

An Extreme Learning Machine Stock Price Prediction Algorithm Based on the Optimisation of the Crown Porcupine Optimisation Algorithm with an Adaptive Bandwidth Kernel Function Density Estimation Algorithm

[Keke Yu](#)^{*}, Qiaozhi Bao, Haosen Xu, Guanghe Cao, Siwei Xia

Posted Date: 15 August 2024

doi: 10.20944/preprints202408.1116.v1

Keywords: Extreme learning machine; Crown porcupine optimisation algorithm; Adaptive bandwidth kernel function density estimation algorithm.



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

An Extreme Learning Machine Stock Price Prediction Algorithm Based on the Optimisation of the Crown Porcupine Optimisation Algorithm with an Adaptive Bandwidth Kernel Function Density Estimation Algorithm

Keke Yu ^{1,*}, Qiaozhi Bao ^{1,2}, Haosen Xu ³, Guanghe Cao ⁴ and Siwei Xia ⁵

¹ University of California, Santa Barbara, CA, US

² Statistics, North Carolina State University, NC, USA

³ Electrical Engineering and Computer Science, University of California, Berkeley, CA, USA

⁴ Computer Science, University of Southern California, CA, USA

⁵ Electrical and Computer Engineering, New York University, NY, USA

* Correspondence: kekeyu01@gmail.com

Abstract: This paper aims to explore the potential application of Extreme Learning Machines (ELM) in stock price prediction and optimise the ELM model by combining the Crown Porcupine Optimisation Algorithm and the Adaptive Bandwidth Kernel Function Density Estimation Algorithm. Specifically, the model's root mean square error (RMSE) on the test set is 0.018, which shows its good prediction accuracy. Meanwhile, the correlation coefficient R reaches 0.968, proving the model's effectiveness in predicting stock prices. In addition, the coefficient of determination R^2 of the training set and test set are 0.987 and 0.954, respectively, which are not much different, indicating that the model has good generalisation ability and is not prone to overfitting. This is crucial for practical application, as it implies that the model fits the existing data well and makes accurate predictions for future data. In terms of other evaluation metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE), the performance is excellent in both the training and test sets. Together, these results indicate that after optimising the ELM by introducing the crown porcupine optimisation algorithm and the adaptive bandwidth kernel function density estimation algorithm, the model can effectively capture the changing stock price pattern, thus achieving efficient and accurate stock price prediction. In summary, this study not only demonstrates the great potential of ELM in stock price prediction but also highlights the importance of improving the model performance through advanced algorithm optimisation, which provides a new way of thinking and methodology for financial market analysis. This strongly supports investors' decision-making and lays the foundation for related research.

Keywords: extreme learning machine; crown porcupine optimisation algorithm; adaptive bandwidth kernel function density estimation algorithm

I. Introduction

The stock market, as an important part of the modern economy, has attracted the attention of a large number of investors and researchers. The fluctuation of stock prices not only affects investors' returns, but also reflects the market's expectation of a company's future development potential. Therefore, accurate prediction of stock prices has become an important topic of research in many fields such as finance, economics, and computer science [1]. With the development of information technology, especially the rise of big data and machine learning technologies, traditional stock analysis methods are gradually replaced by more complex and efficient algorithms [2].

Historically, stock price prediction mainly relied on fundamental and technical analyses. Fundamental analysis focuses on factors such as a company's financial condition and industry outlook, while technical analysis focuses on historical price and volume data. However, these methods often fail to adequately capture the non-linear characteristics of the market and its complex dynamics, so researchers have begun to explore methods based on time series analysis to improve the accuracy of forecasts. Time series algorithms are an important tool for dealing with data that changes over time, and the core idea is to use past data to predict future trends. In stock price forecasting, time series models provide a basis for investment decisions by capturing patterns and regularities in historical price movements.

