https://doi.org/10.62836/jitp.2023.159

Global Science Publishing

JITP

Journal of Information, Technology and Policy

https://ojs.sgsci.org/journals/jitp

Article

# A Lightweight Ensemble Model Based on Knowledge Distillation and Distributed Data Parallelism for Predicting User Advertising Return on Investment

**Yu Qiao [1], Alan Wilson [2] and Zhaoyan Zhang [3,*]**

[1] *Meta Platforms, Inc., Bellevue, WA 98005, USA*

[2] *Intact Financial Corporation, Toronto, ON M5G 1Z5, Canada*

[3] *Zhongke Zhidao (Beijing) Technology Co., Ltd., Beijing 102627, China*

**Abstract:** Advertising plays a pivotal role in enabling businesses to connect with potential customers and promote their offerings. In today's digital age, advertising channels such as online display ads, social media promotions, and targeted email campaigns dominate the marketing landscape. Given the substantial investments companies make in these channels, evaluating advertising effectiveness through Return on Investment (ROI)—a metric representing the ratio of net profit to advertising expenditure—becomes crucial. Accurately predicting user advertising ROI aids in optimizing campaign strategies, ensuring resources are allocated effectively. Traditional heuristic and rule-based methods often fail to capture the complex relationships in user data, leading to limited predictive accuracy. Recent advancements in machine learning, particularly deep learning, have significantly improved ROI prediction by uncovering intricate, non-linear patterns in large datasets. However, deep learning models can be computationally intensive and challenging to deploy in resource-constrained environments. To address these limitations, this study proposes a novel lightweight distributed ensemble model that leverages distributed data parallelism (DDP), knowledge distillation, and ensemble learning. The framework trains a large teacher network using DDP, followed by distilling knowledge into a smaller student network, and integrates high-level representations with other machine learning models. The results demonstrate improved prediction accuracy and computational efficiency, making the model suitable for real-time advertising ROI forecasting.

**Keywords:** ROI prediction; ensemble model; knowledge distillation; distributed training

## 1. Introduction

Advertising has long been recognized as a fundamental strategy for businesses to communicate with potential customers and promote their products or services [1, 2]. In today's highly competitive digital landscape, advertising takes many forms, including online display ads, sponsored social media posts, and targeted email campaigns. As companies invest increasingly large budgets in these advertising channels, measuring the effectiveness of such investments becomes critical. A common metric for assessing this effectiveness is Return on Investment (ROI) [3–5], which captures the ratio of net profit to advertising costs. For user-focused advertising, ROI offers insights into how effectively a campaign reaches its intended audience,

Received: 15 August 2023; Accepted: 11 September 2023.

* Corresponding: Zhaoyan Zhang (zhaoyanzhang9394@gmail.com)

drives engagement, and ultimately generates revenue. Understanding and predicting user advertising ROI is therefore essential for optimizing campaign strategies and resource allocation.

Traditionally, advertisers and marketers relied on heuristic or rule-based approaches to estimate and forecast ROI [6]. These methods might include analyzing past campaign performance, using simple trend extrapolation, or applying basic statistical models [7–9]. While these approaches provided a starting point for evaluating campaign effectiveness, they often struggled to handle the complexity of modern digital advertising. With the proliferation of diverse user data—ranging from demographic information to real-time browsing behavior—traditional models frequently fail to capture nuanced relationships and interactions among multiple variables. As a result, their predictive power can be limited, leading to suboptimal budgeting decisions and reduced returns [10–12].

Over the past decade, machine learning and, more recently, deep learning have emerged as powerful tools to address these shortcomings. In particular, deep learning has demonstrated remarkable success in various domains, including computer vision, natural language processing, and recommender systems [13–16]. Neural networks excel at discovering complex, non-linear patterns in high-dimensional data, making them well-suited for advertising ROI prediction tasks. Researchers have developed numerous deep learning-based methods, such as multi-layer perceptrons (MLPs), recurrent neural networks (RNNs), and transformer-based architectures, each showing improvements over traditional statistical approaches. Alongside deep learning, ensemble methods (e.g., stacking or blending multiple models) further enhance performance by combining the strengths of individual predictors [17,18].

Despite these advances, there remain several gaps in the existing literature. First, many state-of-the-art deep learning models can be computationally heavy, requiring substantial Graphics Processing Unit (GPU) resources and prolonged training times. This high computational cost can be problematic for practitioners who need to update models frequently with new data or who lack large-scale hardware infrastructures. Second, distributed training techniques, such as Distributed Data Parallel (DDP), are increasingly employed to reduce training times and handle large datasets, yet the focus on lightweight approaches that maintain high predictive accuracy while minimizing resource consumption is still relatively sparse. Third, although knowledge distillation has proven effective for compressing large models into smaller, faster student networks without sacrificing too much accuracy, it has not been widely explored in the context of distributed training for advertising ROI prediction. Consequently, there is a clear opportunity to investigate how these techniques—DDP, model compression, and ensemble strategies—can be combined to achieve a more efficient yet accurate system.

In response to these challenges, this work proposes a novel lightweight distributed ensemble model shown in Figure 1 for user advertising return on investment prediction. The core idea involves training a relatively larger teacher network using distributed data parallelism, thereby leveraging multiple computational nodes (or GPUs) to speed up the learning process. Once the teacher network is fully trained, we employ knowledge distillation to transfer its learned representations to a smaller, student network. This student network maintains much of the predictive power of the teacher model but requires fewer parameters and less computational overhead, making it more practical for real-world deployment. Furthermore, we integrate traditional machine learning models, such as KNN or SVR, in an ensemble fashion. By extracting high-level features from the student network's penultimate layer, we combine these distilled representations with the outputs of conventional predictors. This stacking or blending strategy capitalizes on the complementary strengths of different modeling paradigms: deep neural networks excel at discovering intricate feature representations, while classical algorithms can still perform robustly on certain data distributions. The final ensemble aims to deliver improved accuracy and efficiency compared to either deep learning or traditional methods alone.

