

Article

Small-Sample Data Pricing Based on Data Augmentation and Meta-Learning

Junxin Shen ¹, Yi Yang ¹ and Fanghao Xiao ^{2,*}

¹ Faculty of Management and Economics, Kunming University of Science and Technology, Kunming 650504, China; shenjx@kust.edu.cn (J.S.); yangyi068@stu.kust.edu.cn (Y.Y.)

² Marxist College, Xiamen Institute of Technology, Xiamen 361024, China

* Correspondence: xiaofanghao@xit.edu.cn

Abstract: Data trading platforms play a crucial role in facilitating data circulation and promoting the sustainable allocation of data resources. Establishing a transparent, fair, and efficient pricing mechanism is key to ensuring the long-term stability and development of such platforms. However, these platforms face challenges in pricing due to the small sample problem, as traditional machine learning methods typically rely on large amounts of data. To address this issue, this paper proposes a data resource pricing model that combines WGAN-GP data augmentation and the Reptile algorithm. Data augmentation generates related datasets to increase sample size, enhancing the renewability of data resources, while meta-learning transfers knowledge across tasks, improving the model's ability to quickly adapt to new tasks and efficiently utilize resources. Validation using actual trading data from the data trading platform shows that the proposed model accurately predicts data resource prices under small-sample conditions, outperforming other models. This study addresses the limitations of existing pricing methods in small-sample scenarios, providing a sustainable pricing solution for small-sample data resources and improving the accuracy and long-term stability of data pricing in the market.

Keywords: small sample; data price; reptile algorithm; WGAN-GP



Citation: Shen, J.; Yang, Y.; Xiao, F. Small-Sample Data Pricing Based on Data Augmentation and Meta-Learning. *Electronics* **2024**, *13*, 4150. <https://doi.org/10.3390/electronics13214150>

Academic Editor: Guanfeng Liu

Received: 24 September 2024

Revised: 18 October 2024

Accepted: 21 October 2024

Published: 22 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the digital deconomy, data integrates with conventional production factors to enhance efficiency and drive growth. Effective data pricing is crucial for the marketization of data elements [1], yet privacy concerns [2] and data silos [3] necessitate transactions to unlock value. The market transaction of data elements serves as the foundation for the value extraction and creation of data elements, effectively improving resource allocation and production efficiency [4]. The bilateral market facilitated by data trading platforms overcomes the high matching costs and fragmented demand–supply dynamics of unilateral markets [5]. By consolidating, presenting, and disseminating information, these platforms expand consumer access to product details [6]. For effective marketization, data trading platforms must address operational barriers to improve the allocation of data elements [7].

Constructing an acceptable, efficient, and fair data pricing mechanism is paramount to enhancing data circulation and sharing [8]. Such a mechanism ensures platform stability [9], attracts more participants, enriches offerings, and scales up the data market. However, inconsistent data resource valuation and pricing approaches across platforms, coupled with inadequate standards and norms, lead to incomplete content, non-standard formats, and inconsistent calibrations in data transactions [10].

The complexity of data transactions in multilateral markets is exacerbated by incomplete rights, pricing, and trading systems, resulting in low trading volumes on data element platforms [11]. Moreover, the limited diversity and quality of data hinder pattern revelation [12]. Traditional statistical or rule-based methods are inadequate for

high-dimensional relationships, while machine learning offers robust solutions but requires large datasets, posing challenges in small-sample scenarios due to overfitting, poor generalization, and instability.

To address small-sample issues, we propose a combined approach using WGAN-GP for generating synthetic data and the Reptile algorithm for meta-learning. This synergy not only expands a dataset but also enables effective cross-task learning, improving model generalizability under sparse data conditions. Our method, validated on Guoxin Youyi's data trading platform, outperforms alternatives in price prediction accuracy.

The main contributions of this paper are as follows:

A comprehensive framework for data resource pricing, based on data enhancement methods and meta-learning technology, is proposed and validated. WGAN-GP is employed to generate high-quality virtual samples, diversifying and expanding small-sample datasets while alleviating model training difficulties. The algorithm enhances the model's adaptability and robustness, promotes rapid convergence and efficient learning, and improves its ability to adapt to the fluctuating market demands of new datasets and data trading platforms. This innovative integration of technologies not only overcomes the model training difficulties caused by small-sample data but also significantly improves data pricing accuracy, addresses the shortcomings of existing pricing methods under small-sample conditions, and offers an effective solution to the technical challenges faced by data trading platforms.

2. Literature Review

2.1. Pricing Issues on Data Trading Platforms

Policymakers are urged to refine asset evaluation and pricing guidelines, establishing a revenue distribution and incentive framework pertinent to data resources [13]. Nonetheless, China's data element marketization is still evolving, facing a confluence of challenges that result in insufficient data demand, depressed trading activities, and subpar market capabilities on current data trading platforms [7].

The functionality of some data trading platforms is relatively simple, and high transaction costs result in insufficient trading volume [14]. Inadequate legal frameworks and regulations in the data trade sector mean that many online data sources remain untapped due to unclear property rights, leaving unmet market needs [15]. Privacy and security concerns deter data owners from sharing their private data, creating numerous data silos that prevent full data utilization [12].

The lack of clear data property rights and legal guarantees for privacy and security protection has led to low trust among data owners [12,16], making them cautious about transactions and resulting in limited tradable data on data trading platforms. A lack of standardized interoperability between platforms exacerbates the situation; data formats, interfaces, and interaction methods vary widely, impeding data sharing and trading across platforms [15]. Legal, regulatory [16], and interoperability obstacles collectively [15] constrain the quantity of data available on trading platforms.

2.2. The Small Sample Problem

In practical applications, it is often expensive or difficult to obtain a large number of high-quality annotated data [12], resulting in many scenarios where only small-scale datasets can be used for model training [17]. Traditional machine learning models face the following major challenges when dealing with small-sample problems: First, the model has difficulty learning effectively from limited samples, leading to an increased risk of overfitting [18]. Second, data scarcity can lead to insufficient generalization of models to accurately predict new data [19]. Finally, small-sample data usually cannot provide enough diversity and representativeness to affect the stability and predictive performance of the model [20,21].

Current research on small-sample learning primarily explores three dimensions: data, models, and optimization algorithms. Data-centric approaches augment experience

through samples and feature information. Model-focused strategies reduce the hypothesis space through structure and parameter design. Optimization-oriented methods enhance the probability of finding the optimal hypothesis by modifying search strategies within a given hypothesis space. Small data volumes predominantly impact the accuracy and stability of feature selection during the process. While algorithmic improvements to hyper-parameter settings have limitations [22], data augmentation techniques offer an effective means to expand datasets and improve feature selection precision and stability [23].