Common time series algorithms include the Autoregressive Moving Average Model (ARMA), the Seasonal Autoregressive Integrated Sliding Average Model (SARIMA), and the Long Short-Term Memory Network (LSTM). The ARMA model is suitable for smooth time series and forecasts because it establishes a linear relationship between current and past values. At the same time, SARIMA considers seasonal factors on this basis, making the model more flexible and applicable [3].

Deep learning algorithms such as LSTM have received widespread attention in recent years for their powerful nonlinear modelling capabilities. LSTM can effectively capture long-term dependencies and is well suited to handle data with complex patterns and trends, such as stock prices. In addition, LSTM can be combined with other features (e.g., trading volume, macroeconomic indicators, etc.) to enhance the prediction effect further. In the research of stock price prediction, although traditional time series algorithms such as ARMA [4], SARIMA [5] and deep learning models such as LSTM show good results, they often face problems such as high computational complexity and long training time when dealing with large-scale data. In order to solve these challenges, Extreme Learning Machine (ELM) as an emerging machine learning method has gradually attracted the attention of researchers.

Extreme Learning Machine is a kind of Single-hidden Layer Feedforward Neural Network (SLFN), whose main feature is its fast learning and efficient generalisation ability. Unlike traditional neural networks that need to be trained iteratively by back-propagation algorithms, ELM only needs to randomly initialise the weights of the hidden layer nodes during the training process, and then directly calculate the output layer weights by the least squares method. This process significantly reduces the training time and makes ELM perform well when dealing with large-scale datasets [6].

In stock price prediction, the Extreme Learning Machine is able to effectively capture nonlinear features and has strong generalisation ability. Since data in the financial market is usually complex and diverse, ELM can extract valuable information by reasonably selecting and pre-processing the input data. For example, historical prices, trading volumes, technical indicators, and macroeconomic factors can be used as input features to be integrated and analysed through the ELM model. In order to take advantage of the great potential of ELM in stock price prediction, this paper optimises the ELM based on the crown porcupine optimisation algorithm and adaptive bandwidth kernel function density estimation algorithm to predict the stock price, which is worth noting that it is a brand-new approach in stock price prediction.

II. Data from Data Analysis

The dataset selected for this paper is an open-source dataset that records opening, high and closing prices, it is a comprehensive collection of financial data that provides a detailed view of daily trading activity on the CBX. The dataset has been carefully designed to include important components of each trading day, such as the opening price (open), the day's high (high), the closing price at the end of the trading session (close), and volume. We select some of the data for presentation and the results are shown in Table 1.

Table 1. Selected data sets.

date	open_value	high_value	low_value	last_value
2015/12/30	1689.22	1689.71	1673.62	1689.63
2015/12/29	1675.79	1691.02	1673.37	1689.94

2015/12/28	1655.92	1677.17	1652.76	1675.88
2015/12/23	1647.66	1655.77	1641.41	1655.77
2015/12/22	1655.71	1655.71	1642.6	1647.67
2015/12/21	1657.62	1657.86	1649.55	1656.75
2015/12/18	1659.32	1659.58	1650.21	1657.62
2015/12/17	1660.69	1660.69	1656.11	1658.96
2015/12/16	1656.43	1660.81	1652.61	1660.81
2015/12/15	1653.74	1656.34	1646.09	1656.34
2015/12/14	1659.8	1661.31	1652.81	1653.09
2015/12/11	1672.94	1673.43	1657.84	1659.99
2015/12/10	1680.02	1680.25	1673.3	1673.3
2015/12/9	1679.15	1681.56	1675.01	1680.02
2015/12/8	1685.04	1685.39	1676.48	1679.34
2015/12/7	1673.22	1685.7	1669.19	1682.51
2015/12/4	1677.37	1679.78	1671.6	1673.21

We select the time and the opening price in the data to be used in our experiment, which aims to predict the daily opening price of a stock through a time series algorithm.

III. Method

A. Extreme Learning Machine

Extreme Learning Machine (ELM) is an emerging machine learning algorithm mainly used to train single hidden layer feedforward neural networks. Compared with traditional neural network training methods, ELM has significant advantages in speed and efficiency and is especially suitable for handling large-scale datasets. The structure of the Extreme Learning Machine is shown schematically in Figure 1.