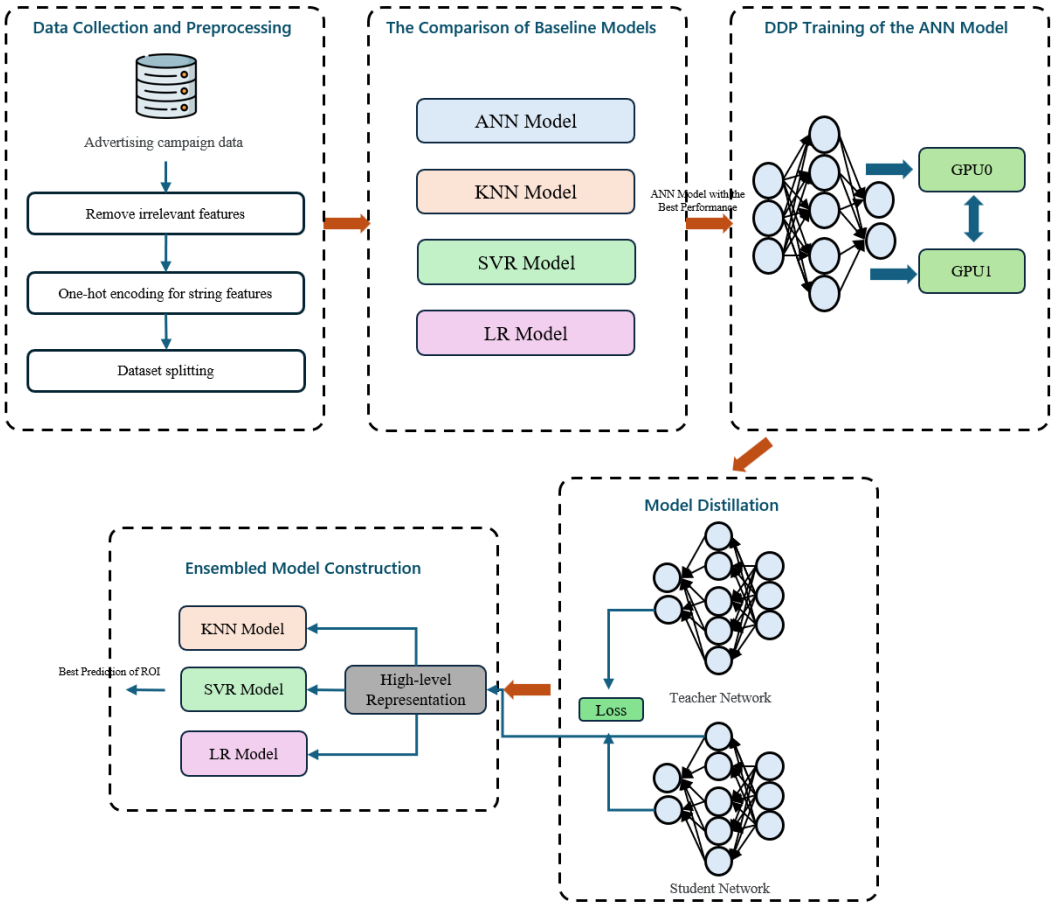


**Figure 1.** The process of the proposed lightweight ensemble model based on knowledge distillation and distributed data parallelism.

## 2. Literature Review

Predicting user advertising ROI has become a focal point in marketing analytics, especially with the advent of machine learning techniques that offer enhanced accuracy and efficiency [19,20]. Shen et al. proposed a two-stage framework designed to optimize ROI in large-scale promotional campaigns. In the initial stage, they employed machine learning models to predict individual user responses to promotions. Subsequently, they formulated an optimization problem that allocates incentives to users based on business objectives and resource constraints. A key innovation in their approach was the introduction of the deep-isotonic-promotion-network (DIPN), a deep neural network architecture that enforces isotonicity and smoothness in promotion response curves, thereby enhancing prediction accuracy [21].

Lewis and Wong addressed the challenge of quantifying advertising incrementality—the causal effect of ad exposure on user behavior. They developed a methodology that integrates ad bidding strategies, attribution models, and experimental data to compute the incremental impact of advertising efforts. Their approach leverages machine learning and causal econometrics to create a computational model that informs both bidding and attribution, aiming to improve ROI by accurately measuring the true effect of advertisements [22]. Kong et al. focused on optimizing bid recommendations to enhance advertising ROI. They introduced a scenario that identifies concavity changes in click prediction curves, determining optimal bid values where the marginal gain begins to diminish. By applying parametric learning and solving the associated constrained optimization problem, their method demonstrated significant improvements in business metrics, including revenue and click-through rates, thereby offering a practical solution for bid optimization in advertising platforms [23]. Nakagawa et al. developed the Ranked Information Coefficient Neural Network (RIC-NN), a deep learning framework aimed at predicting stock returns. RIC-NN incorporates a nonlinear multi-factor approach, utilizes ranked information coefficients as stopping criteria, and applies transfer learning across different regions. This model has demonstrated superior performance compared to traditional machine learning methods and has

outperformed major equity investment funds over a fourteen-year period [24].

Despite these advancements, challenges persist in ROI prediction. Many machine learning models are computationally intensive, necessitating substantial resources, which may not be feasible for all practitioners. Therefore, there is a growing need for lightweight, distributed models that maintain high predictive accuracy while optimizing computational efficiency.

## 3. Method

### *3.1. Dataset Preparation*

We utilized a user advertising dataset comprising 1000 records and 17 original features as well as one label. The distribution of all features is provided in Figures 2 and 3. The prediction target in this dataset is the return on investment, which reflects the effectiveness of advertising campaigns. To prepare the data for modeling, we removed unnecessary features, including 'user_id', 'timestamp' and 'ad_id', resulting in 14 relevant features. The dataset includes both numerical features (6 in total) and categorical features (8 in total). To ensure compatibility with machine learning algorithms, we applied one-hot encoding to the categorical features, expanding the total feature count to 987 after encoding. For model training and evaluation, the dataset was divided into three subsets: (1) 70% of the data (700 records) was used for training the model. (2) 10% of the data (100 records) was reserved for validation to fine-tune hyperparameters. (3) The remaining 20% (200 records) was set aside for testing the model's predictive performance.

