | **77.3** | **81.5** |

Table 6: The G-mean values were achieved with forecast models on balanced and imbalanced training datasets (2nd-year) by various sampling methods. Bold figures reflect the row maximum.

| Forecast model | Imbalanced | Naive bayesian | Bagging | SMOTE | SVM |
|---|---|---|---|---|---|
| LDA | 39.1 | 72.3 | 83.9 | 75.4 | 82.1 |
| LR | 40.9 | 70.5 | 83.6 | 77.6 | 83.4 |
| NN | 42.3 | 70.9 | 84.4 | **77.8** | 81.9 |
| SVM | 41.3 | 73.3 | **85.9** | 76.5 | 82.5 |
| RF | **42.7** | **72.8** | 85.6 | 75.2 | **83.2** |

and F$\beta$ values. This shows that the transfer learning model has the capability to solve the problem of sample scarcity.

*5.2. Comparison of Meta-IP with Other Imbalanced Data Processing Methods.* To compare the imbalanced data processing performance of Meta-IP, we compare it with the traditional and latest imbalanced data processing models. For all imbalanced data processing models, we use open-source implementations. The first five models are compared

Table 7: The G-mean values were achieved with forecast models on balanced and imbalanced training datasets (3rd-year) by various sampling methods. Bold figures reflect the row maximum.

| Forecast model | Imbalanced | Naive bayesian | Bagging | SMOTE | SVM |
|---|---|---|---|---|---|
| LDA | 37.3 | 75.6 | 85.5 | **78.5** | 83.2 |
| LR | 35.5 | 76.3 | 84.6 | 76.5 | **84.3** |
| NN | 36.9 | 77.4 | 85.2 | 75.6 | 82.6 |
| SVM | **37.8** | **78.3** | **86.5** | 77.6 | 82.2 |
| RF | 35.5 | 78.2 | 86.3 | 76.9 | 83.9 |

Table 8: The G-mean values were achieved with forecast models on balanced and imbalanced training datasets (4th-year) by various sampling methods. The bold figures reflect the row.

| Forecast model | Imbalanced | Naive Bayesian | Bagging | SMOTE | SVM |
|---|---|---|---|---|---|
| LDA | **38.7** | 75.4 | 82.3 | 76.2 | 82.1 |
| LR | 36.6 | 75.6 | 84.5 | **77.9** | 83.2 |
| NN | 37.8 | **76.8** | 83.7 | 77.4 | **84.2** |
| SVM | 38.1 | 74.9 | 82.1 | 76.1 | 83.6 |
| RF | 38.2 | 76.7 | **84.6** | 71.1 | 82.0 |

Table 9: The G-mean values were achieved with forecast models on balanced and imbalanced training datasets (5th-year) by various sampling methods. Bold figures reflect the row maximum.

| Forecast model | Imbalanced | Naive Bayesian | Bagging | SMOTE | SVM |
|---|---|---|---|---|---|
| LDA | 42.1 | 75.7 | 81.5 | 72.7 | 75.4 |
| LR | **44.3** | 74.8 | 82.6 | 71.5 | 75.2 |
| NN | 43.5 | 75.6 | 83.7 | 73.6 | 76.5 |
| SVM | 41.9 | **76.4** | 82.2 | 73.9 | 76.5 |
| RF | 42.3 | 74.3 | **83.8** | **74.6** | **77.8** |

as traditional and the latest imbalanced data processing model. Since the data sets are private, the second two models [40, 41] are the latest imbalanced data processing models that use similar datasets as in this article.

We ran it 30 times on an IT dataset project to reduce the occurrence of chance events and then averaged and validated the BACC. The data processing methods were all run with the same default parameter values. When the classes of training data were imbalanced, the overfitting generated during the training further affected the training results. The models overfit disproportionately to a few types of data, leading to a significant performance gap between the majority and minority classes at training time. For an IT project with imbalanced data, we analyzed the overfitting behavior of three models: the naive Bayesian model, the oversampling training model, and the Meta-IP training model.

From the statistical results, we found that naive Bayesian sampling performed better than the standard-based sampling method, as the randomization algorithm itself is uncertain. The results show that even though the other sampling techniques performed well, the model trained with Meta-IP (an imbalanced data processing module) had the best results, with a significant increase in AUC. Table 3 lists the average AUC and BACC for the five years of cross-validation. The table indicates that the imbalanced data processing performance of the Meta-IP (an imbalanced data processing module) algorithm had a higher average BACC than the other algorithms.

