ADVANCED
RESEARCH & INNOVATIONS
International Journal of Advanced Research & Innovations

Volume 15, Issue-1
March, 2025, P20
ISSN: 2319-9253

THE DISTRIBUTED KNOWLEDGE DISTILLATION FRAMEWORK FOR FINANCIAL FRAUD PREVENTION BASED ON TRANSFORMER

#1CHINTAKUNTA CHENNAKESAVA REDDY, PG GRADUATE
#2H. MADHUSUDHANA RAO, Assistant Professor,
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT.
KANDULA OBUL REDDY MEMORIAL COLLEGE OF ENGINEERING, KADAPA

ABSTRACT Financial fraud cases causing serious damage to the interests of investors are not uncommon. As a result, a wide range of intelligent detection techniques are put forth to support financial institutions' decision-making. Currently, existing methods have problems such as poor detection accuracy, slow inference speed, and weak generalization ability. Therefore, we suggest a distributed knowledge distillation architecture for financial fraud detection based on Transformer. Firstly, the multi-attention mechanism is used to give weights to the features, followed by feed-forward neural networks to extract high-level features that include relevant information, and finally neural networks are used to categorize financial fraud. Secondly, for the problem of inconsistent financial data indicators and unbalanced data distribution focused on different industries, a distributed knowledge distillation algorithm is proposed. This algorithm combines the detection knowledge of the multi-teacher network and migrates the knowledge to the student network, which detects the financial data of different industries. The final experimental results show that the proposed method outperforms other methods in terms of F1 score (92.87%), accuracy (98.98%), precision (81.48%), recall (95.45%), and AUC score (96.73%) when compared to the traditional detection methods.

INDEX TERMS Transformer, knowledge distillation, financial fraud detection.

I. INTRODUCTION

The number of listed firms is increasing quickly due to the ongoing social economy development, and their place in the global economy is vital. However, cases of financial fraud are frequent and prohibited, causing great losses to the majority of investors and arousing discussions in all sectors of society. In China, the number of criminals involved in financial counterfeiting activities in 2019 exceeded 961, with a total value of more than US 8 billion [1]. Numerous investors' faith has been damaged by these instances, which has had a detrimental impact on the capital markets and increased financial market volatility [2], [3]. In order to address these counterfeiting issues, the development of new detection methods is imperative. Currently, there are two main means of detecting counterfeiting by listed companies: one is to audit and analyze the company's financial data, and the other is to detect whether there is any suspicion of counterfeiting through big data-driven machine learning algorithms [4]. Manual audits and reviews of publicly traded corporations' financial statements are examples of traditional financial analysis techniques, however they are expensive, time-consuming, and prone to error [4]. These methods are not absolute, and as the methods of financial fraud continue to evolve, it is difficult for practitioners to detect new patterns of fraud. Also, certain anomalies may be legitimate business practices, rendering such methods less feasible. Then, big data-driven machine learning algorithms were used to detect financial fraud, an area where computers were more adept at data analysis than people when dealing with large amounts of data, particularly when it came to highdimensional features. The effectiveness of machine learning models in financial forgery detection has been demonstrated in the literature [5]. But forgers continue to innovate and adopt new concealment methods, making it difficult for and identify new forgery methods in a timely manner, and correlations between data features are difficult to be learned by the models. It is difficult for the model to extract the more critical information for the task at hand from the complex and large data features, resulting in the performance of existing counterfeiting detection models being greatly limited.More significantly, by identifying the relationships between features, the attention model can uncover more concealed counterfeiting information and investigate more counterfeiting patterns. For example, literature [6] proposes a two-level attention model that captures deep representations of features from data sample level and feature level sets, respectively.

The associate editor coordinating the review of this manuscript and

Existing financial fraud detection methods are mostly based on machine learning and deep learning algorithms [4]. These techniques pay less attention to the internal correlations within financial data and instead concentrate on mining the fundamental features of the data. Additionally, different industries may encounter varying challenges in financial data fraud, and the internal correlations of financial data features differ across industries. Furthermore, with the continuous growth in the scale of financial data, these models become increasingly deep and complex, resulting in issues such as model bloat and slow inference speed. Therefore, how to effectively mine the internal correlation of financial data, compress the model size, and enhance the model's ability to detect financial data falsification in different industries is a new direction for researchers to explore. To address the above problems, this research suggests a distributed knowledge distillation architecture based on Transformer. The method uses a multi-attention mechanism to extract the internal correlation of the data, and then the highlevel features that contain the information related to the financial data are extracted through a forward neural network, which is combined with the neural network to classify the financial data fraud. Secondly, to address the problem of inconsistent financial data indicators and unbalanced data distribution focused on different industries, and to reduce the complexity of the financial fraud detection model and improve the accuracy of the model, this paper proposes a distributed knowledge distillation algorithm. The algorithm migrates the detection knowledge of the multi-teacher network to the student network separately, and the student network detects the financial data of different industries. The final experimental results show that the proposed method has better F1 score, accuracy, precision, recall, and AUC score compared to the traditional detection methods, which improves the accuracy of financial forgery detection.

The following are the primary contributions of our research:

(1) For financial fraud detection, considering that Transformer has strong generalization and expressive ability, it is easier to adapt to diverse financial data. Therefore, we propose a financial fraud detection model based on Transformer, which utilizes the multi-head attention mechanism and feedforward neural network to mine the high-level features that incorporate the relevant information of financial data, thus improving the characterization of data relevance.

(2) To address the problem of inconsistent financial data indicators and unbalanced data distribution focused on different industries, and to reduce the complexity of the financial fraud detection model and improve the accuracy of the model, this paper proposes a distributed knowledge distillation algorithm. The algorithm migrates the detection knowledge of the multi-teacher network to the student network separately, and the student network detects the financial data of different industries.

(3) The proposed distributed network was evaluated on the dataset of the 9th ‘‘TipDM Cup’’ listed company financial analysis competition. Experimental results demonstrate that our proposed financial fraud detection method based on Transformer with distributed knowledge distillation outperforms traditional tree models and ensemble models in key performance metrics on the dataset. This confirms the feasibility and effectiveness of our proposed method.

The rest of the paper is structured as follows, the second part is a review of related research, the third part introduces our proposed model for financial fraud detection, the fourth part describes the distributed knowledge distillation framework for detecting fraudulent data in different industries, and the experimental results are discussed in the fifth part. Finally Part VI summarizes the conclusions of this study.

II. BACKGROUND AND RELATED WORK

A. *TRADITIONAL FINANCIAL FRAUD DETECTIONMETHODS*

Financial fraud detection technology can lower investor losses, preserve equity and justice in the trading market, and assist the China Securities Regulatory Commission (CSRC) in determining if listed businesses are suspected of fraud. Traditional approaches for determining a listed firm's involvement in fraudulent operations rely on analyzing financial data, information from listed firms, and third-party evidence. With the continuous development of science and technology, detection methods for fraud have also made significant progress. Artificial intelligence technologies driven by big data have been widely applied and have shown promising results in fraud detection. The core idea ofartificial intelligence is to train a model with strong generalization capabilities, supported by big data, enabling the model to accurately detect the likelihood of listed companies engaging in financial data fraud. According to whether the sample data is labeled, these methods can be roughly divided into two categories: supervised learning and unsupervised learning.

In a supervised learning approach, the model used for financial forgery detection can be viewed as a binary classification task, i.e., whether the company is a forgery or not, and the result is often given in the form of a probability, where the higher the probability the more likely it is that the company is a forgery. Many classification algorithms have been proposed and have achieved good results in various industries. Based on whether the distribution of observed variables is modeled, supervised learning models can be divided into two categories: discriminative models and generative models. Generative models include Naive Bayes (NB), Restricted Boltzmann Machine (RBM), Hidden Markov Model (HMM). Discriminative models include Logistic Regression (LR), Multilayer Perceptron (MLP), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Maximum Entropy Model (ME), Conditional Random Field (CRF), Decision Tree (DT), Random Forest (RF). Such as, in reference [7], the accuracy of four machine learning algorithms–LR, RF, DT,CatBoost–is analyzed and compared as the subject of financial fraud detection is explored through the use of several algorithms. Using a dataset of financial fraud, Liu et al. used the RF technique and contrasted it with other algorithms like LR, KNN, DT, and SVM. They discovered that the RF algorithm had the best interpretability and maximum accuracy [8]. Unsupervised learning does not require labeling the data; it is similar in nature to a statistical tool that detects anomalous data to determine if samples that do not belong to the main class are deceptive. Two common types of algorithms for unsupervised learning are clustering and dimensionality reduction. The clustering algorithms are K-mean clustering, hierarchical clustering, etc., and the dimensionality reduction algorithms are Principal Component Analysis(PCA) and Singular Value Decomposition(SVD). Such as, reference [9] proposed a model framework that separates clusters using the K-means method and compared the performance with two of the most important financial fraud detection systems. Reference [10] introduced an unsupervised learning approach that combines Particle Swarm Optimization(PSO) and K-Means clustering, demonstrating better performance in financial fraud detection compared to K-Means.