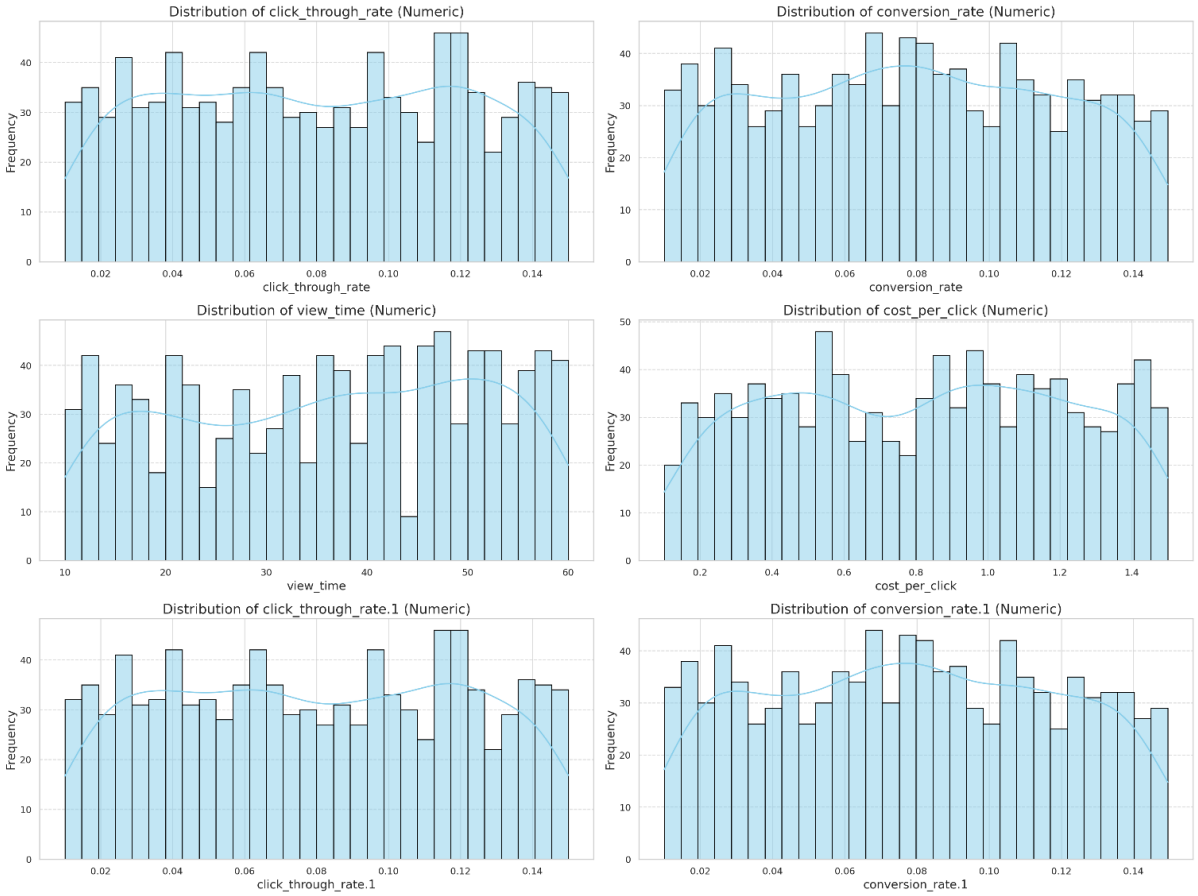


**Figure 2.** The distribution of all numerical features.

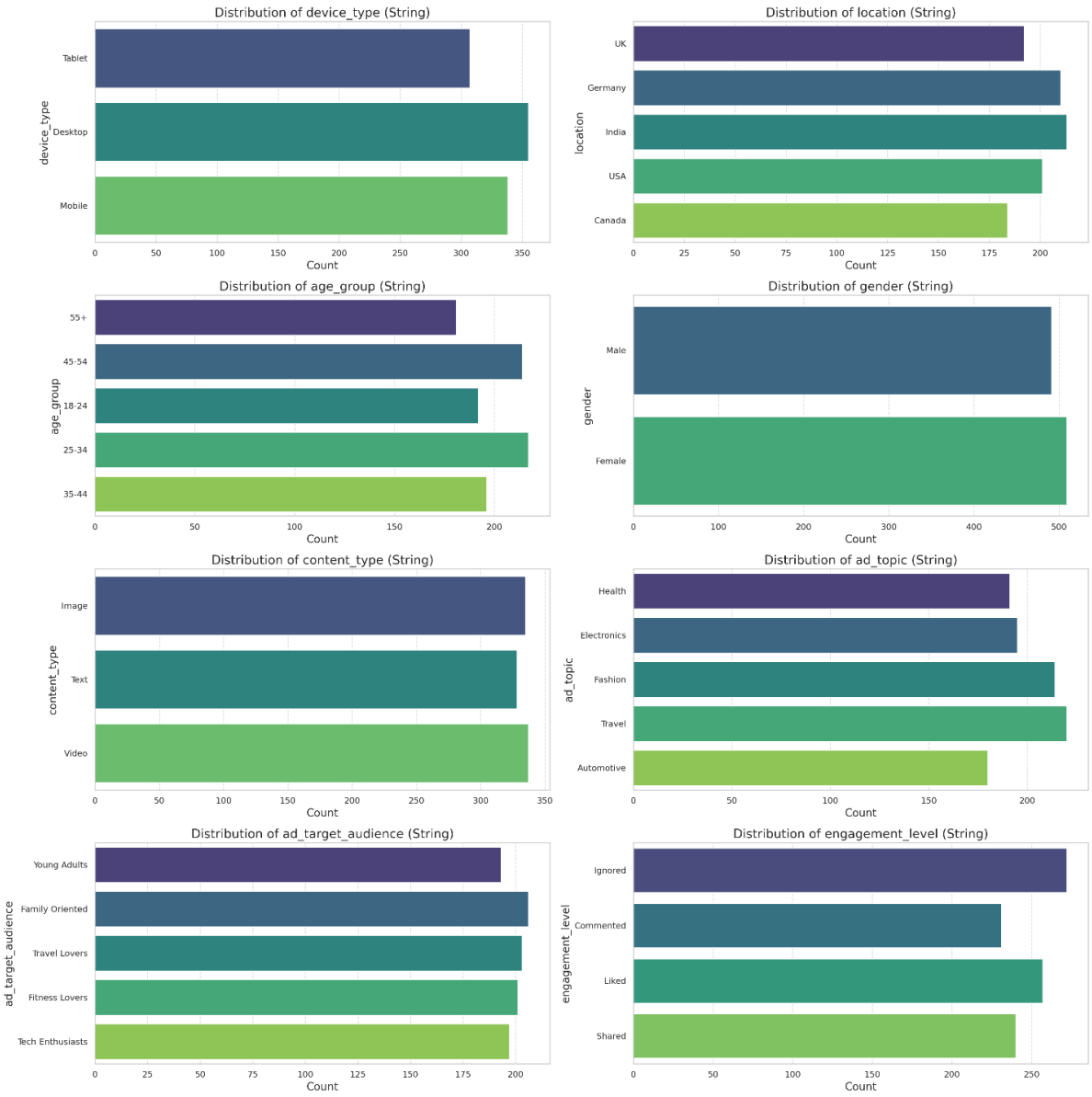


**Figure 3.** The distribution of all categorical features.

### *3.2. Baseline Machine Learning Models*

To predict user advertising ROI, we initially employed multiple baseline machine learning models. This comparative analysis aimed to identify the most suitable model for further optimization. By evaluating each model's performance based on key metrics such as Mean Absolute Error (MAE) [25], Mean Squared Error (MSE) [26], Root Mean Squared Error (RMSE) [27], and $R^2$, we ensured that subsequent enhancements would be built upon the most robust predictive foundation. The baseline models selected for this study include ANN, KNN, SVR, and LR.

#### 3.2.1. Artificial Neural Networks

Artificial neural networks [28–30] are computational models inspired by the structure and functioning of the human brain. They consist of layers of interconnected nodes (neurons), including an input layer, one or more hidden layers, and an output layer. Each connection between neurons has an associated weight, which is adjusted during training through backpropagation to minimize prediction errors. ANN excels at capturing complex, non-linear relationships within data, making it highly versatile across various tasks. The model's flexibility allows it to approximate any continuous function given sufficient hidden units. However, ANN models are often computationally intensive, requiring significant resources and careful tuning of hyperparameters such as the number of layers, neurons per layer, learning rate, and activation functions. Despite

these challenges, ANN remains a powerful tool due to its ability to handle high-dimensional data and uncover intricate patterns.

The architecture of our ANN consists of four fully connected layers designed to capture complex patterns within the dataset. The input layer is tailored to match the dimensionality of the feature space, ensuring seamless integration with the processed data. The first hidden layer contains 64 neurons and utilizes a ReLU activation function, which introduces non-linearity and helps prevent vanishing gradient issues during training. The second hidden layer expands to 128 neurons, also employing the ReLU activation function, allowing the network to capture more complex relationships within the data. Subsequently, the third hidden layer reduces the number of neurons back to 64, maintaining the ReLU activation function to sustain non-linearity and efficient learning. This layer, often referred to as the penultimate layer, plays a crucial role in generating high-level representations that are later utilized in ensemble modeling and knowledge distillation processes. Finally, the output layer consists of a single neuron, providing the final prediction output. The model was trained using MSE Loss as the loss function, optimized with the Adam optimizer (learning rate = 0.001) for 10 epochs, ensuring efficient convergence and accurate predictions.

#### 3.2.2. K-Nearest Neighbors

K-Nearest neighbors [31–33] is a non-parametric, instance-based learning algorithm that can be used for both classification and regression. It operates by identifying the k closest data points in the feature space to a given query point, using distance metrics such as Euclidean, Manhattan, or Minkowski distances. The prediction is made based on the average (for regression) or majority vote (for classification) of the neighboring points. One of KNN's main advantages is its simplicity and ease of interpretation, as it makes no assumptions about the underlying data distribution. However, KNN's performance can be significantly influenced by the choice of k and the distance metric. It can also be computationally expensive during the prediction phase, especially with large datasets, as it requires calculating the distance from the query point to all points in the training set. Nonetheless, its adaptability and intuitive nature make KNN a valuable baseline model.