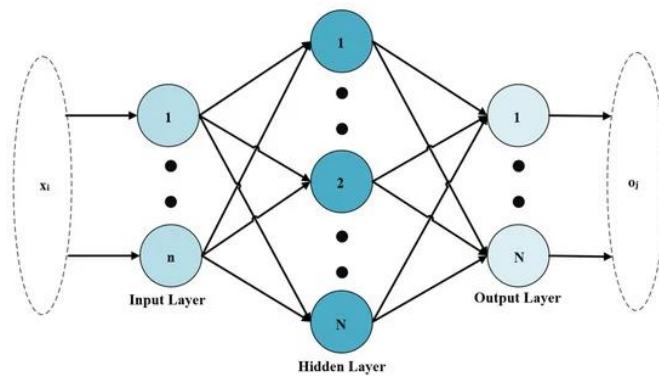


Figure 1. The structure of the Extreme Learning Machine.

The basic principle of an Extreme Learning Machine is to map the input data through randomly generated hidden layer nodes and then calculate the output weights through simple linear regression. Specifically, ELM first randomly initialises the parameters of the hidden layer nodes, including weights and biases. These parameters are not adjusted during the training process, but are fixed. This process allows ELM to quickly construct a high-dimensional feature space, thus improving the model's performance [7].

In ELM, the number of hidden layer nodes and the activation function are pre-set. By linearly combining the input samples, a hidden layer output matrix is generated H . Each column of H corresponds to the output of a hidden layer node, which are processed by activation functions (e.g., sigmoid, ReLU, etc.). Since these hidden layer nodes are randomly generated, the overfitting problem can be effectively avoided. Once the hidden layer output matrix H is obtained, the least squares method can be used to solve the output weights.

B. The Crowned Porcupine Optimisation

The Crowned Porcupine Optimisation (CHO) algorithm is an emerging population intelligence optimisation algorithm inspired by the behavioural traits of the Crowned Porcupine. The algorithm aims to solve complex optimisation problems by simulating the foraging and survival strategies of crowned porcupines in nature [8].

The Crown Hedgehog Optimization Algorithm (CHOA) is an emerging nature-inspired optimisation algorithm inspired by the foraging and survival strategies of the Crown porcupine. The algorithm shows good performance in dealing with complex optimisation problems, especially for high-dimensional, nonlinear and multi-peaked functions.

The Crown Hedgehog is a small mammal that lives on the edges of grasslands and forests and is known for its unique foraging behaviour. When searching for food, this animal evaluates its surroundings through multiple senses, including smell, sight and hearing, and makes decisions accordingly. The Crowned Porcupine Optimisation Algorithm simulates this process to find the optimal solution by establishing a mechanism for information exchange and cooperation between individuals.

The implementation steps of the algorithm are:

Initialisation: a set of initial individual positions are randomly generated, each representing a potential solution, and their fitness values are calculated.

2. fitness evaluation: each individual is evaluated for fitness to determine their performance in the current solution space.

3. Information sharing: Based on the fitness value, the better-performing individual is selected as the "leader", and their information is shared with other individuals. This process allows the group to move in a better direction.

4. Position update: The position of each individual is updated based on the leader's information and its state. Position updating usually includes random perturbations and strategies for moving towards the leader to balance exploration and exploitation.

5. Iteration: repeat steps 2 to 4 until the stopping condition is met (e.g., the maximum number of iterations or the fitness reaches a preset threshold).

6. Output: return the individual's position corresponding to the best fitness, i.e., the optimal solution.

In conclusion, the Crowned Porcupine Optimization Algorithm is a novel and efficient group intelligence optimisation method, which provides a new way of thinking for solving complex optimisation problems by simulating the behaviour of crowned porcupines in nature.