B. *DEEP LEARNING COUNTERFEIT DETECTION METHODS*

Classical machine learning algorithms typically use shallow models, effective for linearly separable tasks or simple non-linear tasks. In contrast, deep learning algorithms are generally employed for deep models, providing stronger nonlinear modeling capabilities and better performance on realworld complex tasks. For tasks with higher complexity and deeper concealment, such as financial data fraud detection, deep learning algorithms generally outperform machine learning algorithms [4]. For example, Rushin et al. compared the performance of LR, gradient boosting trees, and deep learning in detecting credit card fraud, indicating that deep learning methods outperform the other two approaches [11]. In addition, deep learning algorithms can deeply explore the potential connections between data, thereby uncovering more methods for detecting financial fraud and enhancing the effectiveness of detection. For example, the classification results depend on features constructed from domain-specific knowledge, without considering other attributes of the data, such as temporal attribution. Jurgovsky et al. treated fraud detection as a sequence classification task and utilized Long Short-Term Memory (LSTM) for predictions.Experimental results show that LSTM effectively improves the accuracy of credit card fraud compared to random forest [12]. Zhou et al. use a graph embedding algorithm to learn topological features from financial network graphs and represent them as low-dimensional dense vectors. In this way, they utilize deep neural networks to intelligently and efficiently classify and predict data samples from large-scale datasets [13]. The literature [14], taking into account the homogeneity of the data structure, proposes a graph learning algorithm capable of learning topological features and transaction amount features in financial transaction network graphs. In literature [15], a novel graph neural network (GNN) architecture with a time de-biasing constraint based on adversarial loss is proposed. This architecture captures fraud patterns that exhibit fundamental consistency over time and performs well in fraud detection tasks. In literature [16], a new credit card fraud detection model named CCFDNet is introduced, featuring a hybrid architecture combining 1D-Conv and Residual Neural Network (Res-net). This model demonstrates good effectiveness and robustness in credit card fraud detection.

C. *MULTI-TEACHER KNOWLEDGE DISTILLATIONMETHODS*

The single-student-multi-teacher distillation paradigm has made significant progress in converting complicated, multi-attribute instructor information into lightweight student networks. Multi-teacher distillation research focuses on designing appropriate distillation strategies for use in instructing students. In 2017, You et al [17] proposed a framework for multi-teacher distillation. This approach averages the soft labels of logits produced from several teacher models and provides them to student models for learning. Shi et al [18] used another way of directly splicing logits of multiple teachers and then performing PCA dimensionality reduction on the face recognition model. Shin [19] extended the multi-instructor-single-student distillation architecture to a visual multi-attribute recognition task of a target, where each instructor specialises in learning one attribute, and then synthesises the multi-instructor’s knowledge to transfer it to thestudenttoachievethestudent’smulti-attributerecognition learning. Furthermore, in a recent study, Hailin et al. [20] proposed an adaptive multi-instructor knowledge distillation strategy that allows diverse instructor knowledge to be jointly utilised to improve student performance. The multiinstructor knowledge distillation paradigm proposed in the literature [21] empowers students to integrate and capture a variety of knowledge from different sources. Although many studies have used a multi-teacher distillation framework, less attention has been paid to the uneven distribution of positive and negative samples. In this research, we employ a multiteacher knowledge distillation strategy to aggregate various instructors' knowledge of financial fraud detection across industries onto a lightweight student model. The goal is to enhance the model's performance in detecting imbalances in the distribution of positive and negative data using a simple and effective multi-teacher distillation architecture. One distinction between our technique and other multi-teacher approaches is that our multi-teacher model learns about financial fraud in different industries separately, whereas our student network learns about financial fraud in each industry from all of the teacher models, allowing the model to be generalized efficiently in the presence of an imbalanced data distribution.

Machine learning techniques are heavily used in the field of financial fraud detection, and graph networkbased approaches have made significant progress in recent years [4]. However, these methods only focus on the topological features and data features of the network, ignoring the dependencies between data features. Table 1 summarises the existing work related to our problem, compared to other methods, the method proposed in this paper exploits the dependencies between financial indicators for forgery detection, and uses multi-instructor distributed knowledge distillation to improve the speed of model inference and the generalisation of the model when the data is unbalanced. And these are not available in other models.

III. FINANCIAL FRAUD DETECTION MODEL BASED ON TRANSFORMER

A. FINANCIAL FRAUD DETECTION METHODS AND PROCESSES

Transformer is an advanced deep learning model which was first proposed by Vaswani et al. in 2017 and was initially used for natural language processing tasks [22]. However, due to its robust parallelism and expressive capabilities, it has been successfully applied to other domains, including the fields of image processing and classification.

One of Transformer's basic features is the self-attention mechanism, which allows the model to process all points in the input sequence at once rather than step-by-step like a recurrent neural network or convolutional neural network. The self-attention mechanism enables the model to capture correlations by assigning different attentional weights to different sections of the input sequence. To better capture various sorts of relationships, the self-attention mechanism is expanded to several attention heads, each capable of learning varied attention weights. The structure of the Transformer encoder is shown in Figure 1. The encoder typically includes a multi-head attention layer, a feed-forward neural network layer, residual connectivity, and layer normalization. Transformer are usually made up of multiple encoders and decoders stacked on top of each other, and these stacked layers help the model learn complex feature representations.

To enhance the accuracy of data analysis and modeling, the financial dataset is first preprocessed. Subsequently, multiple attention scores are calculated for financial data to obtain a representation of the correlation between features. These multiple attention scores are then fed through a feedforward network to extract higher-level features that integrate relevant information more comprehensively. Following this, a neural network maps these higher-level features to the probability of fraud output. Finally, the cross-entropy loss of the samples is computed, and the model parameters are updated through gradient descent based on the loss value.

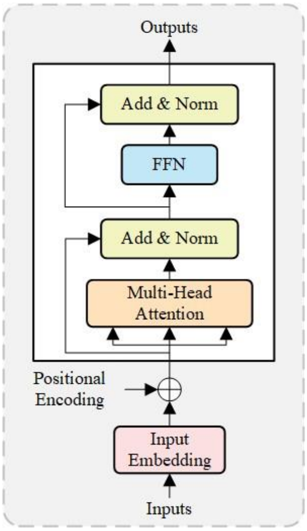


FIGURE 1. Transformer encode block.

B. MODEL ARCHITECTURE

The architecture of the financial fraud detection model based on Transformer is illustrated in Figure 2. The model consists of three modules. The first module is a multi-industry data processing module. The second module is a Transformer Encode Block module, which includes a multi-head attention module and a fully connected feedforward neural network. The feedforward network comprises a linear transformation, ReLU non-linear activation function, along with a residual connection and layer normalization operation. The third module is the output neural network module, containing a linear neural network for output and a softmax function for result normalization.

1) MULTI-HEAD ATTENTION

The financial dataset is represented as $D = \{(X_n,Y_n)\}^{N}_{n=1}$, where the matrix $X = \{x_1,x_2,\dots,x_m\}$ represents financial data features. Here, x_m is a vector of dimension d_{model},$Y = \{y_1,y_2,...,y_m|y_m \in [0,1]\}$, where 0 indicates no fraud and 1 indicates fraud. For a single sample X,the first step involves computing the self-attention scores for its features. Here, we define three matrices for the scaled dotproduct operation:Query(Q),Key(K),and Value(V). Additionally, three learnable weight matrices W_q,W_k,W_v are introduced to map each input feature to query, key, and value vectors:

$$Q = XW_q \quad (1)$$

$$K = XW_k \quad (2)$$

$$V = XW_v \quad (3)$$

where $Q \in Rm\times d,K \in Rm\times d,V \in Rm\times d,Wq \in Rm\times d,Wk \in Rm\times d,W_v \in Rm\times d$.

Then, for the query matrix Q, calculate its similarity score matrix S with the key matrix K. To prevent excessively large scores that could lead to model gradient explosions, divide√ each score by d:

$$S = \frac{QK^T}{\sqrt{d}} \quad (4)$$

where $S \in \mathrm{R}^{m\times m}$ the scores represent the correlation between each financial data feature and other features.