#### 3.2.3. Support Vector Regression

Support vector regression [34–36] is an extension of the support vector machine framework for regression tasks. SVR attempts to fit a function within a margin of tolerance (epsilon) around the data points while minimizing model complexity. The core idea is to ensure that the model is as flat as possible while still fitting the majority of the data points within the defined margin. SVR employs kernel functions such as linear, polynomial, and radial basis function (RBF) kernels to handle non-linear relationships. This flexibility makes SVR suitable for modeling complex patterns in data. The performance of SVR is highly dependent on the choice of kernel, regularization parameter (C), and epsilon, which control the trade-off between model complexity and training error. Although SVR can be computationally intensive, particularly with large datasets and non-linear kernels, it is well-regarded for its robustness and ability to generalize well to unseen data.

#### 3.2.4. Linear Regression

Linear regression [37–39] is one of the simplest and most interpretable models in machine learning. It assumes a linear relationship between the independent variables and the dependent variable, represented by a straight line in two-dimensional space or a hyperplane in higher dimensions. The model estimates coefficients for each feature that minimize the sum of squared differences between the observed and predicted values using techniques such as ordinary least squares. LR provides clear insights into how each feature contributes to the prediction, which can be particularly useful for interpretability. However, LR relies on several assumptions, including linearity, independence of errors, and homoscedasticity, which may not hold in all datasets. Additionally, it is sensitive to multicollinearity, where highly correlated features can distort the estimated coefficients. Despite these limitations, LR remains a fundamental tool, often serving as a benchmark for evaluating the performance of more complex models.

### *3.3. DDP Training of the Artificial Neural Network*

After evaluating the performance of various baseline machine learning models, including ANN, KNN, SVR, and LR, we observed that the ANN consistently outperformed the other models across key performance metrics. Due to its superior ability to capture complex, non-linear relationships in the dataset, ANN was selected as the primary model for subsequent optimization and enhancement. To further improve the training efficiency and scalability of the ANN model, we adopted the DDP training approach. DDP is a parallelization technique provided by PyTorch that allows for efficient training of deep learning models across multiple GPUs. Unlike traditional data parallelism, which can suffer from slowdowns due to gradient aggregation on a single device, DDP distributes the model across multiple GPUs, each processing a portion of the input data in parallel [40–42]. After each forward and backward pass, DDP synchronizes gradients across all devices, ensuring consistent model updates while minimizing communication overhead. This approach leads to near-linear scaling of training speed with the number of GPUs used.

### *3.4. Knowledge Distillation for Buidling Lightweight ANN Model*

The previously constructed ANN demonstrated strong feature extraction capabilities due to its deeper architecture and multiple layers. However, the large number of parameters made it computationally intensive, which is not ideal for real-world commercial deployment where efficiency and scalability are critical. To address this limitation and build a more lightweight model without significantly compromising performance, we adopted Knowledge Distillation (KD) [43–45].

Knowledge Distillation is a model compression technique where a smaller, simpler model (student model) is trained to replicate the behavior of a larger, more complex model (teacher model). The student model learns not only from the ground truth but also from the soft targets provided by the teacher model, which contain rich information about the underlying data distribution. This approach allows the student model to achieve similar performance levels while maintaining a reduced number of parameters and faster inference times.

In this study, we designed a student model with a streamlined architecture consisting of three fully connected layers. The first layer contains 32 neurons, followed by a hidden layer with 16 neurons, and finally an output layer with a single neuron. ReLU activation functions were applied after each hidden layer to introduce non-linearity. Compared to the teacher model, the student model's reduced complexity makes it more suitable for deployment in resource-constrained environments. During the distillation process, the student model was trained using a loss function that combines the traditional prediction loss with a distillation loss. The distillation loss measures the difference between the outputs of the student and teacher models, enabling the student to mimic the teacher's predictive behavior. By balancing these two loss components, the student model inherits the teacher model's generalization capabilities while significantly reducing the computational burden.

### *3.5. Lightweight ANN Model and Its Integration with Other Machine Learning Models*

After training the lightweight student model using knowledge distillation, we further enhanced its predictive performance through an ensemble learning strategy [46–48]. Although the student model retained much of the predictive capability of the larger teacher model while significantly reducing the number of parameters, there was still potential to boost its overall accuracy by leveraging the strengths of other machine learning algorithms. To achieve this, we extracted the high-level features from the penultimate layer (the second-to-last layer) of the trained student model. This layer captures refined and abstracted representations of the input data, which are highly informative for subsequent prediction tasks. Instead of relying solely on the ANN's final output, these high-level features were combined with the outputs of three additional machine learning models: KNN, SVR, and LR. The integration process involved training each of these models using the high-level representations obtained from the student network. Predictions from these individual models were then compared. The final output of the ensemble system was selected based on the best-performing model, evaluated using standard performance metrics such as MAE, MSE, RMSE, and $R^2$.

## 4. Results and Discussion

### *4.1. The Performance Comparison among Baseline Models*

The comparison of different baseline models for ROI prediction highlights notable differences in performance across various evaluation metrics, including MAE, MSE, RMSE, and $R^2$ shown in Table 1 and Figure 4. The ANN consistently outperformed the other models in all key performance measures. Specifically, the ANN achieved the lowest MAE of 0.6967, indicating the smallest average absolute error between predicted and actual values. It also recorded the lowest MSE at 1.5392 and RMSE at 1.2406, reflecting its ability to minimize large errors more effectively than the competing models. The $R^2$ value of 0.5274 further demonstrates that the ANN could explain over half of the variance in the target variable, showcasing a stronger fit to the data compared to the other baseline models.

In contrast, the Support Vector Regression (SVR) model showed a higher MAE of 0.9913, MSE of 3.5921, and RMSE of 1.8953, accompanied by a negative $R^2$ value of −0.1030. This negative $R^2$ suggests that the SVR model performed worse than a simple mean-based prediction, highlighting its limited effectiveness in capturing the underlying patterns of the dataset. Similarly, the K-Nearest Neighbors (KNN) model presented an MAE of 1.1153, MSE of 3.4488, and RMSE of 1.8571, with an $R^2$ value of −0.0590, reflecting its suboptimal performance in predicting ROI accurately. The Linear Regression (LR) model performed the poorest among the four, with the highest MAE of 1.5717, MSE of 4.2424, RMSE of 2.0597, and an $R^2$ value of −0.3026, indicating a weak linear relationship between the features and the target variable. Additionally, scatter plots shown in Figure 5 comparing predicted and actual ROI values reveal that the ANN's predictions align more closely along the ideal diagonal line, representing perfect predictions. In comparison, the predictions from SVR, KNN, and LR models exhibit greater dispersion from the ideal line, highlighting their reduced predictive accuracy.

