International Journal of Innovative Research in Engineering and Management (IJIREM)
ISSN (Online): 2350-0557, Volume-11, Issue-3, June 2024
https://doi.org/10.55524/ijirem.2024.11.3.12
Article ID IJIRD-1336, Pages 77-81
www.ijirem.org

Investigating Financial Risk Behavior Prediction Using Deep Learning and Big Data

Ke Xu[1], You Wu[2], Zichao Li[3], Rong Zhang[4], and Zixin Feng[5]

[1] Columbia University, New York, USA
[2] College of William & Mary, Williamsburg, USA
[3] Canoakbit Alliance Inc, Oakville, Canada
[4] University of California, Davis, Davis, USA
[5] Georgia Institute of Technology, Atlanta, USA

Correspondence should be addressed to Ke Xu; Katiexu.kx@gmail.com

Received: 19 May 2024 Revised: 3 June 2024 Accepted: 18 June 2024

ABSTRACT- This paper introduces a sophisticated deep learning model designed to predict high-risk behaviors in financial traders by analyzing vast amounts of transaction data. The model begins with an unsupervised pre-training phase, learning distributed representations that capture complex data relationships autonomously. It then utilizes a deep neural network, enhanced through supervised learning, to classify and predict traders' risk levels effectively. We specifically focus on financial spread trading related to Contracts For Difference (CFD), identifying potential misuse of insider information and assessing the risks it poses to market makers. By distinguishing between high-risk (A-book) and lower-risk (B-book) clients, the model supports strategic hedging decisions, crucial for market stability. Our extensive evaluations confirm the model's robustness and accuracy, highlighting its significant potential for practical implementation in dynamic and speculative financial markets where past trading performance may not predict future outcomes. This advancement not only refines risk management strategies but also contributes broadly to the domain of financial technology.

KEYWORDS- Financial Risk Prediction; Big Data; Deep Learning; Unsupervised Training

I. INTRODUCTION

Deep neural networks exhibit exceptional learning capabilities, and advancements in this technology have profoundly influenced numerous fields[1-2], achieving notable success across an expanding array of domains, such as disease prediction[3], image segmentation[4-5], natural language processing[6]. Financial risk management stands out as a key application area. Traditionally, most financial institutions employ conventional machine learning algorithms to forecast financial market trends, evaluate borrowers' repayment capacities, and make credit approval decisions, among other functions [7]. Leveraging the robust learning capabilities of deep learning models, this paper explores the application of deep neural networks in predicting financial risk behaviors, with the objective of identifying high-risk customers.

Financial spread trading is a versatile derivative trading method and a significant financial instrument. It typically involves paired stock trading or futures market price differentials. This paper examines the form of spread trading associated with Contracts For Difference (CFD) [8]. In financial markets, retail investors and market makers enter contracts tied to specific financial instruments, exchanging the difference between the closing and opening prices of the financial instrument at the contract's end. Some traders exploit insider information to trade, reaping substantial profits during market upswings. This practice disadvantages market makers, whose primary income comes from the bid-ask spread. In liquid markets, the spread in spread trading markets exceeds that in underlying markets. Conversely, for less liquid financial instruments, the spread is narrower than in underlying markets. Market makers engaging in hedging trades may lose potential profits from the spread and incur transaction costs.

Therefore, creating a predictive classification model to distinguish between A-book clients (clients posing the highest risk to market makers) and B-book clients (lower-risk clients) is essential. Market makers hedge A-book clients' positions to prevent losses while assuming B-book clients' positions to enhance profits. The decision to hedge is framed as a classification issue, utilizing deep neural networks to predict high-risk (A-book) traders.

Defining A-book clients is subjective and hinges on the market maker's strategy. Clients achieving a 5% return within the first 20 trades are categorized as high-risk traders. This classification is dynamic, based on the client's initial 20 trades, meaning a single trade can alter a client's status[9]. While hedging client i's j+k trade, the market maker might bear the risk of client i's j-th trade. In speculative markets, past trading performance can mislead predictions of future profitability, as previous results don't necessarily indicate true ability. Thus, the client classification model aims to generate reliable hedging decisions by considering all relevant data features. Deep neural networks analyze past trading data to understand traders' potential risks. For these networks, extracting high-level distributed representations of target concepts from trading data is crucial, as these representations capture the underlying factors influencing trading behavior changes.