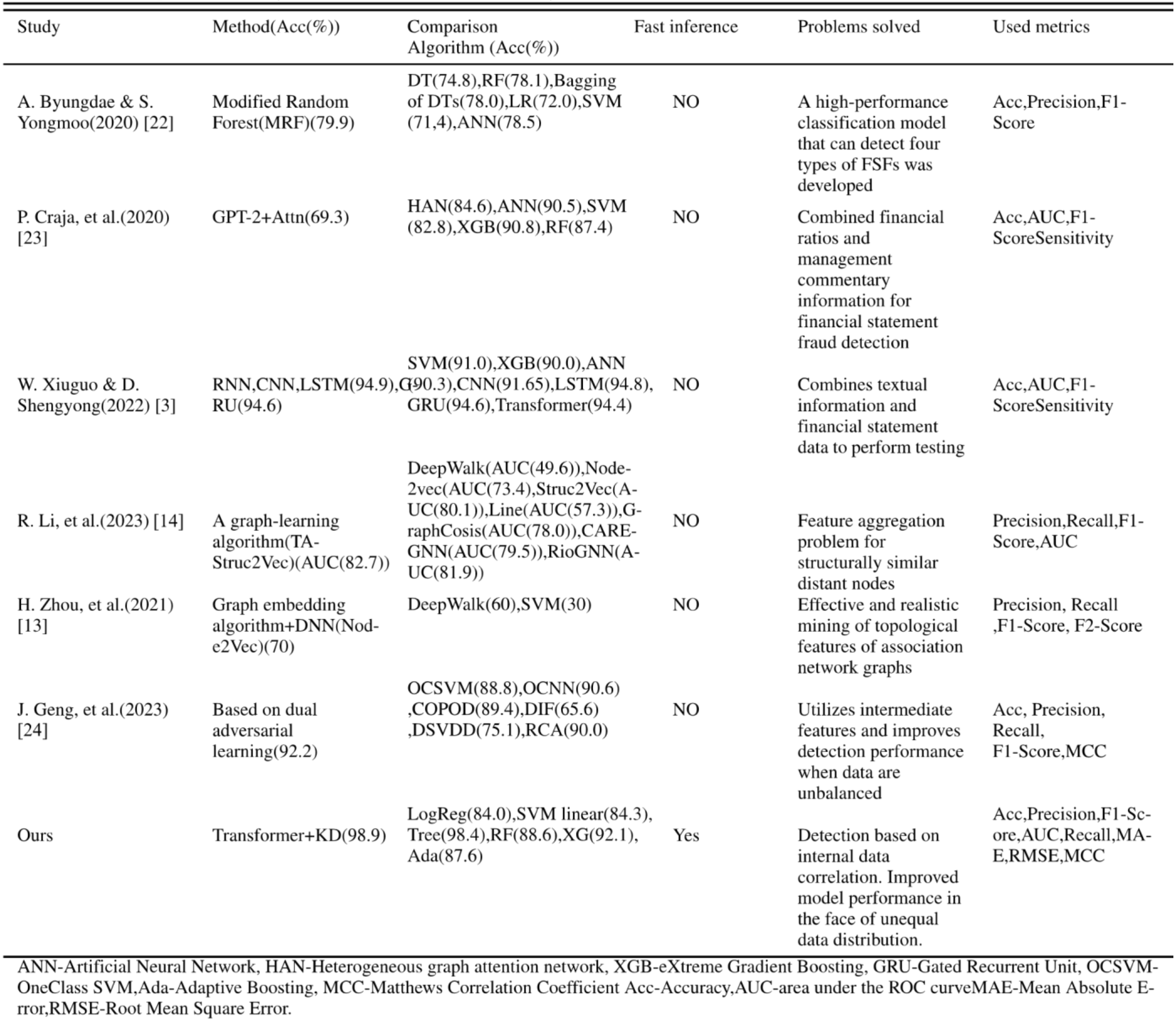
Finally, normalize the scores using the softmax function and multiply the normalized correlation scores by the value matrix V to obtain the self-attention scores O for financial data features:

$$O = softmax(S)V \quad (5)$$

where $O \in \mathrm{R}^{m\times d}$.

The self-attention scores for financial data features can be summarized as formula (6):

TABLE 1. Comparative analysis of methods used to falsify financial statements.



Study	Method(Acc(%))	Comparison Algorithm (Acc(%))	Fast inference	Problems solved	Used metrics
A. Byungdae & S. Yongmoo(2020) [22]	Modified Random Forest(MRF)(79.9)	DT(74.8),RF(78.1),Bagging of DTs(78.0),LR(72.0),SVM (71,4),ANN(78.5)	NO	A high-performance classification model that can detect four types of FSFs was developed	Acc,Precision,F1-Score
P. Craja, et al.(2020) [23]	GPT-2+Attn(69.3)	HAN(84.6),ANN(90.5),SVM (82.8),XGB(90.8),RF(87.4)	NO	Combined financial ratios and management commentary information for financial statement fraud detection	Acc,AUC,F1-ScoreSensitivity
W. Xiuguo & D. Shengyong(2022) [3]	RNN,CNN,LSTM(94.9),GRU(94.6)	SVM(91.0),XGB(90.0),ANN (90.3),CNN(91.65),LSTM(94.8), GRU(94.6),Transformer(94.4)	NO	Combines textual information and financial statement data to perform testing	Acc,AUC,F1-ScoreSensitivity
R. Li, et al.(2023) [14]	A graph-learning algorithm(TA-Struc2Vec)(AUC(82.7))	DeepWalk(AUC(49.6)),Node-2vec(AUC(73.4),Struc2Vec(A-UC(80.1)),Line(AUC(57.3)),G-raphCosis(AUC(78.0)),CARE-GNN(AUC(79.5)),RioGNN(A-UC(81.9))	NO	Feature aggregation problem for structurally similar distant nodes	Precision,Recall,F1-Score,AUC
H. Zhou, et al.(2021) [13]	Graph embedding algorithm+DNN(Nod-e2Vec)(70)	DeepWalk(60),SVM(30)	NO	Effective and realistic mining of topological features of association network graphs	Precision, Recall ,F1-Score, F2-Score
J. Geng, et al.(2023) [24]	Based on dual adversarial learning(92.2)	OCSVM(88.8),OCNN(90.6) ,COPOD(89.4),DIF(65.6) ,DSVDD(75.1),RCA(90.0)	NO	Utilizes intermediate features and improves detection performance when data are unbalanced	Acc, Precision, Recall, F1-Score,MCC
Ours	Transformer+KD(98.9)	LogReg(84.0),SVM linear(84.3), Tree(98.4),RF(88.6),XG(92.1), Ada(87.6)	Yes	Detection based on internal data correlation. Improved model performance in the face of unequal data distribution.	Acc,Precision,F1-Score,AUC,Recall,MAE,RMSE,MCC

ANN-Artificial Neural Network, HAN-Heterogeneous graph attention network, XGB-eXtreme Gradient Boosting, GRU-Gated Recurrent Unit, OCSVM-OneClass SVM,Ada-Adaptive Boosting, MCC-Matthews Correlation Coefficient Acc-Accuracy,AUC-area under the ROC curveMAE-Mean Absolute Error,RMSE-Root Mean Square Error.

$$\mathit{Attention}(Q,K,V) = \mathit{softmax}\frac{QKT}{\sqrt{d}}V. \qquad (6)$$

The multi-head attention mechanism enables the model to capture richer correlations among financial data features, facilitating a more in-depth exploration of patterns related to data falsification. Multi-head attention involves performing the self-attention mechanism multiple times, essentially having *n* individuals focusing attention on different positions of financial data features. This approach increases the likelihood of detecting crucial information related to data falsification:

$$\mathit{MultiHeadAtt}\ (Q,K,V) = \mathit{Concat}\ (\mathit{head}_1,\mathit{head}_2,\ldots,\mathit{head}_h)\cdot W_o \qquad (7)$$