**Table 1.** The performance of baseline models evaluated by different metrics.

| Model Names | MAE | MSE | RMSE | $R^2$ |
|---|---|---|---|---|
| ANN | **0.6967** | **1.5392** | **1.2406** | **0.5274** |
| SVR | 0.9913 | 3.5921 | 1.8953 | −0.1030 |
| KNN | 1.1153 | 3.4488 | 1.8571 | −0.0590 |
| LR | 1.5717 | 4.2424 | 2.0597 | −0.3026 |

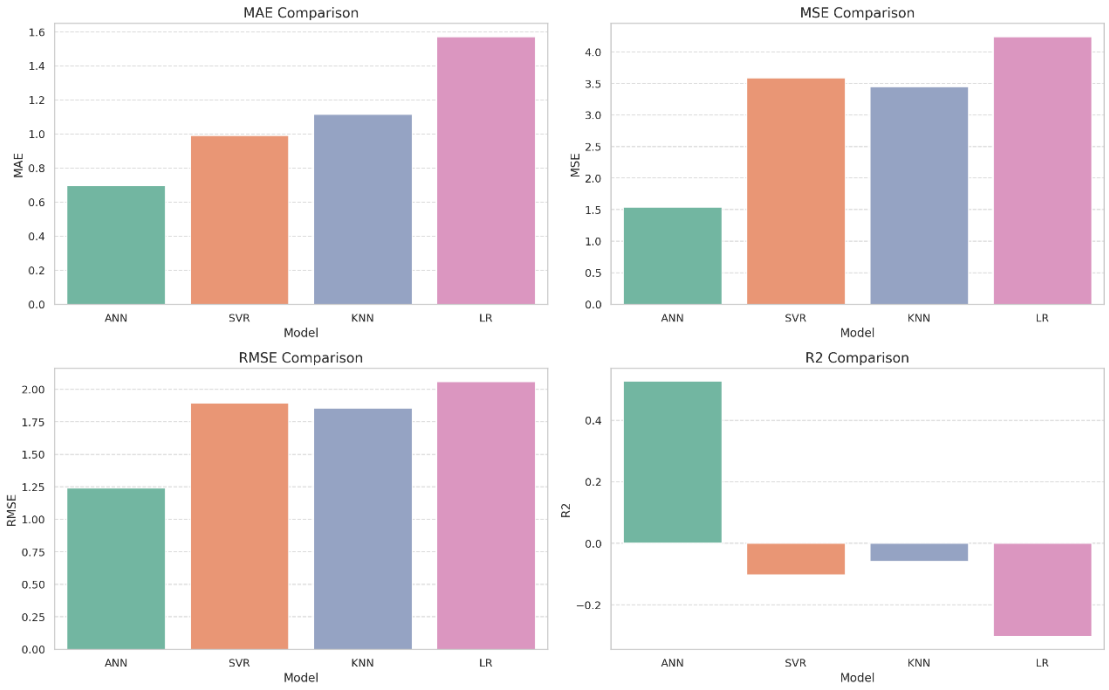


**Figure 4.** The visualization of comparison among various baseline models evaluated by different metrics.

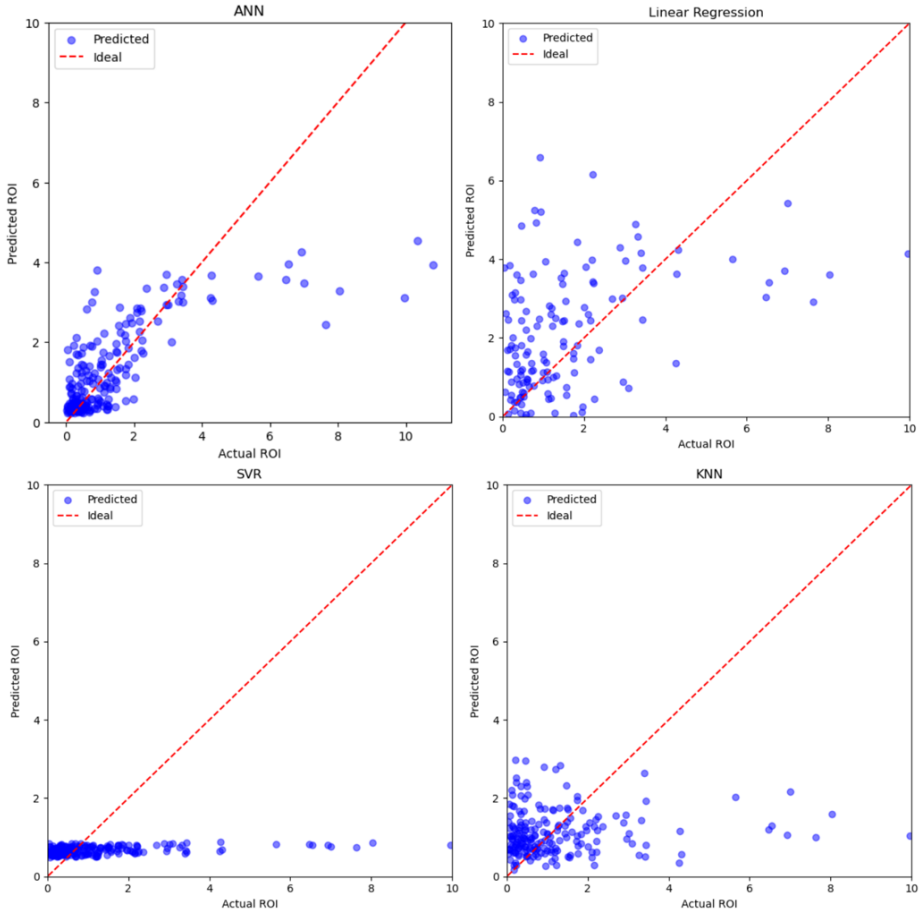


**Figure 5.** The visualization of prediction results among various baseline models.

*4.2. The Performance Comparison between ANN (teacher) and Lightweight ANN (Student) Model*

The comparison between the teacher and student ANN models is presented in Table 2 and visualized in Figures 6 and 7. The teacher ANN model, with a deeper architecture and more parameters, achieved superior performance across all evaluated metrics mentioned before. In contrast, the student ANN model, constructed through knowledge distillation, demonstrated slightly lower predictive performance. The MAE, MSE, and RMSE values increased to 0.8533, 2.4199, and 1.5556, respectively, with the $R^2$ value decreasing to 0.2570. This decline in performance is expected, as the student model was intentionally designed with fewer layers and parameters (32, 129) to reduce computational complexity. Despite this reduction in accuracy, the student model's inference time significantly improved, dropping to 0.007590 s. This represents a substantial increase in processing speed, making the student model highly suitable for real-time prediction scenarios where rapid response times are critical (Table 2, Figure 6). Figure 7 further illustrates the prediction performance by comparing the scatter plots of predicted versus actual ROI values for both models. The teacher ANN's predictions align more closely with the ideal diagonal line, reflecting higher predictive accuracy. Although the student model's predictions show greater dispersion, the overall trend remains consistent with the actual values.

**Table 2.** The performance of teacher and student ANN models evaluated by different metrics.

| Model Names | MAE | MSE | RMSE | $R^2$ | Inference Time | The Number of Parameters |
|---|---|---|---|---|---|---|
| Teacher ANN | **0.6967** | **1.5392** | **1.2406** | **0.5274** | 0.022696 s | 79809 |
| Student ANN | 0.8533 | 2.4199 | 1.5556 | 0.2570 | **0.007590 s** | **32129** |