II. PREDICTIVE CLASSIFICATION USING DEEP NEURAL NETWORKS

Deep learning focuses on learning hierarchical representations from data, where higher levels capture more abstract concepts. Compared to traditional machine learning methods, deep architectures with multiple layers offer superior learning capabilities that have been utilized in many fields, such as medical diagnosis[10-13], computer vision[14-17], LSTM[18-20]. Theoretical studies suggest that a learning machine with depth *k+1* requires exponentially more computational units to represent a function than Deep learning models with depth *k* outperform simple regression models, and classifiers show better performance than individual learners. After training on a dataset, machine learning can execute classification tasks. However, training data may sometimes lack representativeness, which can affect the classification performance. Distributed representations help mitigate the issue of unrepresentative training data.

Consider the example of trader classification. Traders demonstrate varying trading styles, such as employing different strategies and following different stop-loss rules[21]. Assume traders are categorized into five distinct groups, each group sharing a specific trading style. With non-distributed representations, five distinct features are required to represent each cluster uniquely, whereas distributed representations need only three features to model the clustering effectively.

A. *Unsupervised pre-training*

The aim of pre-training is to identify distributed representations of data, which can account for variations in the data and highlight those variations critical for classification. Through a series of nonlinear transformations, the pre-training process develops feature detection layers that prevent erroneous information from spreading through multi-layer networks, thereby mitigating the vanishing gradient problem. Two classic methods for pre-training are Deep Belief Networks [22] and Stacked Denoising Autoencoders[23]. These methods both minimize the log-likelihood of the generative model, typically resulting in similar performance. In this context, a Stacked Denoising Autoencoder is employed for pre-training. The Denoising Autoencoder learns distributed representations from the input samples. Assuming there are N samples, each with p features, and the input sample is x, the learning process of the Denoising Autoencoder involves five steps:

Step 1: The Denoising Autoencoder initially corrupts the input sample x. By sampling from a binomial distribution ($n=N$, $p=p_q$), it randomly corrupts a subset of the samples and introduces noise.

Step 2: The Denoising Autoencoder maps the corrupted input x to a higher-level representation y. This mapping is performed through a hidden neural network layer. Given the weight matrix W, bias b, and the encoding function $h(\cdot)$, y can be represented as:

$$y=h(Wx+b) \tag{1}$$

Step 3: The decoder reconstructs y into z, which has the same structure as the input x. z can be viewed as the prediction of x. The reconstruction process of z is a denoising process that reconstructs the input from the corrupted sample x. Similar to the encoder, given the weight matrix $\widehat{W}y$, bias $\tilde{b}$, and the decoding function $g(\cdot)$, z can be expressed as:

$$z = g(\widehat{W}\,y + \tilde{b}) \tag{2}$$

Step 4: The goal of optimizing the parameters of the Denoising Autoencoder is to minimize the reconstruction error L_{xz}.At this point, each hidden unit in y represents one of the principal components of the data. The choice of the cost function depends on the assumed distribution of the input z. The cross-entropy loss function is employed to measure the reconstruction error. Furthermore, an $L2$ regularization penalty function is utilized to represent weight decay. The regularization parameter λ delineates the trade-off between reconstruction error and model complexity. The final cost function is formulated as follows:

$$L(x,z) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{p}[x_{ik}\log z_{ik} + (1-x_{ik})\log(1-z_{ik})] + \lambda|W|_2 \tag{3}$$

Step 5: Stack multiple Denoising Autoencoders to create a deep architecture. Each layer of the Denoising Autoencoder uses the encoded output from the previous layer as its input. Each layer of the Denoising Autoencoder is trained locally to determine its optimal weights.