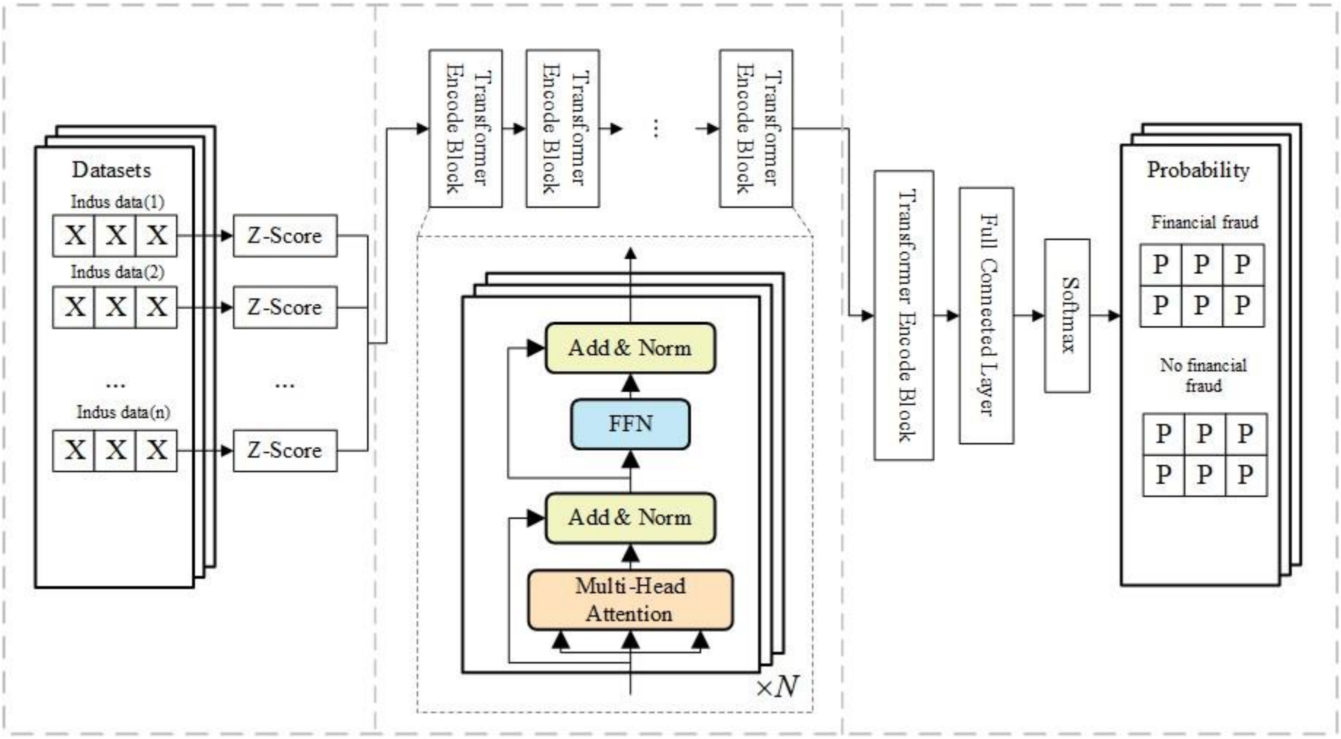


FIGURE 2. The architecture of the financial fraud detection model based on transformer.

where *head* = $\mathit{Attention}(Q_i,K_i,V_i), i \in \{1,\ldots,h\}, W_o \in Rhd\times d$.

2) FEEDFORWARD NEURAL NETWORK

The multi-head attention scores obtained from formula (7) undergo a residual connection and layer normalization operation. The residual connection addresses the training issues of deep networks by adding the output to the original input,enhancingthenetwork'srepresentationalcapacity[25]. Layer normalization normalizes all inputs to have a mean of 0 and a standard deviation of 1. This helps alleviate the problem of internal covariate shift in neural network training, providing more stable and faster training:

$$\mathit{LayerNorm}(X + \mathit{MultiHeadAtt}\ (Q,K,V)). \qquad (8)$$

Subsequently, the multi-head attention scores, after the residual connection and layer normalization, undergo further processing through two linear transformations and a ReLU activation function. This step aims to extract higher-level features with richer contextual information:

$$\mathit{FFN}\ (X) = \max(0,XW_1 + b_1)W_2 + b_2 \qquad (9)$$

while the linear transformations at different positions in the encoder are the same, the parameters between layers are distinct.

In order to prevent overfitting, we introduce dropout into the output of each fully connected layer to ensure the model's generalization. Dropout involves randomly discarding each neuron with a probability *p*. For the neurons that are not discarded, their values are scaled by the reciprocal of the dropout probability, maintaining the expected value of the data. By training different network structures in each iteration, dropout introduces variability, eliminating and weakening the interdependence among neuron nodes, thereby enhancing the model's ability to generalize internal correlations in financial data. The dropout computation process is as follows formula (10).

$$dropout\,(X) = \begin{cases} 0, & p \\ \dfrac{X}{1-p}, & 1-p \end{cases} \qquad (10)$$

3) OUTPUT NEURAL NETWORK

After the financial data goes through the stacked encoder, we map and output the high-level features *X*, which are extracted by the last encoder and contain internal correlation information, through a linear layer. We normalize the output using the softmax function. The normalization calculation is shown in formula (11):

$$Y^{pre} = softmax\; W \cdot X^T + b \qquad (11)$$

where $Y^{pre} \in \mathrm{R}^{1\times 2}$ is the probability distribution vector, *W* is the neural network weight matrix, and *b* is the bias vector.

4) OVERALL LOSS CALCULATION

The financial dataset $D = \{(X_n,Y_n)\}^N{}_{n=1}$ is passed into the Transformer-based financial fraud detection model. After extracting high-level features related to the data, the model maps the samples to predicted label values *f (W,X)*. The true label values Y_n and the predicted label values *f (W,X)* are then used to calculate the cross-entropy loss through formula (12):

$$f_{cls}\,(W,X_n,Y_n) = -[Y_n \cdot \log(f\,(W,X_n)) + (1 - Y_n) \cdot \log(1 - f\,(W,X_n))] \qquad (12)$$

where *W* represents the model's parameter matrix, *f (W,X)* represents the mapping of feature *X* through the model's parameter matrix *W*, and its value is the probability of no fraud.

The model utilizes data samples for training to update the model parameters *W*. Here, we provide the general formula for parameter updates:

$$W = W + \eta \cdot \frac{\partial F_{cls}\,(W)}{\partial W} \qquad (13)$$

where η represents the learning rate, and $F_{cls}\,(W)$ represents the total loss function of the financial dataset D. Its calculation formula is as follows:

$$F_{cls}\,(W) = \frac{1}{N} \sum_{X_n,Y_n \in D} f_{cls}\,(W,X_n,Y_n). \qquad (14)$$

IV. DISTRIBUTED KNOWLEDGE DISTILLATION DETECTION FRAMEWORK

On the one hand, due to the presence of various challenges related to financial data manipulation in different industries, there exist distinct characteristics and internal correlations in the financial data of different industries. Moreover, there are significant differences in the financial data indicators that different industries focus on. Therefore, it is challenging to use a universal model to detect financial data with such substantial variations. On the other hand, traditional models suffer from issues such as complex structures, deep model depths, and slow inference speeds, making it difficult to deploy them in practical application scenarios. Based on the above problem considerations, this paper uses a distributed architecture to train multiple teacher detection models for multiple industries. And a distributed knowledge distillation algorithm is proposed to migrate the detection knowledge from the multi-teacher network to the lightweight student network separately. On the one hand, the detection model is compressed to adapt to practical application scenarios, and on the other hand, the generalisation ability of the model in the case of unbalanced data distribution is improved.

The distributed knowledge distillation detection framework, as shown in Figure 3, is illustrated as follows. Firstly, datasets from various industries are prepared, and these datasets are utilized to train teacher models. Subsequently, untrained student models with simpler structures than the teacher models are prepared. A knowledge distillation algorithm is used so that the knowledge from the multiteacher model is migrated separately to the student network, which finally tests the financial data from different industries.

A. MULTI-TEACHER MODEL

The knowledge distillation algorithm is a model compression technique. It involves transferring knowledge from a large model (usually referred to as the teacher model) to a smaller model (typically known as the student model), with the aim of retaining the performance of the teacher model on a relatively smaller scale student model [26].

The teacher model adopts the Transformer-based financial fraud detection model mentioned in Section III. The multiindustry financial dataset is represented as the set $I = \{D_1,D_2,\ldots,D_m\}$ where $D_m = \{(X_n,Y_n)\}^N{}_{n=1}$ represents the financial dataset of industries such as manufacturing and transportation. The Multi-teacher model is trained using the collection of multi-teacher financial datasets. The performance of the Multi-teacher model is further optimized by adjusting hyperparameters. The training of the Multi-teacher model is illustrated in Algorithm 1.