Data augmentation diversifies datasets without additional sampling costs, preventing overfitting and enhancing the value of small datasets [24]. GANs train two adversarial models concurrently; a generator network creates artificial data by capturing the original data distribution from noise, while a discriminator network learns to distinguish generated from real data [25].

However, standard GANs suffer from training instability, mode collapse, and difficulty in evaluation [26,27], hindering the generator's ability to learn diverse data distributions [27]. Gulrajani et al. introduced WGAN-GP, substituting the KL divergence with the Wasserstein distance and applying gradient penalties to enforce Lipschitz constraints on the loss function [28]. WGAN-GP's robust critic models the underlying distribution by penalizing the model when the gradient norm deviates from the target norm [29].

Most existing small-sample learning methods rely on single data augmentation or transfer learning techniques. Integrating different small-sample learning methods could be beneficial, addressing issues from both data and model perspectives [30]. Meta-learning, exemplified by model-agnostic MAML [31] and Reptile [32] algorithms, enables quick adaptation to new tasks through cross-task training. MAML, despite its successes, requires computationally expensive second-order derivatives for updates. The Reptile algorithm simplifies meta-updates with a more straightforward formula, improving MAML's computational efficiency [33–35]. Its effectiveness has been demonstrated across various problems.

3. Pricing Model for Small-Sample Data

3.1. WGAN-GP

3.1.1. Generator–Discriminator Design

GAN simultaneously trains a generator, which aims to capture data distribution, and a discriminator, which estimates the probability that a given sample comes from real data [36]. It applies the concept of the zero-sum game from game theory, viewing the generator and discriminator as two competing players. Through a dynamic game process between them, the underlying distribution of real data is analyzed, and a model capable of generating data resembling the real data is constructed. The generator's goal is to learn the real data distribution and produce samples that closely resemble real ones to evade the discriminator's detection [17,37]. During training, the generator learns to improve the quality of generated data such that it becomes indistinguishable from real data for the discriminator. It does this by minimizing the Wasserstein distance between real and generated data distributions. On the other hand, the discriminator's goal is to determine whether the input data are real or generated, and its output is the probability that the data are real. The discriminator maximizes the Wasserstein distance between real and generated data by assigning lower scores to synthetic samples and higher scores to real samples. The generator and discriminator, both neural networks, are alternately trained to enhance their abilities until the generated samples are indistinguishable from the real ones. The general GAN objective function is shown in the following equation:

$$\min_G \max_D V(D, G) = E_{x \sim P_{\text{data}(x)}} [\ln D(x)] + E_{z \sim P_z(z)} [\ln(1 - D(G(z)))] \quad (1)$$

The generator is denoted as $\min_G V(D, G)$, and the discriminator is denoted as $\max_D V(D, G)$. During the GAN training process, the discriminator model D is first optimized to maximize the probability that data come from real samples while minimizing the loss for the generator model G . Through alternating operations, G is fixed, and D is

optimized to maximize the discriminator's probability. Then, D is fixed, and G is optimized to minimize the discriminator's probability, leading to a stable Nash equilibrium.

When the input data are sampled from the real sample x , the goal of the discriminator D is to make the output probability $D(x)$ approach 1. The optimization formula is as follows:

$$\max_D V(D, G) = E_{x \sim P_{\text{data}(x)}} [\ln D(x)] + E_{z \sim P_{z(z)}} [\ln(1 - G(D(z)))] \quad (2)$$

Next, G is optimized so that the generated sample is closer to label 1 of the real sample. When the input comes from the generated sample $G(z)$, the goal of D causes $D(G(z))$ to approach 0. The optimization formula is as follows:

$$\min_G V(D, G) = E_{z \sim P_{z(z)}} [\ln(1 - G(D(z)))] \quad (3)$$

In conclusion, our aim is to maximize $V(D)$ and minimize $V(G)$. After iterative training, the network parameters of D and G are constantly updated. When the Nash equilibrium is reached, D cannot accurately distinguish between generated samples and real samples. However, GAN networks commonly face two issues: gradient disappearance during training and the confusion of optimization objectives. This leads to unstable gradients, reduced diversity and accuracy in generated samples, and, ultimately, model collapse.

3.1.2. Structure of the WGAN-GP Model

WGAN-GP (Wasserstein Generative Adversarial Network with Gradient Penalty) is a variant of GAN designed to improve training stability through the use of the Wasserstein distance and gradient penalties. Its primary goal is to create new and unique data that capture the internal features and distributions inherent in the original data.

(1) Wasserstein Distance

During GAN training, issues such as mode collapse and instability can arise [25–27]; the generated samples become too unconstrained and difficult to control, making it hard to focus on specific target samples [37]. To address the problems inherent in GAN model training, Arjovsky et al. introduced the Wasserstein GAN (WGAN), which utilizes the distance between the probability distributions of real and generated samples rather than the discriminator's objective function [38]. The Wasserstein distance is employed to measure the dissimilarity between the generated and real distributions. Compared to the JS divergence or KL divergence used in traditional GANs, the Wasserstein distance is more stable, especially when there is little overlap between distributions.

The target loss function for WGAN is given by the following formula:

$$\text{Loss}_{\text{critic}} = \max_{w \in W} E_{(X \sim p(r))} [f_w(X)] - E_{(z \sim P(Z))} [f_w(g_\theta(Z))] \quad (4)$$

where $\text{Loss}_{\text{critic}}$ stands for discriminator loss, g_θ is the optimal parameter, and if g_θ satisfies the Lipschitz condition, then K is the Lipschitz constant of the function f :

$$|f(X_1) - f(X_2)| \leq K|x_1 - x_2| \quad (5)$$

(2) Gradient Penalty

Although WGAN can achieve stable training by leveraging the Wasserstein distance, it still encounters challenges related to generating subpar samples and failing to converge to the target function. These difficulties stem from the Lipschitz constraint on the WGAN critic's loss. In WGAN, to enforce the Lipschitz constraint, the discriminator's weights are clipped to lie within a compact space $[-r, r]$, where r is a constant. The critic attempts to distinguish between real and generated samples, concentrating its parameter distribution at

the two extremes of maximum and minimum values, i.e., r and $-r$. Consequently, WGAN often becomes stuck in poor regimes and struggles to learn.