B. *Supervised prediction*

To better utilize the network for prediction, supervised fine-tuning is necessary. For this purpose, a softmax regression function is added on top of the stacked Denoising Autoencoders. The distributed representation of the original input is used as features, and a binary indicator variable is used as the target. This binary variable indicates whether the hedging transaction should continue. Given the parameter weights W and bias b, the probability of transaction x belonging to category i is:

$$P(Y = i \mid x, W, b) = \mathrm{softmax_i}(Wx+b) = \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}} \tag{4}$$

The negative log-likelihood function is employed as the loss function during supervised fine-tuning,Assuming y is the true classification of input x, the loss function is formulated as follows:

$$L(W,b,x) = -\sum_{i=1}^{N} log\left(P(Y = y^i \mid x^i, W, b)\right) \tag{5}$$

To mitigate the overfitting issue in deep neural network models, a dropout layer is added after each hidden layer of the deep neural network. During training, dropout removes hidden layer neurons and their corresponding connection weights with a certain probability. Since the removal is random, each mini-batch effectively trains a different neural network. The probability of removing hidden neurons follows a Bernoulli distribution with a specified dropout rate. During prediction, the deep neural network considers all hidden layer neurons and scales the connection weights of each hidden neuron by the expected value of the Bernoulli distribution. Dropout simulates the averaging process of geometric models and considers every possible combination of hidden neurons to enhance prediction accuracy. Dropout prevents hidden neurons from becoming too reliant on each other, thereby helping to prevent overfitting.

This deep neural network employs stacked Denoising Autoencoders for unsupervised pre-training to adjust weights layer by layer, followed by supervised fine-tuning of the entire network, with a dropout layer added after each hidden layer. During the pre-training phase, the parameters to be determined are the weight matrices and biases in each Denoising Autoencoder (encoder and decoder). In the supervised fine-tuning phase, the parameters include the weight matrices and the biases in the encoders and the softmax regression of the stacked Denoising Autoencoders. Stochastic gradient descent with momentum and a decaying learning rate is used for training the deep neural network. The Python library Theano is utilized. The GPU used is an NVIDIA Tesla K20.

III. EXPERIMENTAL EVALUATION

The dataset utilized for the experiment spans a decade of actual transaction data, encompassing more than 30 million transactions conducted by 25,000 traders from Global Insider Trading Data. To address data formats problem, Linked Data methodology was employed[24]. Supervised learning necessitates the use of a labeled dataset $D = \{y_i, x_i\}_{i=,...,n}$, Here, x represents the feature vector of transaction i, and y_i denotes the target variable. Utilize information from previous transactions to determine whether to hedge the current transaction. The target variable y_i takes the value i to indicate the adoption of a hedging strategy, while a value of -1 means that a hedging strategy is not adopted. When the return is greater than or equal to 5%, y_i is set to 1; otherwise, y_i is set to -1. The calculation method for return is shown below:

$$return_i = \frac{\sum_{20\le j\le 100} PL_{ij}}{\sum_{20\le j\le 100} Margin_{ij}} \quad (6)$$

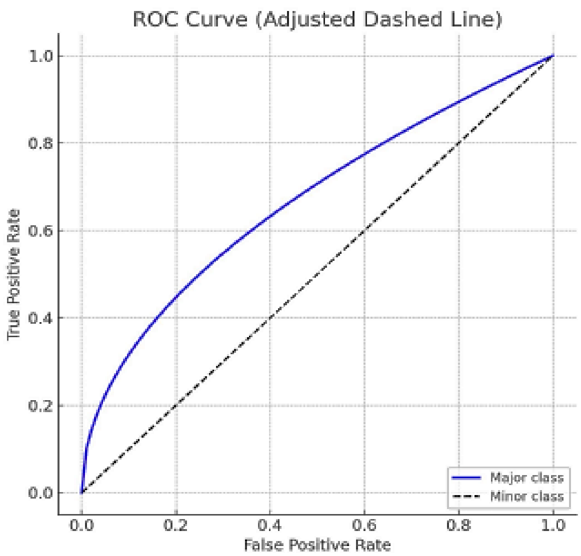
In this context, PL_{ij} represents the profit and loss of transaction j, while $Margin_{ij}$ refers to the capital required by the market maker to execute an order. To identify transaction j, the status of trader i at the time of the transaction issuance is determined: if trader i attains a return exceeding 5% on the subsequent transaction following the *j-th* transaction, trader i is classified as an A-book client. Since the future profit for trader i during transaction j is indeterminate, the predictive model utilizes historical transaction data to forecast y_{ij}. The feature vector x_{ij} comprises both the client information for transaction j and the behavioral data from the 20 transactions preceding transaction j.