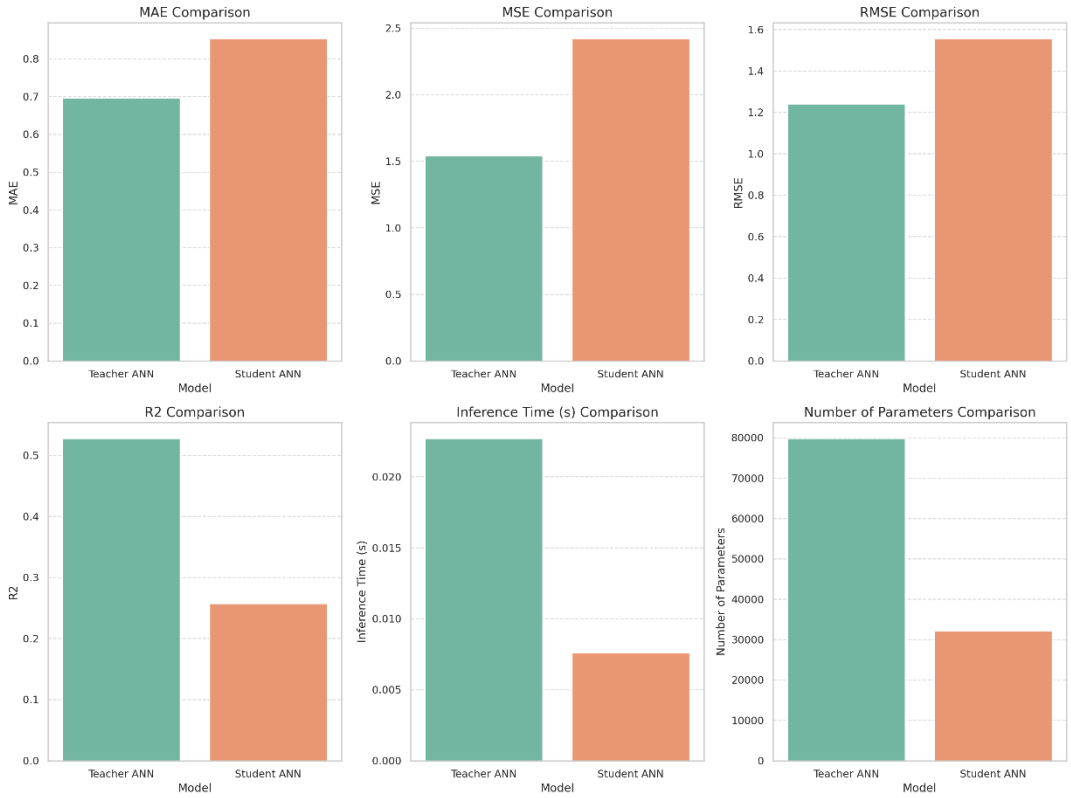


**Figure 6.** The visualization of comparison between teacher and student ANNs evaluated by different metrics.

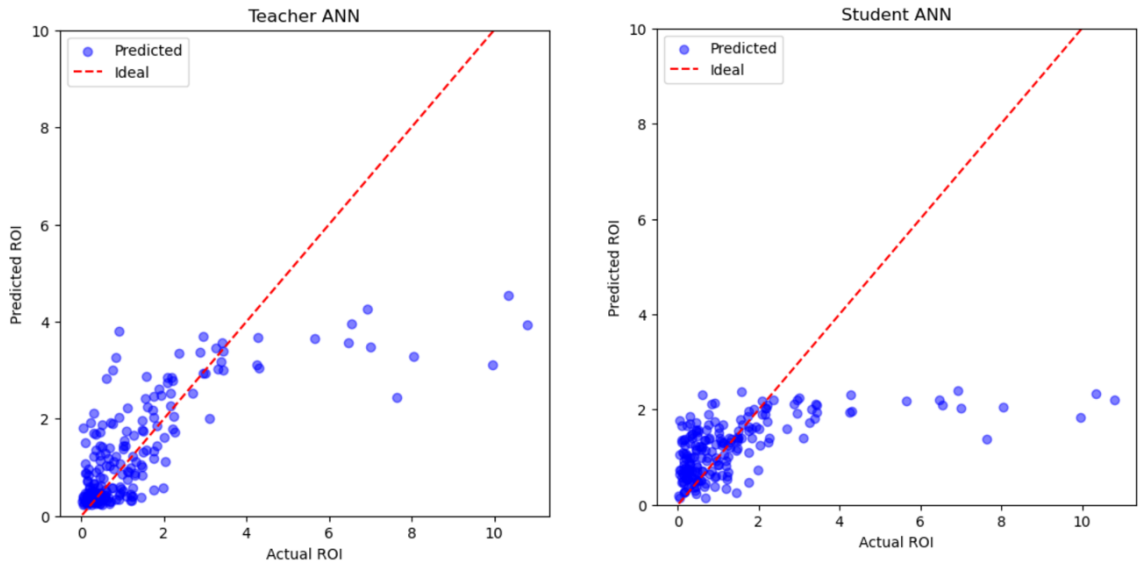


**Figure 7.** The visualization of prediction results between teacher and student ANNs.

### *4.3. The Performance Comparison among Various Ensembled Models*

We further compared ensembled models to evaluate the impact of integrating the student ANN with different models. The performance comparison of various ensembled models is presented in Table 3 and visualized in Figures 8 and 9. Among the evaluated models, the Student ANN + KNN ensemble achieved the best overall performance, with the lowest MAE (0.6606), MSE (1.3450), and RMSE (1.1598), along with the highest $R^2$ value of 0.5870 (Table 3, Figure 8). This improvement can be attributed to KNN's ability to capture local data patterns based on similarity measures, which complements the student ANN's global feature extraction capability. The scatter plot in Figure 9 further confirms this, as the predictions of the Student ANN + KNN ensemble align more closely with the ideal diagonal line, indicating higher prediction accuracy. In contrast, the

Student ANN + LR and Student ANN + SVR ensembles exhibited inferior performance, with negative $R^2$ values ($-0.0457$ and $-0.1229$, respectively), suggesting that these models failed to generalize effectively. The poor performance of the LR ensemble may result from its assumption of linearity, which cannot capture the non-linear relationships in the data. Similarly, the SVR ensemble's lower performance could be due to inappropriate kernel settings or sensitivity to outliers. Overall, the results demonstrate that incorporating KNN into the ensemble effectively enhances the predictive performance of the student ANN model by leveraging local data structures.

**Table 3.** The performance of various ensembled models evaluated by different metrics.

| Model Names | MAE | MSE | RMSE | $R^2$ |
|---|---|---|---|---|
| Student ANN | 0.8533 | 2.4199 | 1.5556 | 0.2570 |
| Student ANN + LR | 1.0927 | 3.4056 | 1.8454 | −0.0457 |
| Student ANN + SVR | 1.0258 | 3.6570 | 1.9123 | −0.1229 |
| Student ANN + KNN | **0.6606** | **1.3450** | **1.1598** | **0.5870** |

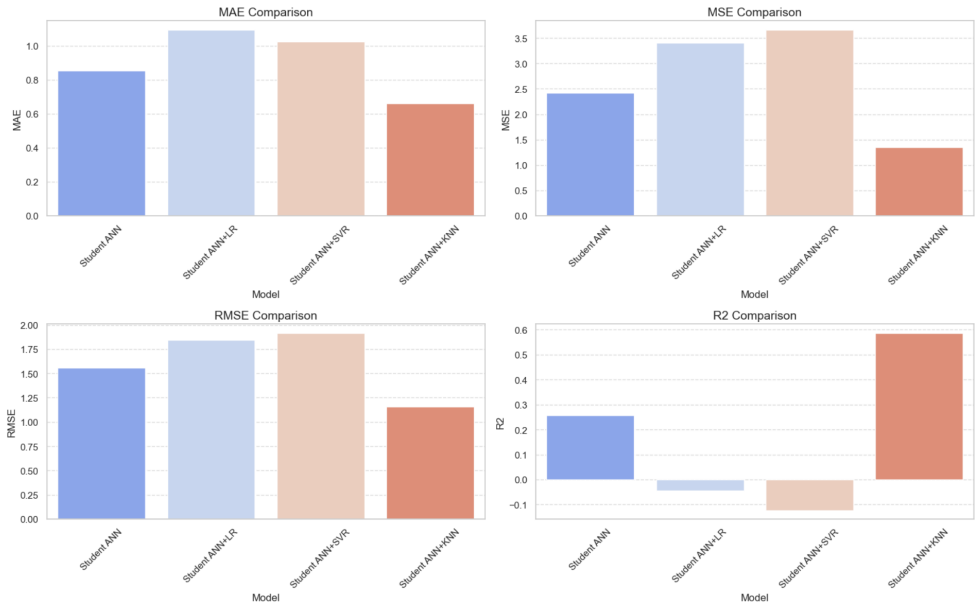


**Figure 8.** The visualization of comparison among different ensembled models.

*4.4. Discussion*

The proposed method, which integrates DDP training, knowledge distillation, and ensemble learning, demonstrates strong predictive performance with notable improvements in computational efficiency. The final ensembled model, particularly the Student ANN + KNN combination, achieves the best trade-off between accuracy and inference speed, making it highly suitable for real-time ROI prediction scenarios. The use of knowledge distillation significantly reduces the number of parameters and inference time while maintaining reasonable predictive accuracy. Moreover, the ensemble strategy further enhances performance by leveraging the complementary strengths of different models.