B. STUDENT MODEL

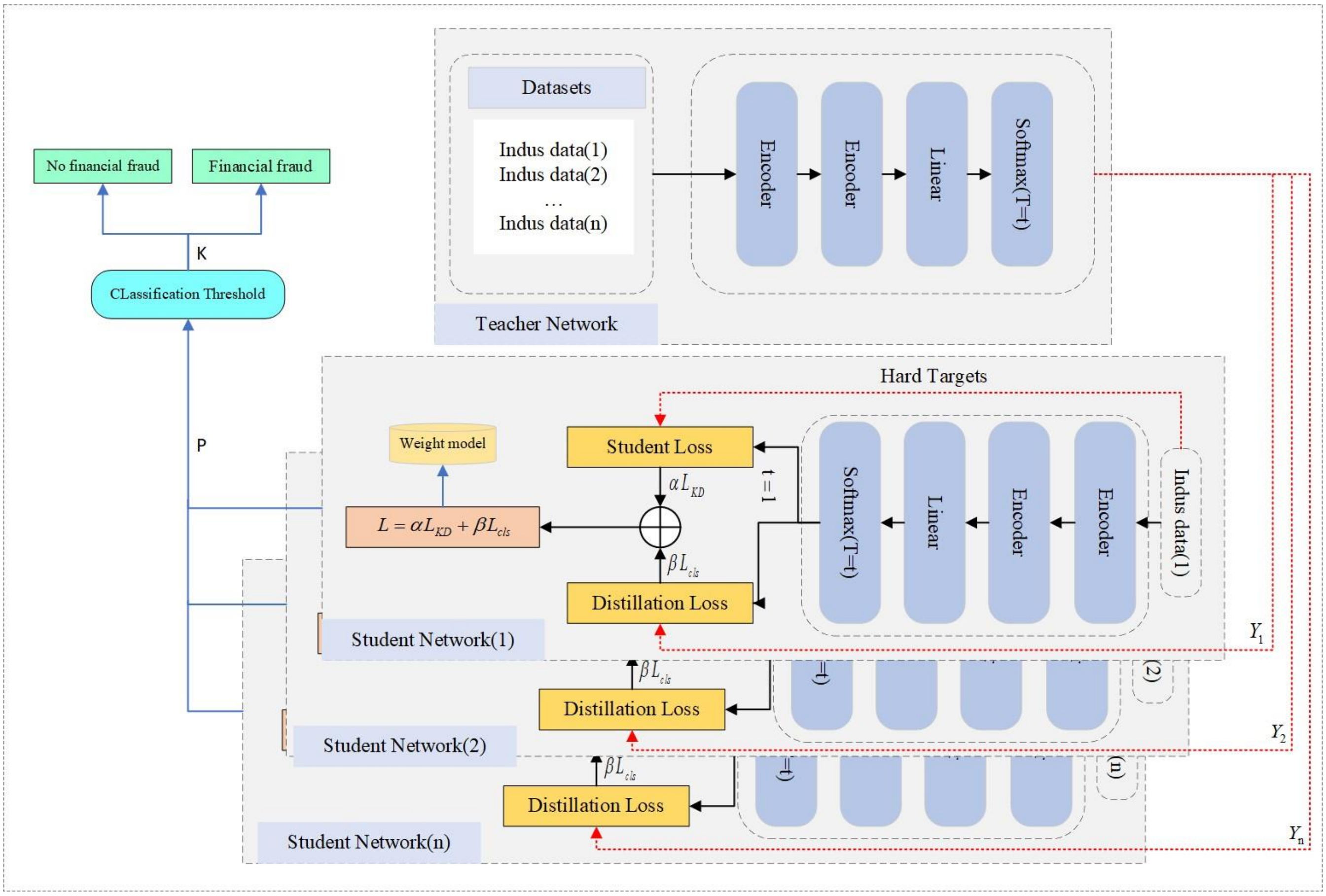


FIGURE 3. Distributed knowledge distillation framework for financial data detection.

For a classification task, the final output of the model is the probabilities for each class, which are referred to as soft targets. Thetrue labels for each sample are called hardtargets. The difference with hard targets is that soft targets not only inform us about the most likely class for a sample but also provide probabilities for other classes, indicating that soft targets contain more information than hard targets. Therefore, when training the Multi-teacher model, we use hard targets. Thepredictionsobtainedfromtrainingtheteachernetworkon asamplecanconveymoreinformationtothestudentnetwork. Consequently, we can use the soft targets from the teacher network to guide the training of the student network.

The student network adopts a smaller Transformer-based financial fraud detection model with fewer parameters. For the financial dataset $D = \{(X_n,Y_n)\}^N_{n=1}$ let $Z^{(t)} \in R^{B\times C}$ and $Z^{(s)} \in R^{B\times C}$ represent the logits output by the teacher network and student network, respectively, where B is the batch size, and C is the number of categories. $Y \in [0,1]$ represents the hard targets for the samples. After applying the softmax function to the outputs $Z^{(t)} \in R^{B\times C}$ and $Z^{(t)} \in R^{B\times C}$ of the teacher and student networks, the probability distributions range from 0 to 1. If we find that the relative sizes between the categories are not sufficiently distinct, we introduce a distillation temperature T. A higher T makes the relative sizes between the categories more pronounced. The introduction of T involves dividing the original softmax values by T. In theory, as T increases,the distillation effect improves, but excessively large T can cause the relative sizes between categories to disappear. Therefore, it's necessary to choose an appropriate value for T. The distillation process is represented as formula (15):

T = exp *ziT* (15) *softmax Z*

P exp *zjT j*

where $Z = \{z_1,z_2,\ldots,z_n\}$.

The guidance of the teacher model in training the student model involves two steps. The first step is to compute the distillation loss. This involves using the distillation formula (16) and formula (17)to calculate the soft targets $P^{(t)}$ and $P^{(s)}$ from the outputs $Z^{(t)}$ and $Z^{(s)}$ of the teacher and student networks, respectively. Then, the KL divergence loss between these soft targets is calculated using formula (18):

$$P(t) = softmax\ YTpreT \qquad (16)$$

$$P(s) = softmax\ YSpreT \qquad (17)$$

$$L_{KD} = \frac{T^2\ XB\ XC}{B \quad i{=}1 \quad j{=}1} \log \frac{p(_{i},s_j)!\ .}{(t)\ pi,j} \qquad (18)$$

The second step is to compute the student loss. This involves using a temperature softmax distiller (with T = 1) on the output $Z^{(s)}$ of the student network to calculate soft targets P,and then calculating the cross-entropy loss between $P^{(t=1)}$ and the hard targets Y_n from the financial data using formula (19):

$$- \qquad Lcls = 1\ XB\ XC$$

$$\frac{}{B \quad i{=}1 \quad j{=}1} -[Yi,j \cdot \log Pi,j$$

$$+\ 1 - Y_{i,j} \cdot \log 1 - P_{i,j}]. \qquad (19)$$

The final knowledge distillation loss is obtained by taking the weighted sum of both the distillation loss and the student loss:

$$L_{tr} = \alpha \cdot L_{cls} + \beta \cdot L_{KD}. \qquad (20)$$

where α and β are weight coefficients, determining the contribution of each loss term in the final knowledge distillation loss.

The model utilizes data samples for training to update model parameters W.The specific algorithm for model training is shown in Algorithm 2. Here, we provide the general formula for parameter updates:

$$W = W + \eta \cdot \frac{\partial L_{tr}}{\partial W} \qquad (21)$$

where η represents the learning rate.

V. EXPERIMENT

In this section, we first describe the structure of the dataset. Subsequently, we compare the performance metrics of the teacher model and the student model. We then compare the student model with other machine learning algorithms, followed by visualization and parameter analysis.

Algorithm 1 Multi-Teacher Model Training Algorithm

Hyperparameters: Enter feature dimension d;Bulk attention $nhead$=6;Number of feedforward neurons dim=1024;Random dropout $dropout$=0.2;Encode layer number $layers$=2;Learning rate η =0.001;Number of iterations T=100;Training data amount N_1;Batch size n_1=32;Optimizer=Adam.

Input: Multi-industry financial data set collection I = $\{D_1,D_2,...,D_m\}$,where $D_m = \{(X_n,Y_n)\}^{N}{}_{n}{=}_1$.

Output: Teacher model convergence parameters $W^{(t)}$.

```
Random initialization $W^{(t)} \leftarrow N(0,1)$;
Random sorting of different industries in the collection $I$;
while $t \le T$ do
    for $n = 1 : N_1 n_1$ do
        Select batch samples from data set $I\ (X_n,Y_n)$;
        for $k = 1 : layers$ do
            for $i = 1 : nhead$ do
                From the formula (1), (2), (3) calculate $Q_i$, $K_i$, $V_i$ according to $X_n$;
                From the formula (6) calculate $head_i$ according to $Q_i$, $K_i$, $V_i$;
            end for
            Calculate the multi-head attention score $M$ based on $head_i$ according to formula (7);
            Calculate the residual network and layer normalization $L$ based on $X$ and $M$ according to formula (8);
            Feed the feedforward neural network $FFN\ (L)$ based on formula (9), and apply random dropout to each fully connected layer according to formula (10);
            Calculate the residual network and layer normalization to obtain the encoder output $\bar{X}$ based on formula (8);
            Feed the output back to the input, and stack the encoder:$X = \bar{X}$;
        end for
        Apply the linear output layer to the output of the last encoder based on formula (11) to obtain the output result $Y^{pre}$;
        Calculate the cross-entropy loss for the dataset based on formula (14);
        Update the model parameters W based on formula (13);
    end for
end while
return Output the convergence parameters $W^{(t)}$ of the teacher model.
```

A. DATASET DESCRIPTION

The dataset used in this experiment is from the 9th ‘‘TipDM Cup’’ Financial Analysis Competition for Listed Companies. All listed companies in the dataset come from 19 different industries. Among them, manufacturing companies significantly outnumber companies from other industries, with 2,667 companies, while the distribution of companies in other industries is relatively even, totaling only 1,496. Due to the uneven distribution of data across different industries, we divide the entire dataset into two categories: manufacturing and other industries. We separately train student models for the manufacturing industry and other industries. These two models serve as subsystems in a distributed framework. The experiment involves training on 70% of the data, with the remaining 30% used as a validation set.