The results also showed that Meta-IP (the imbalanced data processing module) achieved a significant performance compared to the traditional and latest imbalanced data processing models. The naive Bayesian model did not actually learn to recognize the patterns contained in the minority class and was blindly input to classify into the majority class. Therefore, we concluded that training routines that meaningfully deal with class imbalance have better performance than the oversampling and naive Bayesian models in a few classes and also perform better than undersampling models in most classes. During training, Meta-IP (the imbalanced data processing module) reduced the overfitting of a few classes of data and, therefore, achieved higher overall test accuracy than the other models.

*5.3. Comparison of Meta-IP with Other Imbalanced Data Processing Methods.* In this paper, we take $\beta = 1$. The F$\beta$ of the five algorithms in the data after processing by different sampling methods are shown in Figure 6.

For each algorithm, the larger the imbalance ratio of the samples, the smaller the F$\beta$ values. As can be seen from Figure 6, Meta-IP has the largest F$\beta$ values for the same extension forecast method, SVM the second largest, and imbalanced data the smallest. In terms of F$\beta$ values, Meta-IP outperforms the other algorithms.

In previous studies, when the dataset showed an imbalanced distribution, the predictive performance of the minority group (sensitivity) decreased and that of the majority class (specificity) increased. In addition, the classification boundaries of the majority class tended to breach the classification boundaries of the minority class, thereby biasing the classification toward the majority class [42]. Our experimental study examined the impact of different imbalanced data treatments on different extension forecast models, and we determined these metrics through a fixed default threshold that defined the boundary value that classified the sample into extended and nonextended items. Table 4 shows the results obtained for the Meta-IP built on different samples and Tables 5–9 show the results obtained for Table 6 forecast models built on different Table 7 samples. We contextualized the results by analyzing the G-mean values that measured the overall prediction according to a ratio of sensitivity and specificity. The following tables Table 8 shows that the G-mean takes into account Table 9 trade-off between sensitivity and specificity. With all methods running under the same conditions, Meta-IP

achieved a higher overall test accuracy than the other models in Table 4.

The results of the comparisons showed that Meta-IP performs well in handling data imbalances in few-shot extension forecasts of IT projects, preventing over-adaptation to most classes and overfitting. In the case of data scarcity, the standard error of Meta-IP is significantly reduced, and the prediction performance is improved dramatically.

## 6. Conclusions

We have presented a new model dubbed Meta-IP for IT project extension forecasts. Meta-IP solves the problem of transfer learning by directly generating task-specific learner parameters, thereby reducing the difficulty of training on new datasets and then dealing with imbalanced data by MAML. Finally, the experimental results show that our proposed model achieves other advanced methods. Therefore, this study can serve as a heuristic development for researchers to design few-shot postponement prediction models for IT projects with imbalanced datasets and provide new solutions to overcome the imbalanced data problem.

However, Meta-IP lacks the capacity for theoretical and in-depth analysis of the specific selection for generating parameters. There is still a need to develop few-shot meta-learning algorithms with good generalization abilities and fewer labeled samples. Determining how to construct better meta-learners, more effective task-based meta-learners, cross-domain few-shot meta-learners, and multidomain few-shot meta-learners should be the focus of future research.

Despite Meta-IP's performance in handling imbalanced data, our results should be interpreted with caution. The results reflect only the information contained in the data and the characteristics of the input data; further research should be more complex, as most models are validated under experimental conditions that are not representative of real-world scenarios with imbalanced datasets. We speculate that even if research is being conducted to design more complex models, it will not prevent the postponement of a real-world IT project because the complexity of the data makes the model's performance suboptimal, and the data problem cannot be truly addressed from the ground up.

## Data Availability

The data used to support the findings of this study have not been made available because the data are sensitive.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Authors' Contributions

The conceptualization was done by M.L. and Y.Z.; validation was done by D.H and M.Z.; formal analysis was performed by Y.Z.; investigation was done by D.H.; writing—original draft preparation was done by M.L.; writing—review and editing was done by Y.Z. and D.H.; and funding acquisition was done by M.L. All authors have read and agreed to the published version of the manuscript.

## Acknowledgments

The authors thank LetPub (https://www.letpub.com) for its linguistic assistance during the preparation of this manuscript. This research was funded by the Key R&D plan of Shandong Province (Soft Science Project) (2021RZA01016) and Plan of Youth Innovation Team Development of Colleges and Universities in Shandong Province (SD2019-161).