To alleviate this issue, Gulrajani et al. proposed WGAN-GP, which incorporates gradient penalties to augment the clipping of weights. The primary aim of the gradient penalty term is to ensure that the discriminator is Lipschitz continuous. Lipschitz continuity is a mathematical property indicating that the slope of a function has an upper bound across its domain. By introducing gradient penalties, WGAN-GP can better meet the optimization objective of the Wasserstein distance during training, enhancing the stability of both the generator and the discriminator.

The loss function for WGAN-GP is defined as follows:

$$L = E_{z \sim P_z(z)} [D(G_\theta(Z))] - E_{x \sim P_r(x)} [D(x)] + \lambda E_{\hat{x} \sim P_{\hat{x}}(\hat{x})} [\| \nabla_{\hat{x}} D_{\theta_D}(\hat{x}) \|] \quad (6)$$

where P_r is the real data distribution, and P_g is the generated distribution.

Gulrajani et al. demonstrated that WGAN-GP outperforms alternative methods such as Least Squares GAN (LSGAN), Deep Convolutional GAN (DCGAN), and vanilla WGAN in terms of model performance and stability [29]. The overall architecture of the WGAN-GP model in this paper includes a preprocessing module, a generator, and a CNN-based discriminator, as depicted in Figure 1.

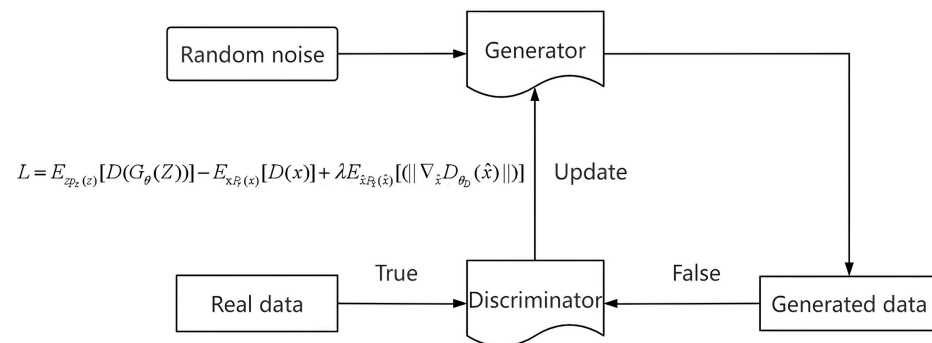


Figure 1. Architecture diagram of WGAN-GP model.

3.2. Reptile Algorithm

Implementation of the Reptile Algorithm

The core idea of meta-learning is “learn to learn”, and a meta-model is trained through cross-task learning so that the model can quickly adapt to new tasks through a small number of gradient updates. MAML [32] and Reptile [33] are two classical methods for solving small-sample classification problems using meta-learners. Finn et al. [32] proposed model-independent meta-learning (MAML), which trains a set of initialization parameters with few update steps so that new tasks can be quickly adapted with only a small amount of labeled data. Similar to MAML, the key goal of the Reptile algorithm is to learn a set of initial model parameters, θ , that can be quickly adapted to new tasks with a small number of update steps. At the same time, the complicated second derivative calculation is avoided, which makes the algorithm more efficient.

The Reptile algorithm performs two steps alternately: an inner loop update and an outer loop update. First, a pricing task is randomly sampled, and the basic pricing model is trained using the training set for this task. Multi-step Stochastic Gradient Descent (SGD) is performed k times on the current task to obtain temporary parameters. Then, the model parameters are updated based on the difference between the previous model parameters and the results of multi-step SGD. The above steps are repeated iteratively to update the model parameters. Through this process, the model can learn features and knowledge related to the current task. The detailed process for each step is as follows:

First of all, the model initialization θ is set to random values, which will be adjusted iteratively during the training process. Then, a task $\mathcal{T}_i$ is randomly sampled from the task

distribution $p(\mathcal{T})$. For each task $\mathcal{T}_i$, the model performs k steps of gradient descent on the training data of this task to update the model parameters. The gradient step size k is determined by the batch training size B_M and the training step E_M , and the accuracy and efficiency of the network are weighed by adjusting the step size. This step is akin to independently training the model for each task.

The inner loop update process is as follows: start with the model parameters θ , and compute the loss $\mathcal{L}_{\mathcal{T}_i}(\theta)$ on the task $\mathcal{T}_i$'s training data. Update the model parameters θ via gradient descent as follows:

$$\theta_i' = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(\theta) \quad (7)$$

Here, α is the learning rate for the inner loop, and $\nabla_{\theta} \mathcal{L}_{\mathcal{T}_i}(\theta)$ is the gradient of the loss with respect to the parameters. After several gradient updates, we obtain task-specific parameters θ_i' .

After completing the inner loop optimization for task $\mathcal{T}_i$, the model's global parameters θ are updated. The Reptile algorithm updates the initialization parameters as follows:

$$\theta = \theta + \beta (\theta_i' - \theta) \quad (8)$$

where β is the meta-learning rate. This formula moves the initialization parameters θ toward the task-specific parameters θ_i' obtained from task $\mathcal{T}_i$. Repeating this process across different tasks gradually refines the model initialization.

As shown in Figure 2 below, the algorithm samples tasks from the task distribution several times and repeats the steps of inner loop update and outer loop update so that the parameters of the model learn together on multiple tasks so that it can quickly adapt to new tasks.

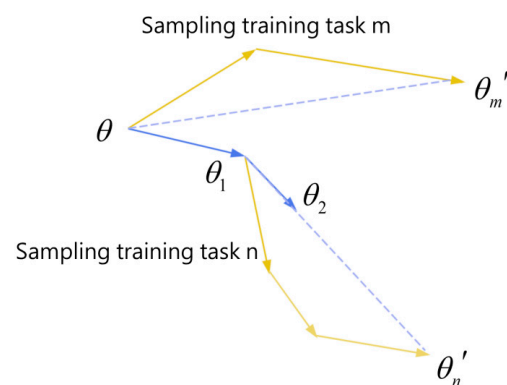


Figure 2. Illustration of the Reptile algorithm steps.

3.3. Design of Model Structure

Multilayer Perceptrons (MLPs) are used to capture correlations among variables. Previous research has shown that MLP models can reduce unnecessary weight connections by introducing sparse or local connectivity and implementing weight-sharing strategies. These approaches significantly decrease the number of parameters, increase the effective data size, and help prevent overfitting. In this study, the WGAN-GP (Wasserstein Generative Adversarial Network with Gradient Penalty) network architecture adopted is as outlined in Tables 1 and 2 below.