C. Adaptive Bandwidth Kernel Function Density Estimation

Adaptive Bandwidth Kernel Function Density Estimation (ABKDE) is a nonparametric probability density function estimation method. Compared with traditional kernel density estimation (KDE), ABKDE improves the accuracy and flexibility of the estimation by adaptively adjusting the bandwidth, mainly when the data is unevenly distributed or has a complex structure.

Kernel density estimation is a commonly used statistical method for inferring potential probability density functions from sample data. The basic idea is to obtain an overall density estimate by placing a kernel function (e.g., a Gaussian kernel) on each observation and superimposing the contributions of all kernel functions. Traditional KDEs use a fixed bandwidth to control the width of the kernel function, which works well when the data is more evenly distributed. Still, in the presence of significant data clustering or sparse regions, a fixed bandwidth can lead to overly smooth or overly volatile results. The specific principle of kernel density estimation is shown in Figure 2.

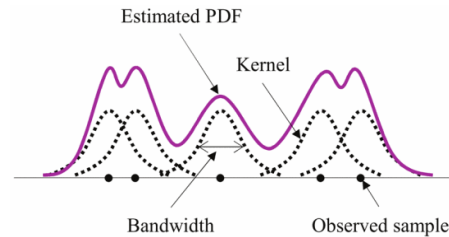
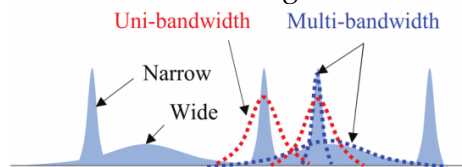


Figure 2. The specific principle of kernel density estimation.

The core idea of adaptive bandwidth is to dynamically adjust the bandwidth according to the distribution of local data points. In regions with denser data points, a smaller bandwidth can capture more details. In contrast, in sparse regions, a larger bandwidth is used to avoid noise from significantly affecting the results. This flexibility allows ABKDE to reflect the actual distribution characteristics of the data more accurately. The specific principle of adaptive bandwidth is shown in Figure 3.



The specific principle of adaptive bandwidth.

The implementation steps of the algorithm are:

1. select the kernel function: first, select a suitable kernel function, e.g., Gaussian kernel, Epanechnikov kernel, etc., which will determine the shape of the influence of each data point on the final density estimation.
2. Calculate local density: For each sample point, calculate the number of data points within a specific range around it and determine the required bandwidth at that location based on the information of those points. Commonly used methods include distance-based methods such as the k-Nearest Neighbour (k-NN) method, which sets the local bandwidth by selecting the k nearest neighbours.
3. Update bandwidth: Dynamically adjust the bandwidth corresponding to each sample point according to the number and distribution of local data points. Typically, denser regions get smaller bandwidths, while sparse regions get larger bandwidths.
4. Perform density estimation: the final kernel density estimation is performed using the adaptive bandwidth, where each sample point corresponds to its adaptively computed bandwidth, and all results are summed to obtain the overall probability density function.
5. Post-processing: Optionally, the results can be smoothed to eliminate possible small-scale fluctuations and improve interpretability.

In conclusion, the adaptive bandwidth kernel function density estimation algorithm overcomes some of the shortcomings of the fixed bandwidth method by dynamically adjusting the data influence in the local region, making the probability density estimation more flexible and accurate [9].

An Extreme Learning Machine Stock Price Prediction Algorithm Based on the Optimisation of the Crown Porcupine Optimisation Algorithm with an Adaptive Bandwidth Kernel Function Density Estimation Algorithm

A new ELM optimisation framework is formed by combining the crown porcupine optimisation algorithm with the adaptive bandwidth kernel function. In this framework, CHOA is firstly utilised to search for the optimal hyperparameter configurations, including the number of hidden layer nodes, the type of activation function, and the adaptive bandwidth parameter. In this way, the large amount of time and effort required for manual parameter tuning can be effectively avoided.