## References

[1] Standish group 2015 chaos report, "Standish group 2015 chaos report," 2015, https://www.infoq.com/articles/standish-chaos-2015/.

[2] K. E. Bennin, J. Keung, P. Phannachitta, A. Monden, Mensah, and S. Mahakil, "Diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction," *IEEE Transactions on Software Engineering*, vol. 44, p. 1, 2018.

[3] A. Roy, R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "A Study on Combining Dynamic Selection and Data Preprocessing for Imbalance Learning," *Neurocomputing*, vol. 286, pp. 179–192, 2018.

[4] J. Mao, Z. Gao, Y. Wu, and M. S. Alouini, "Over-sampling codebook-based hybrid minimum sum-mean- square-error precoding for millimeter-wave 3D-MIMO," *IEEE Wireless Communications Letters*, vol. 7, no. 6, pp. 938–941, 2018.

[5] D. L. Wilson and L. Dennis, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Transactions on Systems Man and Cybernetics*, vol. 2, no. 3, pp. 408–421, 1972.

[6] Y. Cui, M. L. Jia, T. Y. Lin, Y. Song, S. Belongie, and I. Soc, "Comp class-balanced loss based on effective number of samples. Ieee Comp," in *Proceedings of the 32nd IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9260–9269, China, June, 2019.

[7] S. Sangalli, E. Erdil, A. Hoetker, O. Donati, and E. Konukoglu, "Constrained optimization for training deep neural networks under class imbalance," 2021, https://arxiv.org/abs/2102.12894.

[8] R. Chan, M. Rottmann, F. Hüger, P. Schlicht, and H. Gottschalk, "Application of decision rules for handling class imbalance in semantic segmentation," 2019, https://arxiv.org/abs/1901.08394.

[9] N. Rout, D. Mishra, and M. K. Mallick, "Handling imbalanced data: a survey. International conference on advances in soft computing," *Intelligent Systems and Applications*, ASISA, vol. 628, pp. 431–443, 2018.

[10] X. X. Li, Z. Sun, J. H. Xue, and Z. Y. Ma, "A concise review of recent few-shot meta-learning methods," *Neurocomputing*, vol. 456, pp. 463–468, 2021.

[11] X. Wu, D. Sahoo, and S. Hoi, "Meta-RCNN: meta learning for few-shot object detection," in *Proceedings of the 28th ACM International Conference on Multimedia ACM*, Virtual Event, China, October, 2021.

[12] J. Choi, J. Kwon, and K. M. Lee, "Deep meta learning for real-time target-aware visual tracking," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea, October, 2019.

[13] D. ADas and C. Lee, "A two-stage approach to few-shot learning for image recognition," *IEEE Transactions on Image Processing*, vol. 29, p. 1, 2019.

[14] J. Narwariya, P. Malhotra, L. Vig, G. Shroff, and V. Tv, "Meta-learning for few-shot time series classification," 2019, https://arxiv.org/abs/1909.07155.

[15] Q. Wang, G. Wang, G. Kou, M. Zang, and H. Wang, "Application of meta-learning framework based on multiple-capsule intelligent neural systems in image classification," *Neural Processing Letters*, vol. 53, no. 4, pp. 2581–2602, 2021.

[16] X. Zhong, C. Gu, W. Huang, L. Li, and C. W. Lin, "Complementing representation deficiency in few-shot image classification: a meta-learning approach," in *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy, January, 2020.

[17] Q. Liu, X. Zhang, Y. Liu, K. Huo, W. Jiang, and X. Li, "Multi-polarization fusion few-shot HRRP target recognition based on meta-learning framework," *IEEE Sensors Journal*, vol. 21, Article ID 18085, 2021.

[18] T. Hassan, M. Shafay, S. Akçay et al., "Meta-transfer learning driven tensor-shot detector for the autonomous localization and recognition of concealed baggage threats," *Sensors*, vol. 20, no. 22, p. 6450, 2020.

[19] M. Saveski, A. Mantrach, and Acm, *Item Cold-Start Recommendations: Learning Local Collective Embeddings*, in *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys)*, pp. 89–96, Foster City, Silicon Valley, CA, USA, August, 2014.

[20] H. Wang and Y. M. L Zhao, "ML2E: meta-learning embedding ensemble for cold-start recommendation," *IEEE Access*, vol. 8, Article ID 165757, 2020.