Algorithm 2 Student Model Training Algorithm

Hyperparameters: Enter feature dimension d;Bulk attention $nhead$=2;Number of feedforward neurons dim=1024;Random dropout $dropout$=0.2;Encode layer number $layers$=2;Learning rate η=0.001;Distillation temperature Tem=7;Number of iterations T=100;Training data amount N_1;Batch size n_1=32;Optimizer=Adam.

Input: Multi-industry financial data set collection $I = \{D_1,D_2,...,D_m\}$,where $D_m = \{(X_n,Y_n)\}^N{}_{n=1}$.

Output: Multi-industry student network convergence parameters $W(s)$ = n$w(1s)$,$w(2s)$,...,$w(ns)$o, where $w(ns)$

Express the network convergence parameters in a certain industry.

```
Random initialization $W^{(s)} \leftarrow N(0,1)$;
for $i = 1 : m$ do
    Select the industry dataset $D_i$ from the collection $I$;
    Sorting the sample of the industry dataset $D_i$ randomly sort;
    while $t \le T$ do
        for $n = 1 : N_1n_1$ do
            Select batch samples from data set $D_i$ $(X_n,Y_n)$;
            Calculate the output $Z_n^{(t)}$ of the teacher network based on $X_n$ and the teacher network parameters $W^{(t)}$ from algorithm 1;
            Calculate the output $Z_n^{(s)}$ of the student network based on $X_n$ and the student network parameters $w_n^{(s)}$;
            According to equations (16) and (17),distill the classification results $Z_n^{(t)}$ and $Z_n^{(s)}$ through a distillation process with distillation temperature $Tem = t$, resulting in distilled outputs $P_n^{(t)}$ and $P_n^{(s)}$;
            According to equation (15),distill the classification result $Z_n^{(s)}$ of the student network through a distillation process with a distillation temperature $Tem = 1$, obtaining the distilled output $P_n$;
            Calculate the final loss $L_n$ for dataset $D_i$ based on formulas (18),(19) and (20);
            Finally, update the parameters $w_n^{(s)}$ of the student network based on the final loss $L_i$ using formula (21);
        end for
    end while
end for
return Multi-industry student network convergence parameters $W(s)$ = n$w(1s)$,$w(2s)$,...,$wn(s)$o, where $wn(s)$ express the network convergence parameters in a certain industry.
```

TABLE 2. Summary of the analyzed data sets.

Dataset	No.of Negatives	No.of Positives	No.of Features
Other industries	6661	89	85
Manufacturing industry	11219	91	85

B. TEACHER MODEL AND STUDENT MODEL PERFORMANCE COMPARISON ANALYSIS

Our experiment was conducted on the hardware platform of 13th Gen Intel(R) Core(TM) i9-13900KF 3.00 GHz and NVIDIA GeForce RTX3060 Ti. The primary configuration environment for the experiment includes Python 3.9.1, torch 2.0.1, numpy 1.22.4, and pandas 2.1.1. All machine learning algorithms were implemented using the third-party library Scikit-Learn. The Transformer-based financial fraud detection model was constructed using the PyTorch deep learning framework.

For the final detection criteria, we utilize the following metrics:

$$precision = \frac{TP}{} \tag{22}$$

~~*TP + FP*~~

$$\frac{TP}{TP + FN} \qquad recall = (23)$$

$$f1\_score = 2 \cdot \frac{recall \cdot precision}{recall + precision} \qquad (24)$$

$$\frac{TP + TN}{TP + FP + TN + FN} \qquad accuracy = (25)$$

where *TP*,*TN*,*FP* and *FN*,represent true positives,true negatives,false positives, and false negatives, respectively.

After training the proposed model on the training set, evaluation was conducted using the test set to assess the detection performance and speed of both the teacher model and the student model. As shown in Table 3, in terms of detection performance, the student model that learned distillation had average Accura values of 98.98% and 98.83% on the other industries and manufacturing datasets, respectively, compared to only 97.54% and 97.38% for the instructor model, implying that the student model outperformed the instructor model in terms of detection accuracy. The average Recall on the dataset Other Industries and Manufacturing was 92.51% and 90.12% for the teacher model, and 95.45% and 92.70% for the student model, suggesting that the student model outperforms the teacher model at proper detection.

TABLE 3. Comparison of evaluation metrics between teacher and student models.

Dataset	Method	Accura1	Recall	Precision	F1 score
Other industries	Teacher Model	0.9754	0.9251	0.6236	0.8142
	Student Model	**0.9898**	**0.9545**	**0.8148**	**0.9287**
Manufact-uring industry	Teacher Model	0.9738	0.9012	0.5036	0.7213
	Student Model	**0.9883**	**0.9270**	**0.6774**	**0.8765**

TABLE 4. Inference time comparison between teacher and student models.

Dataset	Method	Avg.Time/μs	
		cpu	gpu
Other industries	Teacher Model	825.5	121.4
	Student Model	187.3	31.2
Manufact-uring industry	Teacher Model	1048.7	262.3
	Student Model	256.4	40.7

In terms of model inference speed, as shown in Table 4, on the dataset from other industries, the teacher model has average inference times of 825.5μ s and 121.4μ s on CPU and GPU, respectively. In comparison, the student model has average inference times of 187.3μ s and 31.2μ s, which are faster by 638.2μ s and 90.2μ s, respectively. On the manufacturing industry dataset, the teacher model has average inference times of 1048.7μ s and 262.3μ s on CPU and GPU, while the student model has average inference times of 256.4μ s and 40.7μ s. The student model is faster by 792.3μ s and 221.3μ s, respectively.From the table, it can be observed that the inference speed of the student model is generally faster than that of the teacher model. This is because the student model has fewer parameters and a simpler structure than the teacher model, leading to faster inference speed. Additionally, the inference speed on GPU is faster compared to CPU, as GPUs are better suited for matrix operations. The experimental results of comparing the performance of the teacher model and the student model show that after multi-teacher distributed knowledge distillation, the student model improves detection performance,generalizationability,andinferencespeedmore than the teacher network does.

C. COMPARISON RESULTS OF STUDENT MODEL DETECTION PERFORMANCE WITH OTHER ALGORITHMS

In order to further evaluate the performance of the student model, we compared the proposed method with advanced machine learning algorithms, including Log Reg [27], SVM linear [28], DT [29], RF [30], XGBoost [31], and Adaboost [32].

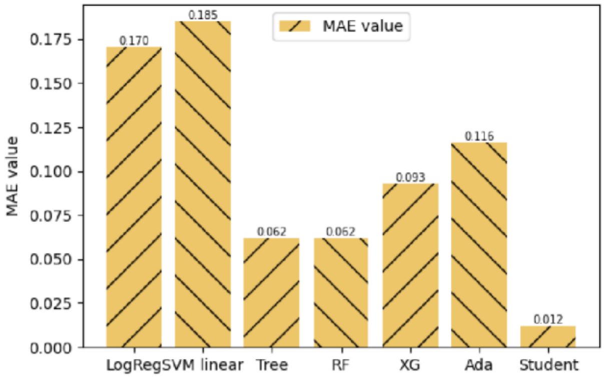


FIGURE 4. Comparative analysis of MAE values of the proposed method with other models.

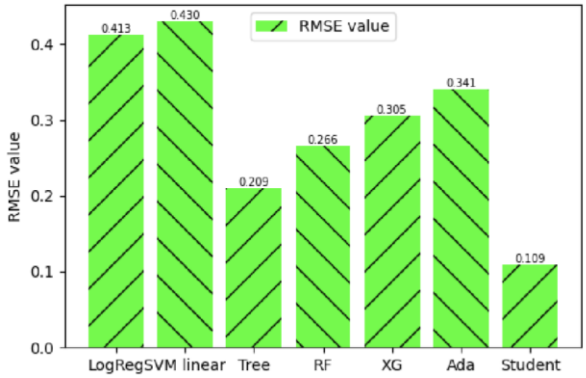


FIGURE 5. Comparative analysis of RMSE values of the proposed method with other models.