The features are categorized into five groups. The first group includes trader information such as age, nationality, employment status, and salary. The second group captures the trader's historical performance, utilizing the mean and standard deviation to assess features from the preceding 20 trades. Besides profitability, this group also computes related performance metrics like average win rate and average profit points. The third group outlines the trader's preferences for markets and channels, summarizing the trader's entire history and the most favored market segments in the last 20 trades. Channel features separately track the number of trades opened and closed via the web front end and mobile applications. The fourth group addresses the disposition effect, which reflects the tendency of investors to quickly sell winning trades while holding onto losing ones. This feature set records the average amount and duration of positions won and lost by each trader, calculating the respective ratios. The fifth group signals the consistency of the trader's strategy, detailing the standard deviation of trade sizes and trading frequency, and their variations. This group also considers the trend of trading during and outside regular hours, indicating the trader's professionalism. The proposed model is evaluated against artificial neural networks, adaptive boosting, and support vector machine models. Table 1 presents a comparison of the four classification models across multiple evaluation criteria. The results in Table 1 are averaged from a 10-fold cross-validation. According to the performance metrics in Table 1, the proposed deep neural network consistently outperforms other machine learning models.

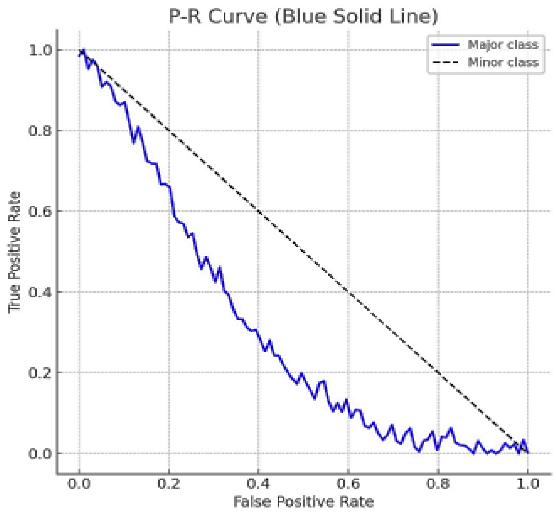
Table 1: Performance Comparison of Classification Models

Model Type	Profit and Loss (USD)	Misclassification Cost (USD)	Sensitivity Metric	Accuracy
Proposed Model	1079.23	4363.35	0.640	0.990
Artificial Neural Networks	984.23	8258.22	0.309	0.981
AdaBoost (Adaptive Boosting)	913.00	8454.37	0.293	0.970
Support Vector Machines	616.30	7445.05	0.380	0.972

To demonstrate the value of deep structures, we compared our deep neural network model with a logistic regression model that lacks deep hidden layers (denoted as the simple logistic regression model). Figure 1(a) presents the ROC (Receiver Operating Characteristic) [25] curves figure 1(b)and Precision-Recall (P-R) [26]curves for both the deep neural network and the simple logistic regression model. The ROC curve indicates that our deep neural network model has a larger AUC (Area Under the Curve) [27], signifying higher accuracy. Furthermore, the P-R curve results confirm that the deep architecture enhances the network's classification capabilities.



(a)



(b)

Figure 1: ROC and P-R curves

Subsequently, the study evaluates the performance of the unsupervised pretraining phase, aiming to ascertain whether the deep neural network acquires distributed representations capable of distinguishing between A-book and B-book customers in unlabelled data. Figure 2 illustrates the activation values of neurons in the first dA layer. The findings reveal that transactions from B-book customers generally produce activation values below 0.4, whereas those from A-book customers typically generate activation values of 0.4 or higher. This disparity in activation values between customer types indicates that the first dA layer can effectively differentiate A-book from B-book customer transactions even in the absence of labelled data.