Secondly, in the feature mapping stage, the input data is processed by the adaptive bandwidth kernel function, which enables the ELM to better capture the nonlinear features in the data. This

process not only improves the fitting ability of the model on the training set, but also enhances its generalisation performance on the test set [10].

IV. Experiment

In the division of the dataset, we divide the training set and test set according to the ratio of 7:3, 70% of the data is used for training, and 30% of the data is used for testing the trained model. This experiment is conducted using a local server with a 3090 graphics card and 32G of RAM, this experiment is based on Matlab R2023b.

The adaptation curve during the model training process is shown in Figure 4, which shows that the training effect of the model becomes better gradually from the model adaptation curve.

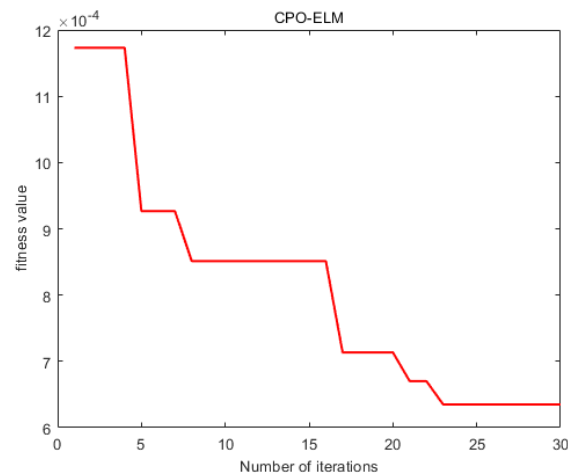


Figure 3. The adaptation curve.

Using the test set to test the prediction effect of the model, draw the residual plot of the model, i.e. the difference between the actual value and the predicted value, the results are shown in Figure 5, from Figure 5, it can be seen that the residuals are mainly distributed in the top and bottom of 0 and fluctuation amplitude is not large, the RMSE value of 0.018, which indicates that the model in the test set of the performance of the effect of the model, the model's ability of the generalisation of the model is relatively strong.

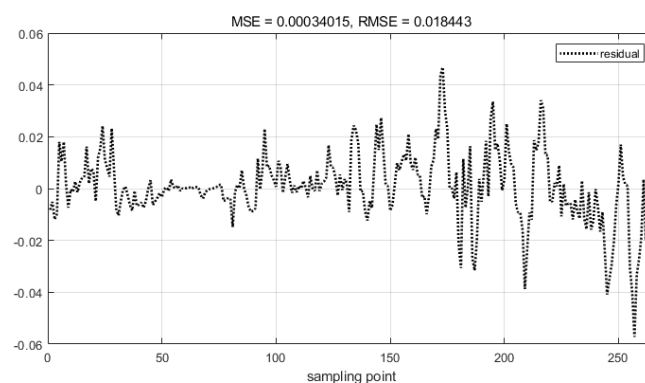


Figure 4. The difference between the actual value and the predicted value.

The model prediction effectiveness is evaluated by outputting 95% confidence kernel density estimation curves, optimised adaptive bandwidth kernel density estimation curves and unoptimised fixed bandwidth kernel density estimation curves and the results are shown in Figure 6.

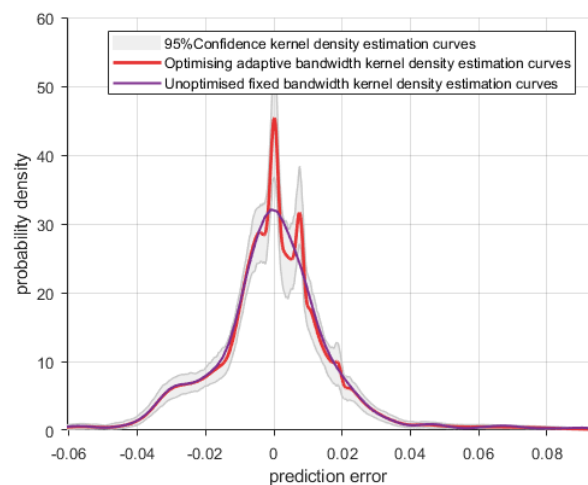


Figure 5. Optimised adaptive bandwidth kernel density estimation curves.