Table 1. Structure of the WGAN-GP generator.

Structure	Input Layer	Output Layer	Number of Nodes	Activation Function
Input Layer	77	128	128	ReLU
Hidden Layer 1	128	256	128	ReLU
Hidden Layer 2	256	256	256	ReLU
Output Layer	256	1	256	tanh

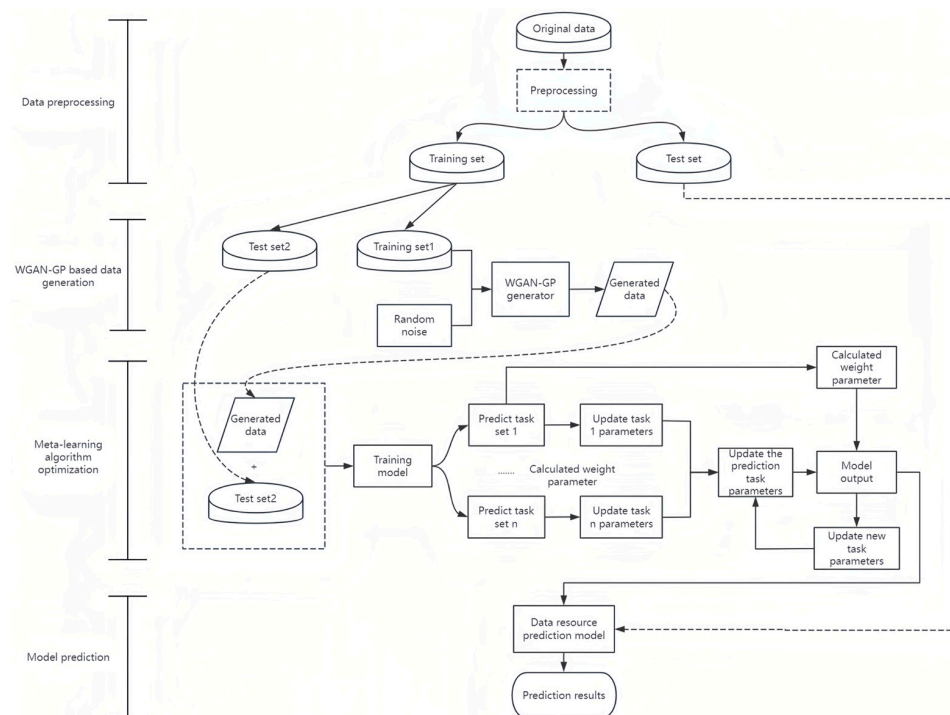
Table 2. Structure of the WGAN-GP discriminator.

Structure	Input Layer	Output Layer	Activation Function	Negative Activation Slope
Input Layer	77	64	LeakyReLU	0.01
Hidden Layer 1	77	64	LeakyReLU	0.01
Output Layer	32	1		

The model employs a Multilayer Perceptron (MLP) structure with three fully connected layers, including batch normalization layers and ReLU activation functions. Batch normalization speeds up convergence and stabilizes training, while ReLU solves the vanishing gradient issue, boosting the model's ability to represent complex data. L2 regularization is applied to control the model's complexity and prevent overfitting.

The model also employs the Reptile algorithm, which trains across multiple tasks. After each iteration, the model updates its parameters based on small changes, helping it adapt quickly to new tasks. This setup reduces training costs for new tasks while improving adaptability. Batch normalization and L2 regularization further enhance the model's generalization, reducing the risk of overfitting. The MLP structure's flexibility allows it to be used across various tasks by adjusting the size and number of hidden layers. The Reptile algorithm's iterative training with fewer steps lowers computational costs without sacrificing performance.

The specific process of the data resource pricing prediction method, which combines data augmentation using WGAN-GP and the Reptile algorithm, is shown in Figure 3. First, the processed existing data resource features and prices are input into the generator to enrich the training dataset and enhance the model's generalization ability. Next, the data resource features generated by the generator are combined with real data resource features and input into the discriminator for evaluation. If the discriminator can correctly distinguish between the real and generated data resource, the generator is encouraged to produce more realistic data. If the discriminator fails to distinguish them, the final data resource features are obtained as output.

**Figure 3.** Research flow chart.

The data resource pricing prediction method combines WGAN-GP data augmentation with the Reptile algorithm. First, processed data resource features and prices are fed into the WGAN-GP generator, enriching the dataset and improving generalization. The generated features are then combined with real data and evaluated by the discriminator. If the discriminator correctly identifies real versus generated data, the generator is adjusted to produce more realistic data. If it fails, the final data resource features are output.

4. Experiment

4.1. Dataset and Experimental Design

The data used in this study come from the Guoxin Youyi data trading platform. This paper uses web crawler technology to obtain transaction data information of multiple industries from this website. The dataset comprises 2179 entries across various industry categories, including industrial economy, healthcare, transportation and geography, financial credit, precision marketing, scientific research technology, enterprise management, life services, industry monitoring, and public opinion monitoring. The diversity of industries ensures the broad scope and representativeness of the data sources. This diversity allows the pricing model to be more adaptable and accurate across different application scenarios. It includes data overall score, data scarcity, consistency, application value, structured degree, quantity, information redundancy, integrity, timeliness, sales, data category, data label, and 13 other evaluation indicators, as well as data resource prices. The platform manages scores according to the dual dimensions of rule attributes and application scope to improve the accuracy of matching. By using the network analysis method, the dynamic weight distribution scoring model is formed, and the scoring is reliable. By integrating data from various fields, the pricing model can learn the characteristics and market dynamics of each industry, enabling more precise pricing for small-sample data. This cross-industry data fusion provides the model with rich feature information, enhancing its effectiveness and stability in real-world applications.

In the process of data preprocessing, One-Hot coding is carried out for the “category” and “label” columns in the classified data. This coding technique converts classification data into binary vector form, where each possible classification value is represented as a new binary feature column. The purpose of this step is that machine learning models often cannot directly process categorical data, so they need to be converted into numerical data for the model to understand and apply. Second, Box-Cox transformation is applied to the data in the “price” column. Box-Cox transformation is a statistical method used to adjust the distribution of data, the main purpose of which is to make the data more normal or close to normal distribution. The transformation adjusts the form of the transformation by introducing a parameter, thus processing the non-normal distribution of data and eliminating the skew of the data. This transformation helps to make the data more consistent with the assumptions of the linear model, improves the performance of the regression model, and helps to reduce model prediction errors caused by data skew.