To begin, this study uses the MAE and RMSE to assess each model's error performance on the test data. Figures 4 and 5 demonstrate a comparison examination of MAE and RMSE, with the findings indicating that our suggested model has lower MAE and RMSE than the other models.

Second, MCC is used to evaluate the classification model's performance; the MCC can provide a more accurate performance assessment in unbalanced datasets. The closer the MCC metric is to 1 indicates better model classification performance. The comparison study of MCC is displayed in Figure 6, and the findings reveal that our proposed model has a higher MCC than the other models, implying that our model has better classification performance in unbalanced datasets.

The performance was assessed based on accuracy, precision, recall, and F1 score. As indicated in Table 5, our proposed method achieved the highest accuracy of 98.98% percent on other sectors and 98.83% percent on manufacturing industries. Log Reg and linear SVM achieved the lowest accuracy in other industries and manufacturing, with values of 84.01% and 81.47%, respectively. Our proposed method achieved the highest recall in other industries at 95.45%, while the Tree algorithm slightly surpassed our model in manufacturing with a recall of 93.36%. Our proposed method also achieved the highest precision, with values of 81.48% and 67.74% for other industries and manufacturing, respectively. The Tree algorithm slightly lagged behind our proposed method, with precision values of 72.41% and 66.66% for other industries and manufacturing. Furthermore, our proposed method obtained the highest F1 scores, with values of 92.87% and 87.65% for other industries and manufacturing, respectively. The F1 scores of the Tree algorithm were lower than our proposed method,

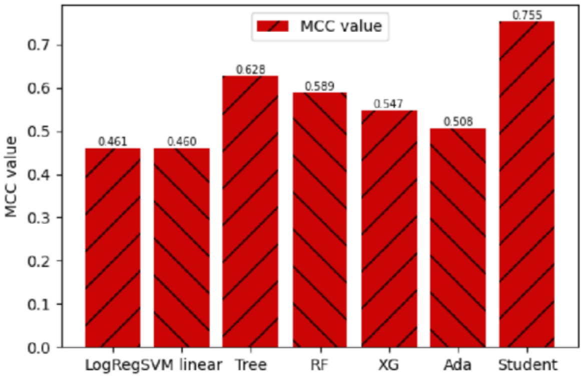


FIGURE 6. Comparative analysis of MCC values of the proposed method with other models.

TABLE 5. Comparison of evaluation metrics between student model and machine learning model.

Dataset	Method	Accura l	Recall	Precision	F1 score
Other industries	LogReg	0.8401	0.7297	0.1268	0.5605
	SVM linear	0.8430	0.7379	0.1311	0.5648
	Tree	0.9840	0.9314	0.7241	0.8920
	RF	0.8860	0.6999	0.1525	0.5861
	XG	0.9217	0.8389	0.2727	0.6790
	Ada	0.8763	0.8087	0.1827	0.6125
	Student Model	**0.9898**	**0.9545**	**0.8148**	**0.9287**
Manufact-uring industry	LogReg	0.8296	0.7119	0.0717	0.5167
	SVM linear	0.8147	0.7177	0.0688	0.5101
	Tree	0.9880	0.9336	0.6666	0.8756
	RF	0.9293	0.7628	0.1679	0.6121
	XG	0.9072	0.8722	0.1657	0.6135
	Ada	0.8836	0.8601	0.1358	0.5853
	Student Model	**0.9883**	**0.9270**	**0.6774**	**0.8765**

with values of 89.20% and 87.56% for other industries and manufacturing.

The ROC curve is a measure of the model's overall classification performance, and the area under the ROC curve is the AUC; the closer the AUC value is to one, the better the model's correct classification performance, and the closer it is to zero, the worse the surface model's correct classification performance. Figures 7 and 8 display the ROC curves of the proposed method and other machine learning algorithms on other and manufacturing industry datasets. The ROC curves of the proposed method are positioned closest to the topleft corner of the graphs, indicating superior performance of the proposed fraud detection model on both datasets. These results demonstrate the effectiveness of our proposed method.

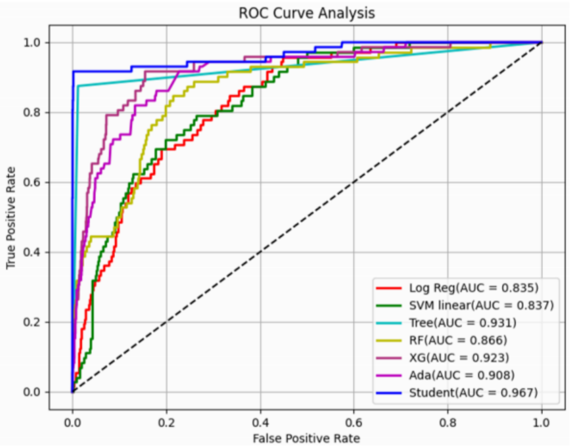


FIGURE 7. The proposed method and AUC curves compared to other machine learning algorithms on datasets from various industries.

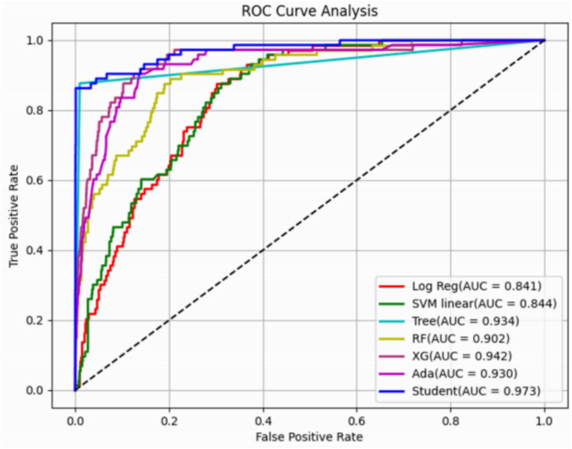


FIGURE 8. The proposed method and AUC curves for a manufacturing dataset compared to other ML algorithms.

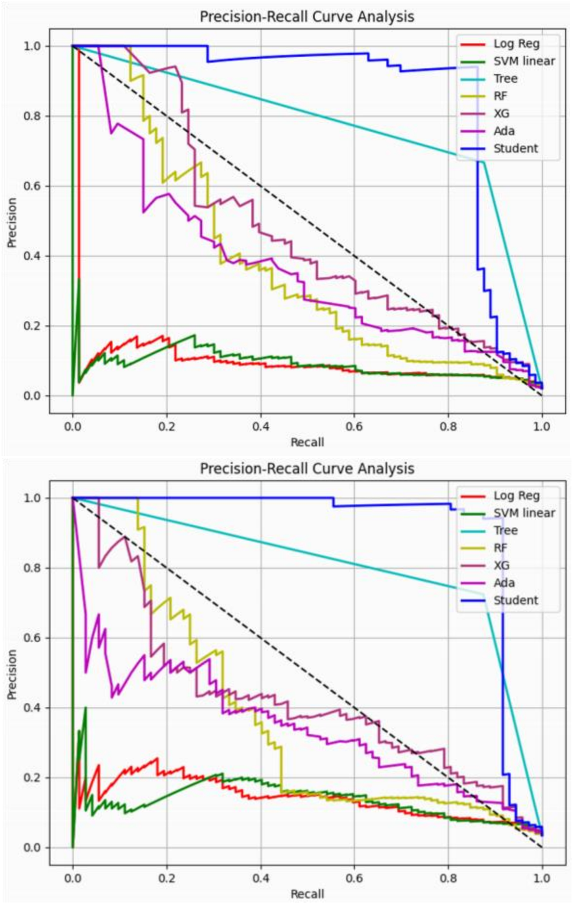


FIGURE 9. The proposed method and precision-recall curves on datasets from various industries compared to other ML algorithms.

FIGURE 10. The proposed method and precision-recall curves on a manufacturing dataset compared to other ML algorithms.

Precision and recall are important metrics for comparing classifier performance. Precision-recall (PR) curves can be plotted based on precision and recall, and the quality of a system can be judged based on these curves. The PR curve is plotted with recall on the x-axis and precision on the y-axis. Figures 9 and 10 clearly illustrate the PR curves of our proposed method and other machine learning algorithms. The PR curve of the proposed method is positioned in the upper-right corner of the graphs, indicating good performance on both datasets. Additionally, the PR curve of the proposed method is higher than the PR curves of other algorithms, suggesting that, compared to other machine learning algorithms.