Confidence intervals for the predicted values are plotted around the actual values and the results are shown in Figure 7.

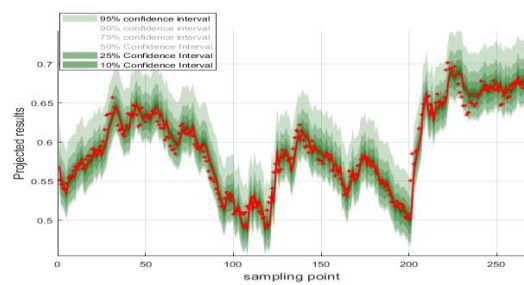


Figure 6. Confidence intervals for the predicted values.

The scatter plot of predicted and actual values of the test set is plotted and the results are shown in Figure 8. As can be seen from the results, $R=0.968$, which indicates that the model has a good prediction effect and can predict the stock price well.

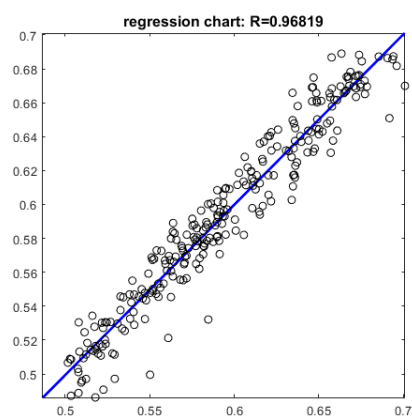


Figure 7. The scatter plot of predicted and actual values of the test set .

The prediction effectiveness of the training and test sets is evaluated using R^2 , MAE, RMSE and MAPE and the results are shown in Table 2.

Table 2. modelling assessment.

R^2	MAE	RMSE	MAPE
-------	-----	------	------

Training Set	0.98768	0.0084355	0.011914	0.01399
Test set	0.95494	0.0081253	0.010891	0.013614

The R2 of the training set is 0.987, while the R2 of the test set is 0.954, which is not much different from each other, indicating that the model has a better generalisation ability. Meanwhile, on the MAE, RMSE and MAPE, both the training set and test set perform well.

V. Conclusion

In this paper, we delve into the potential application of Extreme Learning Machines (ELMs) in stock price prediction and optimise them effectively with the Crown Porcupine optimisation algorithm and the Adaptive Bandwidth Kernel Function Density Estimation algorithm. Specifically, we divide the dataset into training and testing sets in the ratio of 7:3 to ensure that the model can be effectively validated in different data environments.

In addition, we calculated the correlation coefficient R and obtained a value of 0.968, a value that further emphasises the reliability of the model in terms of stock price prediction. By comparing the coefficient of determination R² for the training and test sets, we found that the training set is 0.987 while the test set is 0.954, which is not much different. This indicates that our model not only fits the training data well, but also has a strong generalisation ability to effectively deal with unseen data.

In summary, this study has deeply optimised the ELM by introducing advanced optimisation algorithms and kernel function density estimation techniques, so that it demonstrates excellent performance in the stock price prediction task. In the future, with the increasing volume and complexity of data in the financial market, this method is expected to provide new solutions to a wider range of financial prediction problems. Meanwhile, it also provides a useful reference for further research on applying machine learning algorithms in finance.

VI. Discussion

Extreme Learning Machine (ELM), as an emerging machine learning method, has shown good performance in the field of stock price prediction. Combining the Crown Porcupine optimisation algorithm and the adaptive bandwidth kernel function density estimation algorithm, the ELM is effectively optimised to make it more efficient and accurate in handling complex financial data. This optimisation not only improves the learning ability of the model, but also enhances its sensitivity to market fluctuations, thus providing investors with more reliable decision support.