4.2. Evaluation Metrics

In order to evaluate the performance of the proposed model, four different evaluation indexes are used, namely mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2). These indicators are calculated by the following formula:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (9)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (10)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (11)$$

$$R^2 = 1 - \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{y}_i - y_i)^2} \quad (12)$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and y_i and y_i are the number of samples tested. MAE is used to measure the average difference between the intended target and the predictions generated by the proposed model. MAPE is used as a loss function in model evaluation due to its intuitive representation of relative errors. RMSE is also used to assess forecast quality, which reflects the deviation between the reference value and the estimate. R^2 is used to indicate the regression model with its baseline average model ($y_i = \bar{y}_i$).

4.3. Results and Analysis

4.3.1. Ablation Experiment

To verify the improvement in data resource pricing performance under small-sample conditions through data augmentation using WGAN-GP and the Reptile algorithm, an ablation study was conducted. The results of the ablation experiment are shown in Table 3, where MLP represents the prediction model using only a single Multilayer Perceptron, GAN refers to the results from the basic generative adversarial network model, and WGAN is the data resource pricing model based on Wasserstein distance. WGAN-GP introduces gradient penalty on top of this. WGAN-GP-Reptile is the proposed model that first performs data augmentation using WGAN-GP, followed by optimization of the MLP prediction model with the Reptile algorithm. In experiments using GAN, WGAN-GP, and WGAN-GP-Reptile for data augmentation, 1000 new samples were generated.

Table 3. Results of ablation experiment.

Model	MSE	RMSE	MAE	R^2
MLP	3.9096	1.9773	1.4882	0.5268
GAN	3.767	1.9409	1.4015	0.6187
WGAN	3.7278	1.9307	1.4174	0.6227
WGAN-GP	3.695	1.9222	1.4241	0.6489
WGAN-GP-Reptile	3.2424	1.8007	1.2545	0.6807

As can be seen from the table, compared to using only data augmentation or only the Reptile algorithm for training, without combining data augmentation and meta-learning algorithms, the proposed WGAN-GP-Reptile combined model significantly improves the performance of the data resource pricing model. The WGAN-GP-Reptile model leverages the strengths of both WGAN-GP and the Reptile algorithm, resulting in a significant performance boost, especially in small-sample scenarios.

4.3.2. Comparison with Other Data Enhancement Methods

The authenticity of the generated samples, the probability fitting accuracy, and the diversity of the samples are all important indicators to evaluate the generation ability of the model. This paper includes BEGAN, VAE, and WCGAN generation models as reference objects, evaluates the generation ability of the three generation models, and discusses their performance in the authenticity of generated samples, probability fitting accuracy, and sample diversity.

First of all, compared with BEGAN (Figure 4b), VAE (Figure 4c), and WCGAN (Figure 4d), WGAN-GP (Figure 4a) method has better performance in maintaining sample authenticity. This can be attributed to the introduction of the gradient punishment mechanism in WGAN-GP, which can better maintain the consistency of distribution between generated samples and real samples, thus producing more realistic samples. Secondly, in terms of probability fitting accuracy, the three methods all show certain accuracy, especially in the prediction of price trend. However, compared with BEGAN, VAE, and WCGAN, WGAN-GP captures the subtle changes and fluctuations in price trends more accurately.

While maintaining sample diversity, WGAN-GP also pays more attention to the authenticity of sample generation and probability sub-prediction. This may be related to the use of gradient punishment and additional parameter constraints, which make the generated samples more interpretable and predictive.