However, despite these advantages, some limitations remain. The student model, while lightweight, still experiences a decline in accuracy compared to the teacher model, as seen in reduced $R^2$ values. This performance drop indicates that knowledge distillation might not fully capture the complex patterns learned by the larger teacher network. Additionally, the ensemble model relies on predefined machine learning algorithms, which may limit its adaptability to highly dynamic advertising environments. Future work could explore more advanced ensemble techniques or adaptive learning methods to address these challenges. Incorporating more diverse data sources and testing on larger datasets could also improve generalization and robustness.

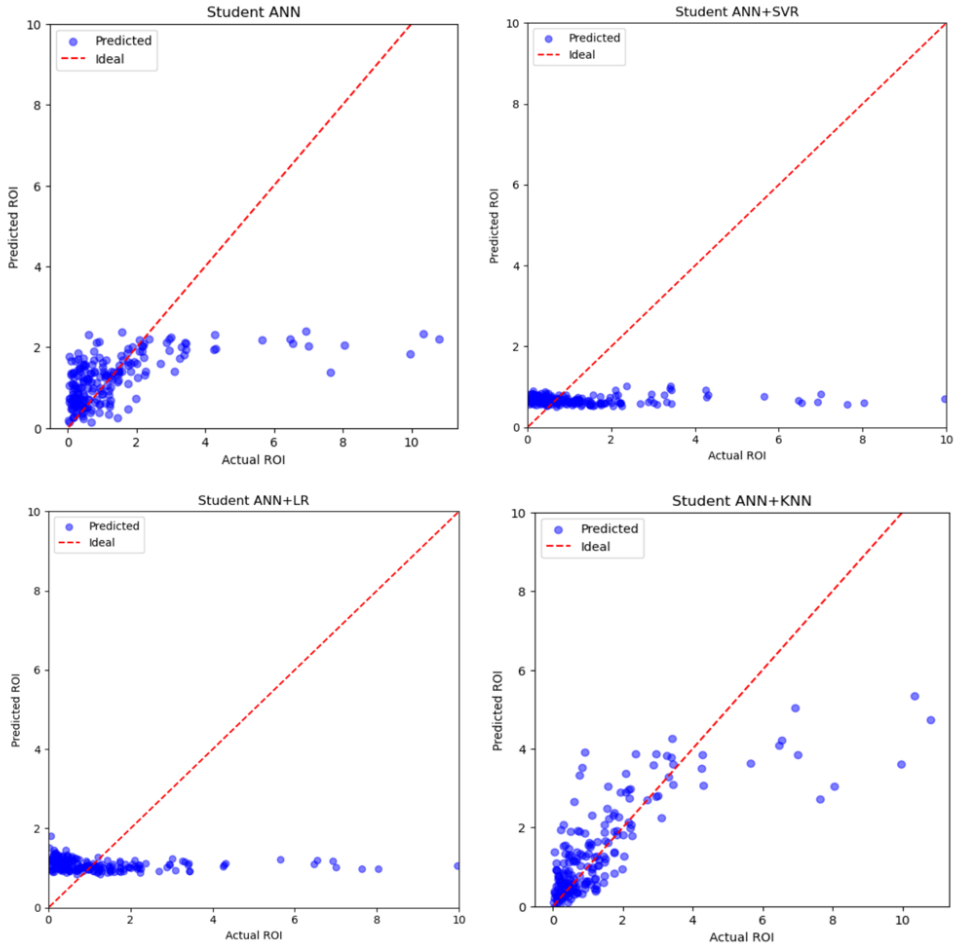


**Figure 9.** The visualization of prediction results among different ensembled models.

## 5. Conclusions

This study presents an efficient framework for user advertising ROI prediction by combining DDP training, knowledge distillation, and ensemble learning techniques. The proposed approach addresses the computational challenges associated with deep learning models, offering a lightweight alternative without significantly compromising accuracy. Experimental results indicate that the student ANN model, generated through knowledge distillation, achieves significantly faster inference times while maintaining acceptable prediction performance. Furthermore, the integration of machine learning models such as KNN, SVR, and LR with high-level features extracted from the student network enhances predictive accuracy. Among the tested configurations, the Student ANN + KNN ensemble achieved the best performance, highlighting the complementary strengths of neural networks and KNN's local pattern-capturing ability. However, the approach is not without limitations. The performance trade-off between accuracy and model simplification remains a challenge, as some complex patterns from the teacher model are lost during distillation. Additionally, the reliance on predefined ensemble models may limit adaptability in dynamic advertising environments. Future research should focus on adaptive ensemble methods and more diverse datasets to further enhance model robustness and generalization. Overall, this framework demonstrates significant potential for practical deployment in real-time advertising ROI prediction tasks, balancing speed, accuracy, and resource efficiency.

## Funding

This research was supported by the National Natural Science Foundation of China under Grant No. 61872364 and 71974036.

**Author Contributions**

Conceptualization, Y.Q. and Z.Z.; writ-ing—original draft preparation and writing—review and editing, Y.Q., A. W. and Z.Z. All authors have read and agreed to the published version of the manuscript.

**Institutional Review Board Statement**

Not applicable.

**Informed Consent Statement**

Not applicable.

**Data Availability Statement**

Data is available upon request from the corresponding author.

**Conflicts of Interest**

The authors declare no conflict of interest.