[21] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, August, 2017.

[22] A. Nagabandi, I. Clavera, S. Liu et al., "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning," 2018, https://arxiv.org/abs/1803.11347.

[23] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun, and Ieee, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, Nevada, USA, June, 2016.

[24] T. Kim, J. Yoon, O. Dia, S. Kim, Y. Bengio, and S. Ahn, "Bayesian model-agnostic meta-learning," 2018, https://arxiv.org/abs/1806.03836.

[25] E. Grant, C. Finn, S. Levine, T. Darrell, and T. Griffiths, "Recasting gradient-based meta-learning as hierarchical bayes," 2018, https://arxiv.org/abs/1801.08930.

[26] Z. Li, F. Zhou, C. Fei, and L. Hang, "Meta-S. G. D.: Learning to learn quickly for few-shot learning," 2017, https://arxiv.org/abs/1707.09835.

[27] M. Al-Shedivat, T. Bansal, Y. Burda, I. Sutskever, I. Mordatch, and P. Abbeel, "Continuous adaptation via meta-learning in nonstationary and competitive environments," 2017, https://arxiv.org/abs/1710.03641.

[28] A. A. Rusu, D. Rao, J. Sygnowski et al., "Meta-learning with latent embedding optimization," 2018, https://arxiv.org/abs/1807.05960.

[29] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," *IEEE*, 2018, https://arxiv.org/abs/1905.08233.

[30] A. Bansal, M. Goldblum, and V. Cherepanova, "MetaBalance: high-performance neural networks for class-imbalanced," 2021, https://arxiv.org/abs/2106.09643.

[31] A. Gupta, C. Devin, Y. X. Liu, P. Abbeel, and S. Levine, "Learning invariant feature spaces to transfer skills with reinforcement learning," 2017, https://arxiv.org/abs/1703.02949.

[32] S. Wang, D. Wang, D. Kong, J. Wang, W. Li, and S. Zhou, "Few-shot rolling bearing fault diagnosis with metric-based meta learning," *Sensors*, vol. 20, no. 22, p. 6437, 2020.

[33] R. Raileanu, M. Goldstein, A. Szlam, and R. Fergus, "Fast adaptation via policy-dynamics value functions," 2020, https://arxiv.org/abs/2007.02879.

[34] V. Albiero, K. Zhang, and K. W. Bowyer, "How does gender balance in training data affect face recognition accuracy?" in *Proceedings of the 2020 IEEE International Joint Conference on Biometrics (IJCB)*, Houston, TX, USA, September, 2020.

[35] E. Markopoulos, "An IT project management methodology generator based on an agile project management process framework," *10th Int Conf on Appl Human Factors and Ergon (AHFE)/AHFE Int Conf Human Factors in Artificial Intelligence and Social Comp/AHFE Int Conf on Human Factors, Software, Serv and Syst Engn/AHFE Int Conf of Human Factors in Energy*, vol. 965, pp. 421–431, 2020.

[36] J. M. Lobo, A. Jimenez-Valverde, and R. Real, "AUC: a misleading measure of the performance of predictive distribution models," *Global Ecology and Biogeography*, vol. 17, no. 2, pp. 145–151, 2008.

[37] M. Kubat, "Adressing the curse of imbalanced training sets: one-sided selection," in *Proceedings of the International Conference on Machine Learning*, Atlanta GA USA, June, 2016.

[38] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *Proceedings of the 2010 20th International Conference on Pattern Recognition*, DBLP, Istanbul, Turkey, August, 2010.

[39] A. Saadi and H. Belhadef, "Towards an optimal set of initial weights for a deep neural network archi- tecture," *Neural Network World*, vol. 29, no. 6, pp. 403–426, 2019.

[40] Y. J. Jang, I. B. Jeong, Y. K. Cho, and Y. Ahn, "Predicting business failure of construction contractors using long short-term memory recurrent neural network," *Journal of Construction Engineering and Management*, vol. 145, no. 11, p. 145, 2019.

[41] N. Basurto, A. Jimenez, S. Bayraktar, and A. Herrero, "Improving the prediction of project success in the telecom sector by means of advanced data balancing," *Journal of Construction Engineering and Management*, vol. 0, 2022.

[42] M. J. Kim, D. Ki Kang, and H. B. Kim, "Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1074–1082, 2015.