In our study, the correlation coefficient R reaches 0.968, a high value that fully demonstrates the model's reliability in stock price prediction. Usually, the closer the correlation coefficient is to 1, the stronger the linear relationship between the predicted and actual values. By comparing the coefficient of determination R² between the training and test sets, we find that the training set is 0.987 while the test set is 0.954, which is not much different. This result suggests that although the model was well fitted during training, it also has a strong generalisation ability to deal effectively with unseen data. This is particularly important for financial markets, where the market environment changes rapidly, and the model needs to be adaptable to maintain a stable predictive performance.

In conclusion, by combining the Crown Porcupine optimisation algorithm with the adaptive bandwidth kernel function density estimation algorithm, we have successfully enhanced the effectiveness of the Extreme Learning Machine in stock price prediction. These research results provide new perspectives for academics and strong support for investment decisions in practice. In the future, we can continue to explore more optimisation methods and apply them to other financial fields to improve model performance further and create greater value for investors.

References

1.

Xu, Y., Liu, Y., Xu, H., & Tan, H. (2024). AI-Driven UX/UI Design: Empirical Research and Applications in FinTech. *International Journal of Innovative Research in Computer Science & Technology*, 12(4), 99-109.

2.

Liu, Y., Xu, Y., & Song, R. (2024). Transforming User Experience (UX) through Artificial Intelligence (AI) in interactive media design. *Engineering Science & Technology Journal*, 5(7), 2273-2283.