**References**

1 Bagwell K. The Economic Analysis of Advertising. *Handbook of Industrial Organization* 2007; **3**: 1701–1844.
2 Nelson P. Advertising as Information. *Journal of Political Economy* 1974; **82**(**4**): 729–754.
3 Fitzgerald J. Evaluating Return on Investment of Multimedia Advertising with a Single-Source Panel: A Retail Case Study. *Journal of Advertising Research* 2004; **44**(**3**): 262–270.
4 Danaher PJ, Rust RT. Determining the Optimal Return on Investment for an Advertising Campaign. *European Journal of Operational Research* 1996; **95**(**3**): 511–521.
5 Tang CY, Li C. Examining the Factors of Corporate Frauds in Chinese A-Share Listed Enterprises. *OAJRC Social Science* 2023; **4**(**3**): 63–77.
6 Sheikh M, Conlon S. A Rule-Based System to Extract Financial Information. *Journal of Computer Information Systems* 2012; **52**(**4**): 10–19.
7 Huang W, Ma J. Analysis of Vehicle Fault Diagnosis Model Based on Causal Sequence-to-Sequence in Embedded Systems. *Optimizations in Applied Machine Learning* 2023; **3**(**1**).
8 Ma J, Zhang Z, Xu K, *et al*. Improving the Applicability of Social Media Toxic Comments Prediction Across Diverse Data Platforms Using Residual Self-Attention-Based LSTM Combined with Transfer Learning. *Optimizations in Applied Machine Learning* 2022; **2**(**1**).
9 Zhou Z, Wu J, Cao Z, *et al*. On-Demand Trajectory Prediction Based on Adaptive Interaction Car Following Model with Decreasing Tolerance. In Proceedings of the 2021 International Conference on Computers and Automation (CompAuto), Paris, France, 7–9 September 2021; pp. 67–72.
10 Zhang G, Zhou T, Cai Y. CORAL-based Domain Adaptation Algorithm for Improving the Applicability of Machine Learning Models in Detecting Motor Bearing Failures. *Journal of Computational Methods in Engineering Applications* 2023; **3**(**1**): 1–17.
11 Gan Y, Ma J, Xu K. Enhanced E-Commerce Sales Forecasting Using EEMD-Integrated LSTM Deep Learning Model. *Journal of Computational Methods in Engineering Applications* 2023; **3**(**1**): 1–11.
12 Chen X, Zhang H. Performance Enhancement of AlGaN-Based Deep Ultraviolet Light-Emitting Diodes with AlxGa1-xN Linear Descending Layers. *Innovations in Applied Engineering and Technology* 2023; **2**(**1**): 1–10.
13 Hao Y, Chen Z, Jin J, *et al*. Joint Operation Planning of Drivers and Trucks for Semi-Autonomous Truck Platooning. *Transportmetrica A: Transport Science* 2023; 1–37.
14 Dai, W. Safety Evaluation of Traffic System with Historical Data Based on Markov Process and Deep-Reinforcement Learning. *Journal of Computational Methods in Engineering Applications* 2021; **1**(**1**): 1–14.
15 Dai, W. Evaluation and Improvement of Carrying Capacity of a Traffic System. *Innovations in Applied Engineering and Technology* 2022; **1**(**1**): 1–9.

16 Dai, W. Design of Traffic Improvement Plan for Line 1 Baijiahu Station of Nanjing Metro. *Innovations in Applied Engineering and Technology* 2023; **2**(**1**): 1–11
17 Wenjun D, Fatahizadeh M, Touchaei HG, *et al*. Application of Six Neural Network-Based Solutions on Bearing Capacity of Shallow Footing on Double-Layer Soils. *Steel and Composite Structures* 2023; **49**(**2**): 231–244.
18 Tang Y, Li C. Exploring the Factors of Supply Chain Concentration in Chinese A-Share Listed Enterprises. *Journal of Computational Methods in Engineering Applications* 2023; **3**(**1**): 1–17.
19 Li, C. Analysis of Luxury Goods Marketing Strategies Based on Consumer Psychology. *OAJRC Social Science* 2022; **3**(**5**): 432-443
20 Li C, Tang Y. The Factors of Brand Reputation in Chinese Luxury Fashion Brands. *Journal of Integrated Social Sciences and Humanities* 2023; 1–14.
21 Shen Y, Wang Y, Lu X, *et al*. A Framework for Massive Scale Personalized Promotion. *arXiv* 2021; arXiv: 2108.12100.
22 Lewis R, Wong J. Incrementality Bidding and Attribution. *arXiv* 2022; arXiv:2208.12809.
23 Kong D, Shmakov K, Yang J. Do Not Waste Money on Advertising Spend: Bid Recommendation via Concavity Changes. *arXiv* 2022; arXiv:2212.13923.
24 Nakagawa K, Abe M, Komiyama, J. Ric-nn: A Robust Transferable Deep Learning Framework for Cross-Sectional Investment Strategy. In Proceedings of the 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), Sydney, NSW, Australia, 6–9 October 2020; pp. 370–379.
25 Error MA. Mean Absolute Error. *Retrieved September* 2016; **19**: 14.
26 Error MS. *Mean Squared Error*; Springer: Saugus, MA, USA, 2010; pp. 653–653.
27 Hodson TO. Root Mean Square Error (RMSE) or Mean Absolute Error (MAE): When to Use Them or Not. *Geoscientific Model Development Discussions* 2022; **15**: 5481–5487.
28 Zou J, Han Y, So SS. Overview of Artificial Neural Networks. In *Artificial Neural Networks: Methods and Applications*; Humana Press: Totowa, NJ, USA, 2009; pp. 14–22.
29 Yegnanarayana B. *Artificial Neural Networks*; PHI Learning Pvt. Ltd.: Delhi, India, 2009.
30 Krenker A, Bešter J, Kos, A. Introduction to the Artificial Neural Networks. In *Artificial Neural Networks: Methodological Advances and Biomedical Applications*; InTechOpen: London, UK, 2011; pp. 1–18.
31 Kramer O, Kramer O. K-Nearest Neighbors. In *Dimensionality Reduction with Unsupervised Nearest Neighbors*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 13–23.
32 Laaksonen J, Oja E. Classification with Learning k-Nearest Neighbors. In Proceedings of International Conference on Neural Networks (ICNN'96), Washington, DC, USA, 3–6 June 1996; Volume 3, pp. 1480–1483.
33 Zhang Z. Introduction to Machine Learning: K-Nearest Neighbors. *Annals of Translational Medicine* 2016; **4**(**11**): 218.
34 Awad M, Khanna R, Awad M, *et al*. Support Vector Regression. In *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*; Apress: Berkeley, CA, USA, 2015; pp. 67–80.
35 Smola AJ, Schölkopf B. A Tutorial on Support Vector Regression. *Statistics and Computing* 2004; **14**: 199–222.
36 Zhang F, O'Donnell LJ. Support Vector Regression. In *Machine Learning*; Academic Press: Cambridge, MA, USA, 2020; pp. 123–140.
37 Su X, Yan X, Tsai CL. Linear regression. *Wiley Interdisciplinary Reviews: Computational Statistics* 2012; **4**(**3**): 275–294.
38 Montgomery DC, Peck EA, Vining GG. *Introduction to Linear Regression Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2021.
39 James G, Witten D, Hastie T, *et al*. Linear Regression. In *An Introduction to Statistical Learning: With Applications in Python*; Springer International Publishing: Cham, Switzerland, 2023; pp. 69–134.
40 Li S, Zhao Y, Varma R, *et al*. Pytorch Distributed: Experiences on Accelerating Data Parallel Training. *arXiv* 2020; arXiv:2006.15704.
41 Zhao Y, Gu A, Varm R, *et al*. Pytorch FSDP: Experiences on Scaling Fully Sharded Data Parallel. *arXiv* 2023; arXiv:2304.11277.

42 Mishra, P. Distributed PyTorch Modelling, Model Optimization, and Deployment. In *PyTorch Recipes: A Problem-Solution Approach to Build, Train and Deploy Neural Network Models*; Apress: Berkeley, CA, USA, 2022; pp. 187–212.

43 Gou J, Yu B, Maybank SJ, *et al*. Knowledge Distillation: A Survey. *International Journal of Computer Vision* 2021; **129**(**6**): 1789–1819.

44 Phuong M, Lampert C. Towards Understanding Knowledge Distillation. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 5142–5151.

45 Chen P, Liu S, Zhao H, *et al*. Distilling Knowledge via Knowledge Review. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 5008–5017.

46 Webb GI, Zheng Z. Multistrategy Ensemble Learning: Reducing Error by Combining Ensemble Learning Techniques. *IEEE Transactions on Knowledge and Data Engineering* 2004; **16**(**8**): 980–991.

47 Ovelgönne M, Geyer-Schulz A. An Ensemble Learning Strategy for Graph Clustering. *Graph Partitioning and Graph Clustering* 2012; **588**: 187.

48 Huang F, Xie G, Xiao R. Research on Ensemble Learning. In Proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence, Shanghai, China, 7–8 November 2009; Volume 3, pp. 249–252.