VI. CONCLUSION

The detection of fraudulent financial data in listed companies is of significant importance for safeguarding the interests of shareholders and investors. This paper proposes a distributed knowledge distillation framework based on Transformer for detecting fraudulent financial data in listed companies. Experimental validation was conducted using the dataset from the 9th ‘‘TipDM Cup’’ Financial Analysis Competition for Listed Companies. The performance of the proposed method was evaluated by comparing it with other advanced machine learning algorithms, including logistic regression, linear support vector machine, decision tree, random forest, XGBoost, and Adaboost. The experimental results demonstrate that the proposed method outperforms other machine learning algorithms, achieving the highest performance in terms of AUC, accuracy, precision, recall, and F1 score.

REFERENCES

[1] C. Defang and L. Baichi, ‘‘SVM model for financial fraud detection,’’ *Northeastern Univ., Natural Sci.*, vol. 40, pp. 295–299, Feb. 2019.

[2] T. Shahana, V. Lavanya, and A. R. Bhat, ‘‘State of the art in financial statement fraud detection: A systematic review,’’ *Technological Forecasting Social Change*, vol. 192, Jul. 2023, Art. no. 122527.

[3] W. Xiuguo and D. Shengyong, ‘‘An analysis on financial statement fraud detection for Chinese listed companies using deep learning,’’ *IEEE Access*, vol. 10, pp. 22516–22532, 2022.

[4] M. N. Ashtiani and B. Raahemi, ‘‘Intelligent fraud detection in financial statements using machine learning and data mining: A systematic literature review,’’ *IEEE Access*, vol. 10, pp. 72504–72525, 2022.

[5] M. El-Bannany, A. H. Dehghan, and A. M. Khedr, ‘‘Prediction of financial statement fraud using machine learning techniques in UAE,’’ in *Proc. 18th Int. Multi-Conf. Syst., Signals Devices (SSD)*, Mar. 2021, pp. 649–654.

[6] R. Cao, G. Liu, Y. Xie, and C. Jiang, ‘‘Two-level attention model of representation learning for fraud detection,’’ *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 6, pp. 1291–1301, Dec. 2021.

[7] A. Singh, A. Singh, A. Aggarwal, and A. Chauhan, ‘‘Design and implementation of different machine learning algorithms for credit card fraud detection,’’ in *Proc. Int. Conf. Electr., Comput., Commun. Mechatronics Eng. (ICECCME)*, Nov. 2022, pp. 1–6.

[8] C. Liu, Y.-C. Chan, S. H. Alam, and H. Fu, ‘‘Financial fraud detection model: Based on random forest,’’ in *Econometrics: Econometric Model Construction*, 2015.

[9] H. Shivraman, U. Garg, A. Panth, A. Kandpal, and A. Gupta, ‘‘A model frame work to segregate clusters through K-means method,’’ in *Proc. 2nd Int. Conf. Comput. Sci., Eng. Appl. (ICCSEA)*, Sep. 2022, pp. 1–6.

[10] N. Sharma and V. Ranjan, ‘‘Credit card fraud detection: A hybrid of PSO and K-means clustering unsupervised approach,’’ in *Proc. 13th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Jan. 2023, pp. 445–450.

[11] G. Rushin, C. Stancil, M. Sun, S. Adams, and P. Beling, ‘‘Horse race analysis in credit card fraud—Deep learning, logistic regression, and gradient boosted tree,’’ in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Apr. 2017, pp. 117–121.

[12] J. Jurgovsky, M. Granitzer, K. Ziegler, S. Calabretto, P.-E. Portier, L. He-Guelton, and O. Caelen, ‘‘Sequence classification for credit-card fraud detection,’’ *Exp. Syst. Appl.*, vol. 100, pp. 234–245, Jun. 2018.

[13] H. Zhou, G. Sun, S. Fu, L. Wang, J. Hu, and Y. Gao, ‘‘Internet financial fraud detection based on a distributed big data approach with node2vec,’’ *IEEE Access*, vol. 9, pp. 43378–43386, 2021.

[14] R. Li, Z. Liu, Y. Ma, D. Yang, and S. Sun, ‘‘Internet financial fraud detection based on graph learning,’’ *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 3, pp. 1394–1401, 2023.

[15] A. Singh, A. Gupta, H. Wadhwa, S. Asthana, and A. Arora, ‘‘Temporal debiasing using adversarial loss based GNN architecture for crypto fraud detection,’’ in *Proc. 20th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2021, pp. 391–396.

[16] X. Liu, K. Yan, L. Burak Kara, and Z. Nie, ''CCFD-net: A novel deep learning model for credit card fraud detection,'' in *Proc. IEEE 22nd Int. Conf. Inf. Reuse Integr. Data Sci. (IRI)*, Aug. 2021, pp. 9–16.

[17] S. You, C. Xu, C. Xu, and D. Tao, ''Learning from multiple teacher networks,'' in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1285–1294.

[18] W. Shi, G. Ren, Y. Chen, and S. Yan, ''ProxylessKD: Direct knowledge distillation with inherited classifier for face recognition,'' 2020, *arXiv:2011.00265*.

[19] M. Shin, ''Semi-supervised learning with a teacher–student network for generalized attribute prediction,'' in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 509–525.

[20] H. Zhang, D. Chen, and C. Wang, ''Adaptive multi-teacher knowledge distillation with meta-learning,'' in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2023, pp. 1943–1948.

[21] A. Amirkhani, A. Khosravian, M. Masih-Tehrani, and H. Kashiani, ''Robust semantic segmentation with multi-teacher knowledge distillation,'' *IEEE Access*, vol. 9, pp. 119049–119066, 2021.

[22] B. An and Y. Suh, ''Identifying financial statement fraud with decision rules obtained from modified random forest,'' *Data Technol. Appl.*, vol. 54, no. 2, pp. 235–255, May 2020.

[23] P. Craja, A. Kim, and S. Lessmann, ''Deep learning for detecting financial statement fraud,'' *Decis. Support Syst.*, vol. 139, Dec. 2020, Art. no. 113421.

[24] J. Geng and B. Zhang, ''Credit card fraud detection using adversarial learning,'' in *Proc. Int. Conf. Image Process., Comput. Vis. Mach. Learn. (ICICML)*, 2023, pp. 891–894.

[25] E. Orhan, ''Skip connections as effective symmetry-breaking,'' 2017, *arXiv:1701.09175*.

[26] H. Hong and H. Kim, ''Feature distribution-based knowledge distillation for deep neural networks,'' in *Proc. 19th Int. SoC Design Conf. (ISOCC)*, Oct. 2022, pp. 75–76.

[27] D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, ''Credit card fraud detection–machine learning methods,'' in *Proc. 18th Int. Symp. Infoteh-Jahorina (INFOTEH)*, Mar. 2019, pp. 1–5.

[28] T. Priyaradhikadevi, S. Vanakovarayan, E. Praveena, V. Mathavan, S. Prasanna, and K. Madhan, ''Credit card fraud detection using machine learning based on support vector machine,'' in *Proc. 8th Int. Conf. Sci. Technol. Eng. Math. (ICONSTEM)*, Apr. 2023, pp. 1–6.

[29] C.-C. Lin, A.-A. Chiu, S. Y. Huang, and D. C. Yen, ''Detecting the financial statement fraud: The analysis of the differences between data mining techniques and experts' judgments,'' *Knowl.-Based Syst.*, vol. 89, pp. 459–470, Nov. 2015.

[30] V. Arora, R. S. Leekha, K. Lee, and A. Kataria, ''Facilitating user authorization from imbalanced data logs of credit cards using artificial intelligence,'' *Mobile Inf. Syst.*, vol. 2020, pp. 1–13, Oct. 2020.

[31] L. Torlay, M. Perrone-Bertolotti, E. Thomas, and M. Baciu, ''Machine learning–XGBoost analysis of language networks to classify patients with epilepsy,'' *Brain Informat.*, vol. 4, no. 3, pp. 159–169, Sep. 2017.

[32] P. Yu and X. Liu, ''Construction and application of bid fraud prediction model based on AdaBoost algorithm,'' in *Proc. 2nd Int. Conf. Electron. Inf. Eng. Comput. Technol. (EIECT)*, Oct. 2022, pp. 292–295.

[33] T. Zhang and S. Gao, ''Graph attention network fraud detection based on feature aggregation,'' in *Proc. 4th Int. Conf. Intell. Inf. Process. (IIP)*, Oct. 2022, pp. 272–275.

[34] A.Vaswani,N.M.Shazeer,N.Parmar,J.Uszkoreit,L.Jones ,A.N.Gomez, L. Kaiser, and I. Polosukhin, ''Attention is all you need,'' in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 1–11.