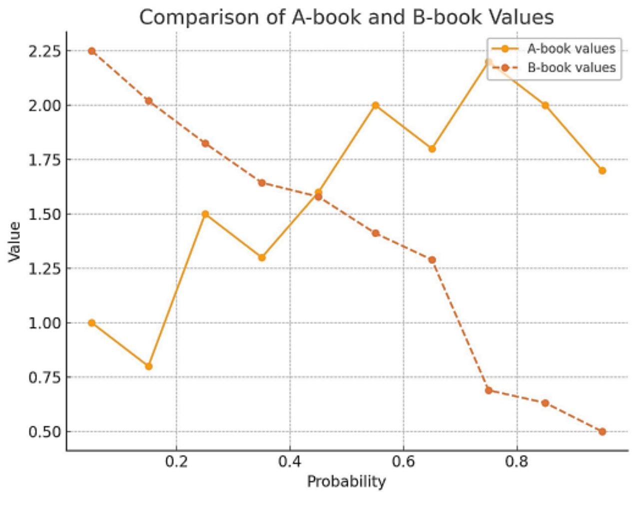


Figure 2: Activation values of neurons in the first dA layer

IV. CONCLUSION

This study delves into the utilization of advanced deep learning techniques for detecting and predicting high-risk financial behaviors, particularly in the context of financial spread trading. Our comprehensive predictive model, developed around deep neural networks, was thoroughly tested using a substantial real-world dataset, highlighting its capability to achieve high accuracy in identifying high-risk traders, especially those engaging in spread trading with Contracts For Difference (CFD). Despite the relatively small proportion of high-risk customers (i.e., A-book customers), their potential impact on market stability is substantial, making their precise identification crucial. The model not only discriminates between A-book and B-book customers effectively but also adapts to the dynamic nature of trader behaviors, capturing shifts in risk profiles that traditional models might overlook. This adaptability is critical in speculative markets where past performance does not reliably predict future risk. The integration of our deep learning model into financial risk management processes promises not only enhanced accuracy in risk assessment but also supports a more nuanced and strategic approach to hedging, thereby bolstering the resilience and stability of financial markets. These findings advocate for a broader adoption of deep learning strategies in financial risk management, potentially transforming the landscape of financial analysis and decision-making.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest between them and with any third party

REFERENCES

[1] Y. Yang, Z. Chen, Y. Yan, M. Li, and T. Gegen, "A new method of image recognition based on deep learning generates adversarial networks and integrates traditional algorithms," Journal of Computing and Electronic Information Management, vol. 13, no. 1, pp. 57-61, 2024.

[2] Y. Yan, S. He, Z. Yu, J. Yuan, Z. Liu, and Y. Chen, "Investigation of Customized Medical Decision Algorithms Utilizing Graph Neural Networks," arXiv preprint arXiv:2405.17460, 2024.

[3] X. Yan, W. Wang, M. Xiao, Y. Li, and M. Gao, "Survival Prediction Across Diverse Cancer Types Using Neural Networks," arXiv preprint arXiv:2404.08713, 2024.

[4] H. Zhang, S. Diao, Y. Yang, J. Zhong, and Y. Yan, "Multi-scale image recognition strategy based on convolutional neural network," Journal of Computing and Electronic Information Management, vol. 12, no. 3, pp. 107-113, 2024.

[5] Z. Gao, Q. Wang, T. Mei, X. Cheng, Y. Zi, and H. Yang, "An Enhanced Encoder-Decoder Network Architecture for Reducing Information Loss in Image Semantic Segmentation," arXiv preprint arXiv:2406.01605, 2024.

[6] T. Mei, Y. Zi, X. Cheng, Z. Gao, Q. Wang, and H. Yang, "Efficiency optimization of large-scale language models based on deep learning in natural language processing tasks," arXiv preprint arXiv:2405.11704, 2024.

[7] I. Lee and Y. J. Shin, "Machine learning for enterprises: Applications, algorithm selection, and challenges," Business Horizons, vol. 63, no. 2, pp. 157-170, 2020.

[8] T. Kristiansen, "Pricing of Contracts for Difference in the Nordic market," Energy Policy, vol. 32, no. 9, pp. 1075-1085, 2004.