3. Zhang, P. (2024). A STUDY ON THE LOCATION SELECTION OF LOGISTICS DISTRIBUTION CENTERS BASED ON E-COMMERCE. *Journal of Knowledge Learning and Science Technology* ISSN: 2959-6386 (online), 3(3), 103-107.
4. Zhang, P., & Gan, L. I. U. (2024). Optimization of Vehicle Scheduling for Joint Distribution in Logistics Park based on Priority. *Journal of Industrial Engineering and Applied Science*, 2(4), 116-121.
5. Li, H., Wang, S. X., Shang, F., Niu, K., & Song, R. (2024). Applications of Large Language Models in Cloud Computing: An Empirical Study Using Real-world Data. *International Journal of Innovative Research in Computer Science & Technology*, 12(4), 59-69. (加 10 个)
6. Ping, G., Wang, S. X., Zhao, F., Wang, Z., & Zhang, X. (2024). Blockchain Based Reverse Logistics Data Tracking: An Innovative Approach to Enhance E-Waste Recycling Efficiency.
7. Xu, H., Niu, K., Lu, T., & Li, S. (2024). Leveraging artificial intelligence for enhanced risk management in financial services: Current applications and future prospects. *Engineering Science & Technology Journal*, 5(8), 2402-2426.
8. Shi, Y., Shang, F., Xu, Z., & Zhou, S. (2024). Emotion-Driven Deep Learning Recommendation Systems: Mining Preferences from User Reviews and Predicting Scores. *Journal of Artificial Intelligence and Development*, 3(1), 40-46.
9. Wang, Shikai, Kangming Xu, and Zhipeng Ling. "Deep Learning-Based Chip Power Prediction and Optimization: An Intelligent EDA Approach." *International Journal of Innovative Research in Computer Science & Technology* 12.4 (2024): 77-87.
10. Li A, Zhuang S, Yang T, Lu W, Xu J. Optimization of logistics cargo tracking and transportation efficiency based on data science deep learning models. *Applied and Computational Engineering*. 2024 Jul 8;69:71-7.
11. Xu, J., Yang, T., Zhuang, S., Li, H. and Lu, W., 2024. AI-based financial transaction monitoring and fraud prevention with behaviour prediction. *Applied and Computational Engineering*, 77, pp.218-224.
12. Guo, L., Song, R., Wu, J., Xu, Z., & Zhao, F. (2024). Integrating a Machine Learning-Driven Fraud Detection System Based on a Risk Management Framework.
13. Ling, Z., Xin, Q., Lin, Y., Su, G. and Shui, Z., 2024. Optimization of autonomous driving image detection based on RFACnv and triplet attention. *Applied and Computational Engineering*, 77, pp.210-217.
14. Bao, W., Xiao, J., Deng, T., Bi, S., & Wang, J. (2024). The Challenges and Opportunities of Financial Technology Innovation to Bank Financing Business and Risk Management. *Financial Engineering and Risk Management*, 7(2), 82-88.
15. He, Z., Shen, X., Zhou, Y., & Wang, Y. (2024, January). Application of K-means clustering based on artificial intelligence in gene statistics of biological information engineering. In *Proceedings of the 2024 4th International Conference on Bioinformatics and Intelligent Computing* (pp. 468-473).
16. Gong, Y., Zhu, M., Huo, S., Xiang, Y., & Yu, H. (2024, March). Utilizing Deep Learning for Enhancing Network Resilience in Finance. In *2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE)* (pp. 987-991). IEEE.
17. Xin, Q., Xu, Z., Guo, L., Zhao, F., & Wu, B. (2024). IoT Traffic Classification and Anomaly Detection Method based on Deep Autoencoders.
18. Fruehwirth, Jane Cooley, Alex Xingbang Weng, and Krista MPerreira."The effect of social media use on mental health of college students during the pandemic." *Health Economics* (2024).
19. Li, J., Wang, Y., Xu, C., Liu, S., Dai, J., & Lan, K. (2024). Bioplastic derived from corn stover: Life cycle assessment and artificial intelligence-based analysis of uncertainty and variability. *Science of The Total Environment*, 174349.
20. Yang, T., Li, A., Xu, J., Su, G. and Wang, J., 2024. Deep learning model-driven financial risk prediction and analysis. *Applied and Computational Engineering*, 77, pp.196-202.
21. Haowei, Ma, et al. "CRISPR/Cas-based nanobiosensors: A reinforced approach for specific and sensitive recognition of mycotoxins." *Food Bioscience* 56 (2023): 103110.
22. Zhou, Y., Zhan, T., Wu, Y., Song, B., & Shi, C. (2024). RNA Secondary Structure Prediction Using Transformer-Based Deep Learning Models. *arXiv preprint arXiv:2405.06655*.
23. Weng A. Depression and Risky Health Behaviors[J]. Available at SSRN 4843979.
24. Liu, B., Cai, G., Ling, Z., Qian, J., & Zhang, Q. (2024). Precise Positioning and Prediction System for Autonomous Driving Based on Generative Artificial Intelligence. *Applied and Computational Engineering*, 64, 42-49.
25. Cui, Z., Lin, L., Zong, Y., Chen, Y., & Wang, S. (2024). Precision Gene Editing Using Deep Learning: A Case Study of the CRISPR-Cas9 Editor. *Applied and Computational Engineering*, 64, 134-141.
26. Zhang, X., 2024. Machine learning insights into digital payment behaviors and fraud prediction. *Applied and Computational Engineering*, 67, pp.61-67.
27. Zhang, X. (2024). Analyzing Financial Market Trends in Cryptocurrency and Stock Prices Using CNN-LSTM Models.
28. Ping G, Wang S X, Zhao F, et al. Blockchain Based Reverse Logistics Data Tracking: An Innovative Approach to Enhance E-Waste Recycling Efficiency[J]. 2024.

29. Shang F, Shi J, Shi Y, et al. Enhancing E-Commerce Recommendation Systems with Deep Learning-based Sentiment Analysis of User Reviews[J]. International Journal of Engineering and Management Research, 2024, 14(4): 19-34.
30. Xu Y, Liu Y, Xu H, et al. AI-Driven UX/UI Design: Empirical Research and Applications in FinTech[J]. International Journal of Innovative Research in Computer Science & Technology, 2024, 12(4): 99-109.
31. Xu H, Niu K, Lu T, et al. Leveraging artificial intelligence for enhanced risk management in financial services: Current applications and future prospects[J]. Engineering Science & Technology Journal, 2024, 5(8): 2402-2426.

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