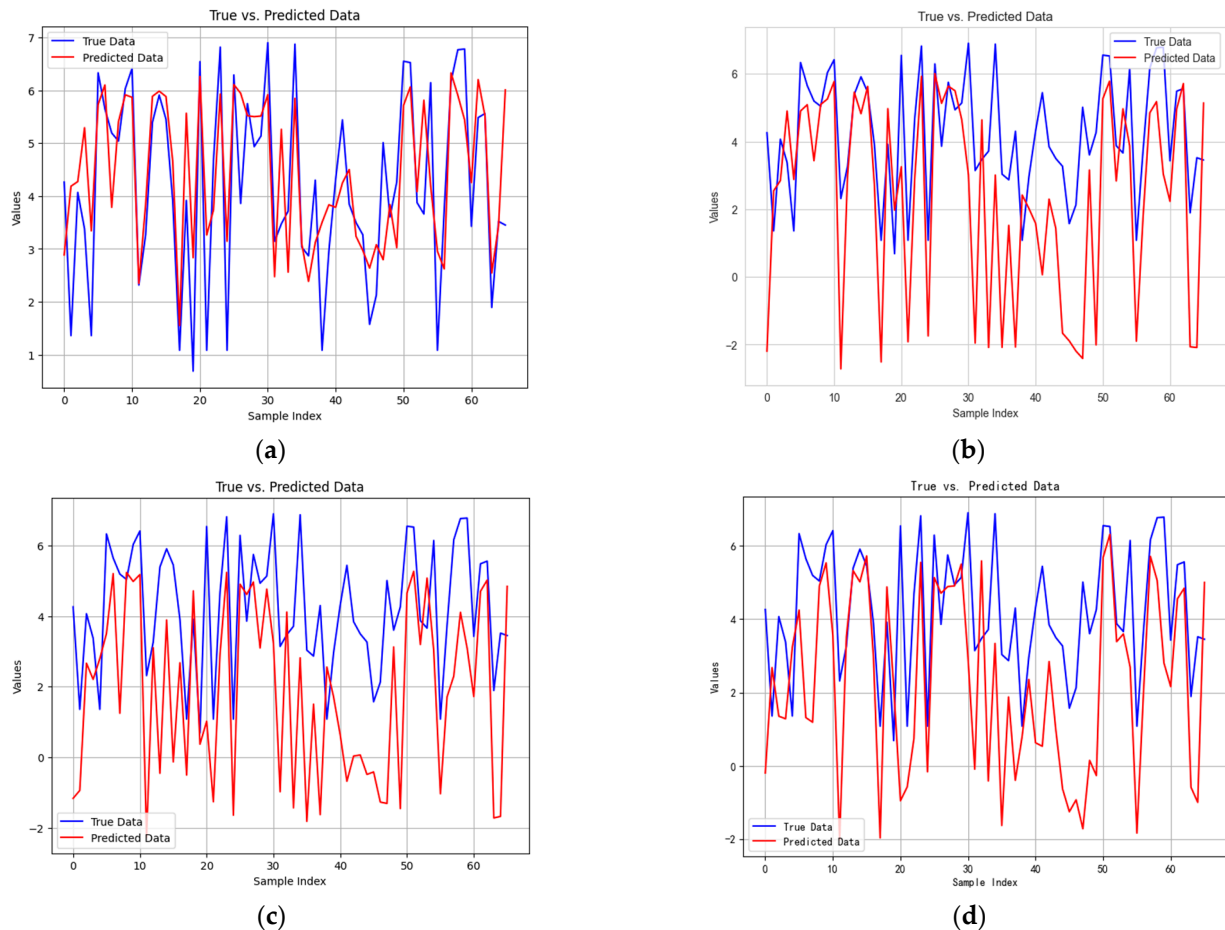


Figure 4. Prediction performance of different data augmentation models. (a) WGAN-GP; (b) BEGAN; (c) VAE; and (d) WCGAN.

4.3.3. The Effect of the Reptile Optimization Algorithm

To verify the effectiveness of the Reptile optimization algorithm, we conducted experiments on a dataset expanded by 100 samples using the WGAN-GP model. Predictions were made using the original CNN, RNN, and MLP models, as well as the same models optimized with the Reptile algorithm. The evaluation values before and after optimization for the three models are shown in Figure 5.

The R^2 comparison results before and after model optimization are shown in Figure 5. The prediction performance of all models improved after optimization with the Reptile algorithm, although the degree of improvement varied. Specifically, the Reptile algorithm improved the CNN model by only 0.11%, this may be because CNN models typically contain a large number of parameters and complex convolutional operations. Simply improving the initialization parameters has a relatively limited impact on performance since CNNs rely more on fine feature extraction and deep structure optimization with multiple layers of convolutions. Reptile's global optimization effect may not fully leverage its advantages in this context. The RNN models have fewer parameters, by 2.54% compared to CNNs, and their architecture is more sensitive to the initialization of parameters. The Reptile algorithm improves parameter initialization, enabling RNNs to better adapt to the variations in sequential data, thereby enhancing performance on small-sample datasets.

And the MLP model saw the most significant improvement, reaching 4.12%. The MLP model has a relatively simple structure and fewer parameters. In this case, the Reptile algorithm can significantly improve the initialization parameters, allowing the model to better adapt to small-sample tasks. Without many layers of deep networks, the performance of MLP is more easily influenced by the positive impact of meta-learning optimization.

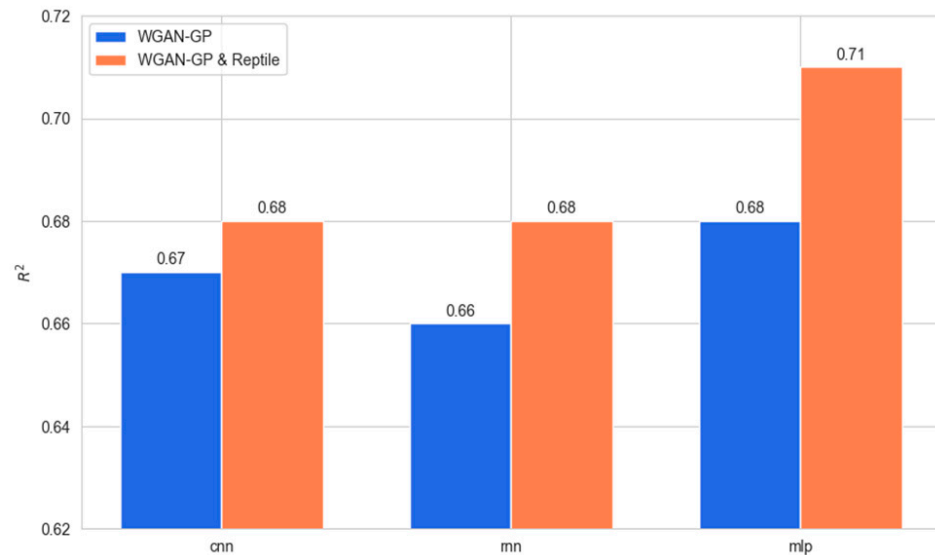


Figure 5. Comparison of model R^2 before and after Reptile algorithm optimization.

The difference in the performance improvements of various models optimized by the Reptile algorithm is mainly the result of multiple factors, including model structure, characteristics of the optimization algorithm, and task complexity. For the relatively simple MLP model with fewer parameters, the Reptile algorithm can more effectively adjust the parameters and improve model performance. By contrast, for the CNN model, which has a larger number of parameters and already exhibits high performance, the optimization effect of the Reptile algorithm is relatively limited.

This suggests that the Reptile algorithm may be more suited for simpler models or models with fewer parameters, where it can more effectively enhance performance in small-sample learning scenarios.

4.3.4. Comparison of Model Predictions on Different Sample Datasets

WGAN-GP was used to generate 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 new data samples, and then the Reptile algorithm was applied for regression prediction. The resulting values and MSE values are shown in the figure below. These results indicate a certain trend: as the amount of generated data increases, the performance of the regression prediction model fluctuates slightly but generally remains at a high level. This suggests that the combination of WGAN-GP data augmentation and the Reptile algorithm is effective in improving the performance of regression prediction tasks.

As shown in Figure 6, with the increase in the number of virtual samples, the performance first rises, then slightly decreases, and finally increases again. Although increasing the number of generated data helps improve model performance, the performance gains seem to slow down after the data amount reaches a certain threshold. This might suggest a critical point in the data quantity, beyond which further increases in data provide diminishing returns in performance improvement. This is mainly because the generated virtual samples fill the information gap during the initial training stage, enhancing model prediction accuracy. When the volume of virtual samples becomes large enough, the influence of the initial samples diminishes, and the differences between the initial and virtual samples become more significant.