[9] A. Kim, Y. Yang, S. Lessmann, T. Ma, M. C. Sung, and J. E. Johnson, "Can deep learning predict risky retail investors? A case study in financial risk behavior forecasting," European Journal of Operational Research, vol. 283, no. 1, pp. 217-234, 2020.

[10] J. Yuan, L. Wu, Y. Gong, Z. Yu, Z. Liu, and S. He, "Research on Intelligent Aided Diagnosis System of Medical Image Based on Computer Deep Learning," arXiv preprint arXiv:2404.18419, 2024.

[11] S. Lu, Z. Liu, T. Liu, and W. Zhou, "Scaling-up medical vision-and-language representation learning with federated

learning," Engineering Applications of Artificial Intelligence, vol. 126, p. 107037, 2023.

[12] Z. Yao, F. Lin, S. Chai, W. He, L. Dai, and X. Fei, "Integrating Medical Imaging and Clinical Reports Using Multimodal Deep Learning for Advanced Disease Analysis," arXiv preprint arXiv:2405.17459, 2024.

[13] L. Xiao, M. Li, Y. Feng, M. Wang, Z. Zhu, and Z. Chen, "Exploration of Attention Mechanism-Enhanced Deep Learning Models in the Mining of Medical Textual Data," arXiv preprint arXiv:2406.00016, 2024.

[14] Z. Liu and J. Song, "Comparison of Tree-based Feature Selection Algorithms on Biological Omics Dataset," in Proceedings of the 5th International Conference on Advances in Artificial Intelligence, pp. 165-169, November 2021.

[15] S. Chai, X. Fei, Y. Wang, L. Dai, and M. Sui, "Deep Learning-Based Lung Medical Image Recognition," International Journal of Innovative Research in Computer Science & Technology, vol. 12, no. 3, pp. 100-105, 2024.

[16] Z. Gao, Q. Wang, T. Mei, X. Cheng, Y. Zi, and H. Yang, "An Enhanced Encoder-Decoder Network Architecture for Reducing Information Loss in Image Semantic Segmentation," arXiv preprint arXiv:2406.01605, 2024.

[17] J. Yao, T. Wu, and X. Zhang, "Improving depth gradient continuity in transformers: A comparative study on monocular depth estimation with cnn," arXiv preprint arXiv:2308.08333, 2023.

[18] W. Dai, J. Tao, X. Yan, Z. Feng, and J. Chen, "Addressing Unintended Bias in Toxicity Detection: An LSTM and Attention-Based Approach," in 2023 5th International Conference on Artificial Intelligence and Computer Applications (ICAICA), pp. 375-379, November 2023, IEEE.

[19] Q. Zhan, Y. Ma, E. Gao, D. Sun, and H. Yang, "Innovations in Time Related Expression Recognition Using LSTM Networks," International Journal of Innovative Research in Computer Science & Technology, vol. 12, no. 3, pp. 120-125, 2024.

[20] S. Wang, Z. Liu, and B. Peng, "A Self-training Framework for Automated Medical Report Generation," in Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 16443-16449, December 2023.

[21] A. W. Lo and A. Remorov, "Stop-loss strategies with serial correlation, regime switching, and transaction costs," Journal of Financial Markets, vol. 34, pp. 1-15, 2017.

[22] R. Salakhutdinov and I. Murray, "On the quantitative analysis of deep belief networks," in Proceedings of the 25th international conference on Machine learning, pp. 872-879, July 2008.

[23] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," Journal of Machine Learning Research, vol. 11, no. 12, 2010.

[24] Y. Li, X. Yan, M. Xiao, W. Wang, and F. Zhang, "Investigation of Creating Accessibility Linked Data Based on Publicly Available Accessibility Datasets," in Proceedings of the 2023 13th International Conference on Communication and Network Security, pp. 77–81, 2024, Association for Computing Machinery.

[25] M. S. Pepe, "Receiver operating characteristic methodology," Journal of the American Statistical Association, vol. 95, no. 449, pp. 308-311, 2000.

[26] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in Proceedings of the 23rd international conference on Machine learning, pp. 233-240, June 2006.

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