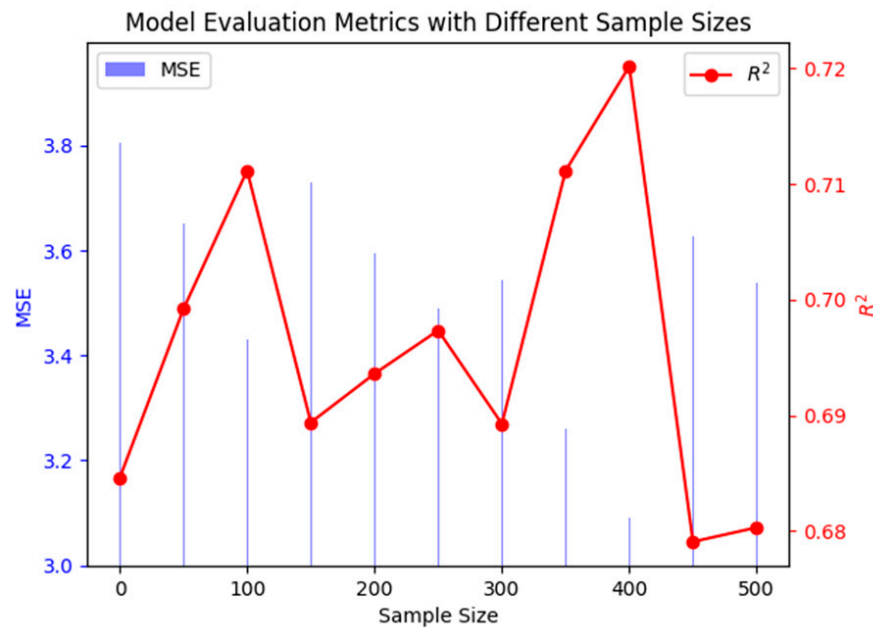


Figure 6. MSE and R^2 for different numbers of generated samples.

During the model training process, the consumption of computational resources is an undeniable challenge. To address this issue, various optimization techniques can be employed, such as model pruning and mixed precision training, which significantly reduce training time and resource costs while maintaining model performance. Additionally, using distributed computing and parallelized data generation can improve the scalability of the algorithms, allowing for efficient training on large-scale datasets. Specifically, by introducing progressive training and heterogeneous computing architectures, we can not only accelerate the WGAN-GP data augmentation process but also reduce the computational load of the Reptile algorithm when processing synthetic samples. For future research, further exploration of how to optimize the training process in different hardware environments will be a crucial direction to enhance the practical value of the algorithms.

5. Conclusions

In the digital economy era, data trading platforms not only facilitate the circulation of data and expand the market scale by integrating and transferring data resources but also lay the foundation for the sustainable allocation of data resources. Data pricing is the key to realize the market-oriented allocation of data elements, and a transparent, fair, and efficient pricing mechanism can enhance the stability and attractiveness of the platform. However, small-sample data make it difficult for traditional pricing methods to accurately evaluate the value of data, and it is easy to overfit and produce subjective bias. In this paper, a method combining WGAN-GP data enhancement and the Reptile algorithm was proposed to expand the dataset through data enhancement and fine-adjust the model using meta-learning to improve the accuracy and stability of pricing. The performance of the network was verified on a real dataset, and the experimental results show that the following can be achieved:

Using the WGAN-GP network can generate synthetic data highly similar to real data, thereby efficiently expanding the dataset required for deep learning training. With a data augmentation of 1000 samples, the average recognition rate increased by 4.88% compared to the GAN network.

Optimizing network training through the Reptile algorithm effectively enhances model performance, reduces overfitting, and achieves the desired predictive model. Compared to models without meta-learning optimization, the Reptile-optimized model's accuracy further improved by 4.90%. The data resource prediction model constructed using data augmentation and meta-learning methods effectively addresses the small-sample data

resource pricing prediction problem. The prediction accuracy of the model combining both techniques improved by 10.02% compared to using only the GAN network.

By addressing the problem of pricing small-sample data resources, this study provides a sustainable solution to improve the accuracy and fairness of data trading platform pricing, thereby enhancing trust between both parties involved in transactions, attracting more participants, expanding the market scale, and boosting the platform's long-term competitiveness. Moreover, the model's accuracy in pricing under small-sample conditions will help optimize resource allocation, promote the standardization and sustainable development of the data market, and provide reliable support for innovation and decision making across industries, fostering the progress of digitalization and intelligent development.

In practical applications, this model is not only suitable for data pricing but can also be extended to resource allocation optimization, supply chain management, and demand forecasting. Its transparency and robustness make it promising for enhancing trust in data trading platforms. Future research could explore integrating this model with blockchain technology to improve the transparency and traceability of the pricing process. Additionally, the improvement in pricing accuracy under small-sample conditions lays the foundation for establishing industry standards in the data market.

Future studies could focus on expanding the application of this model to different data types and economic contexts, developing more versatile and flexible models for resource pricing and optimization, thus driving the intelligent and standardized development of the digital economy.

Author Contributions: Conceptualization, J.S. and Y.Y.; methodology, J.S. and Y.Y.; validation, F.X. and Y.Y.; formal analysis, J.S.; investigation, Y.Y.; writing—original draft preparation, Y.Y.; writing—review and editing, J.S.; visualization, F.X.; supervision, F.X.; project administration, J.S.; funding acquisition, J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 72464018, Science Development Strategy and Policy Research Project of Yunnan, grant number 202404AL030013, and Applied Basic Research Key Project of Yunnan, grant number 202401AS070112.

Data Availability Statement: The original contributions presented in this study are included in the article; further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ouyang, R.; Du, Q. Research Progress on the Pricing Mechanisms of Data. *Econ. Perspect.* **2022**, *2*, 124–141.
2. Lv, Z.; Qiao, L.; Hossain, M.S.; Choi, B.J. Analysis of using blockchain to protect the privacy of drone big data. *IEEE Netw.* **2021**, *35*, 44–49. [\[CrossRef\]](#)
3. Li, L.; Fan, Y.; Tse, M.; Lin, K. A review of applications in federated learning. *Comput. Ind. Eng.* **2020**, *149*, 106854. [\[CrossRef\]](#)
4. Zhang, Y. Data factor pricing model based on market mechanism and transaction scenario. *Soc. Sci.* **2023**, *72–76*, 84. [\[CrossRef\]](#)
5. Rochet, J.; Tirole, J. Two-sided markets: A progress report. *Rand J. Econ.* **2006**, *37*, 645–667. [\[CrossRef\]](#)
6. Zheng, X.; Ji, Y. Two-Sided Markets, Multi-Sided Markets and High-Quality Economic Development—Inframarginal General Equilibrium Analysis of Labor Division Network. *RD Manag.* **2022**, *34*, 27–40. [\[CrossRef\]](#)
7. Wu, J.; Yuan, Y.; He, C.; Qian, L.; Du, L.; Miao, J. Research on Multilateral Platform for Data Element Transactions: CurrentStatus, Approaches, and Framework. *J. Inf. Resour. Manag.* **2024**, *14*, 4–20. [\[CrossRef\]](#)
8. Hao, J.; Deng, Z.; Li, J. The evolution of data pricing: From economics to computational intelligence. *Heliyon* **2023**, *9*, e20274. [\[CrossRef\]](#)
9. Tian, G.; Wu, W. Big data pricing in marketplace lending and price discrimination against repeat borrowers: Evidence from China. *China Econ. Rev.* **2023**, *78*, 101944. [\[CrossRef\]](#)
10. He, P.; Wang, X. Predicament and Countermeasure Research about Big Data Trading Platform in China. *J. Mod. Inf.* **2017**, *37*, 98–105,153.
11. Liu, J.; Wang, S. Dilemmas and Suggestions on Market-based Data Allocation. *Bull. Chin. Acad. Sci.* **2022**, *37*, 1435–1444. [\[CrossRef\]](#)
12. Jiang, X.; Niu, C.; Ying, C.; Wu, F.; Luo, Y. Pricing GAN-based data generators under Rényi differential privacy. *Inf. Sci.* **2022**, *602*, 57–74. [\[CrossRef\]](#)

13. Yang, M.; Wang, J.; Dou, Y. Research on the institutional approach of data elements participating in income distribution under the background of digital economy. *E-Government* **2022**, *2*, 31–39. [\[CrossRef\]](#)
14. Zhao, X. Cultivating Data Factor Market: Legal Dilemmas, Foreign Experiences and Chinese Countermeasures. *Sci. Technol. Prog. Policy* **2022**, *39*, 123–131.
15. Liu, J. Development practice of Data factor market in China: Achievements Problems and Suggestions. *Price Theory Pract.* **2021**, *12*, 18–22. [\[CrossRef\]](#)
16. Spiekermann, S.; Acquisti, A.; Böhme, R.; Hui, K. The challenges of personal data markets and privacy. *Electron. Mark.* **2015**, *25*, 161–167. [\[CrossRef\]](#)
17. Lin, L.; Li, D.; Pan, C. Improving. Virtual Sample Generation for Small Sample Learning with Dependent Attributes. In Proceedings of the 2016 5th Iiai International Congress on Advanced Applied Informatics (iiai-aai), Kumamoto, Japan, 10–14 July 2016. [\[CrossRef\]](#)
18. Wu, P.; Bhatnagar, R.; Epshtein, L.; Bhandaru, M.; Shi, Z. Alarm correlation engine (ACE). In Proceedings of the Noms 98 1998 IEEE Network Operations and Management Symposium, New Orleans, LA, USA, 15–20 February 1998. [\[CrossRef\]](#)
19. Zhou, C.; Zou, L. Semi-Supervised Gaussian Processes Active Learning Model for Imbalanced Small Data Based on Tri-Training with Data Enhancement. *IEEE Access* **2023**, *11*, 17510–17524. [\[CrossRef\]](#)
20. Han, S.; Williamson, B.D.; Fong, Y. Improving random forest predictions in small datasets from two-phase sampling designs. *BMC Med. Inform. Decis. Mak.* **2021**, *21*, 322. [\[CrossRef\]](#)
21. Etz, K.E.; Arroyo, J. A Small Sample Research: Considerations Beyond Statistical Power. *Prev. Sci.* **2015**, *16*, 1033–1036. [\[CrossRef\]](#)
22. Kuncheva, L.I.; Matthews, C.E.; Arnaiz-González, A.; Rodríguez, J.J. Feature selection from high-dimensional data with very low sample size: A cautionary tale. *arXiv* **2020**, arXiv:2008.12025. [\[CrossRef\]](#)
23. Zhang, W.; Zhang, H. A feature extraction method for small sample data based on optimaensemble random forest. *J. Northwestern Polytech. Univ.* **2022**, *40*, 1261–1268. [\[CrossRef\]](#)
24. Zhang, F.; Zhou, Z.; Wang, R.; Huang, X.; Han, Y. Radio Data Augmentation with GAN for Automatic Modulation Classification. *J. Cyber Secur.* **2023**, *8*, 47–60. [\[CrossRef\]](#)
25. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative Adversarial Networks: An Overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [\[CrossRef\]](#)
26. Arjovsky, M.; Bottou, L. Towards principled methods for training generative adversarial networks. *arXiv* **2017**, arXiv:1701.04862. [\[CrossRef\]](#)
27. Goodfellow, I. Nips 2016 tutorial: Generative adversarial networks. *arXiv* **2016**, arXiv:1701.00160.
28. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of wasserstein gans. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5769–5779.
29. Chen, J.; Yan, Z.; Lin, C.; Yao, B.; Ge, H. Aero-engine high speed bearing fault diagnosis for data imbalance: A sample enhanced diagnostic method based on pre-training WGAN-GP. *Measurement* **2023**, *213*, 112709. [\[CrossRef\]](#)
30. Chen, L.; Fu, D. Survey on Machine Learning Methods for Small Sample Data. *Comput. Eng.* **2022**, *48*, 1–13. [\[CrossRef\]](#)
31. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. *Int. Conf. Mach. Learn.* **2017**, *70*, 1126–1135.
32. Nichol, A.; Achiam, J.; Schulman, J. On first-order meta-learning algorithms. *arXiv* **2018**, arXiv:1803.02999.
33. Zhang, X.; Hu, C.; He, B.; Han, Z. Distributed Reptile Algorithm for Meta-Learning Over Multi-Agent Systems. *IEEE Trans. Signal Process.* **2022**, *70*, 5443–5456. [\[CrossRef\]](#)
34. Nguyen, D.Q.; Vien, N.A.; Dang, V.; Chung, T. Asynchronous framework with Reptile+ algorithm to meta learn partially observable Markov decision process. *Appl. Intell.* **2020**, *50*, 4050–4062. [\[CrossRef\]](#)
35. Tian, C.; Zhu, X.; Hu, Z.; Ma, J. A transfer approach with attention reptile method and long-term generation mechanism for few-shot traffic prediction. *Neurocomputing* **2021**, *452*, 15–27. [\[CrossRef\]](#)
36. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
37. Cheng, J.; Yang, Y.; Tang, X.; Xiong, N.; Zhang, Y.; Lei, F. Generative adversarial networks: A literature review. *Ksii Trans. Internet Inf. Syst.* **2020**, *14*, 4625–4647. [\[CrossRef\]](#)
38. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. *Int. Conf. Mach. Learn.* **2017**, *70*, 214–223.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.