

The application of machine learning methods for risk analysis of investment projects

Dilan Özcan Yaylalı*, İman Askerzade

Department of Computer Engineering Graduate School of Natural and Applied Sciences, Ankara University, Ankara, Turkey

ARTICLE INFO	ABSTRACT
<hr/> <i>Article history:</i> Received 17.09.2025 Received in revised form 14.10.2025 Accepted 04.11.2025 Available online 20.03.2026 <hr/> <i>Keywords:</i> Machine learning Risk analysis Investment projects	<hr/> <i>The evaluation of investment projects is the centerpiece of all investment activities. Given the substantial financial losses that can result from various risk variables and uncertainty, risk analysis of investment projects is particularly important. Most frequently, investment appraisals employ cost-benefit metrics to guide investment decisions. Aiming at the risk analysis of the investment projects, we examined an application and compared several machine learning methods including deep learning techniques, to build a risk assessment model for investment projects and analyze investment risks scientifically and effectively. The suggested deep learning approach, which combines long short-term memory and convolutional networks, yielded the best performance. The study proves that using machine learning techniques for the risk analysis of investment projects enables businesses to detect and manage investment risks.</i> <hr/>

1. Introduction

Risk analysis of investment projects is critical because of the significant economic losses that can be generated by risk factors. It is necessary to consider numerous factors and techniques while doing risk analysis on investment projects. An unpredictably unclear future, involving numerous kinds of risk and uncertainty, frequently affects the expected outcomes for feasibility as well as profitability. In most cases, investment appraisals make use of tools like cost-benefit analysis and risk analysis. Net Present Value (NPV) and Internal Rate of Return (IRR) are the most commonly used metrics in cost-benefit analysis (excluding the process of risk analysis). Sensitivity analysis, probability analysis, break-even analysis, decision trees, uncertainty analysis, and the Monte-Carlo simulation technique are well-known and commonly used approaches [2]. According to a review of the literature and evaluation reports, it is typical to find that the assessment of such values does not provide enough information for a meaningful decision, particularly for energy investment projects such as the production of oxygen. Deep learning algorithms have been extensively studied in the realm of risk or investing. To improve investors' investment strategy and rate of return, Xie and Liu examined and discussed the use of deep learning to financial data processing and investment models, as well as investment plan establishment. [4, 5] Convolutional Neural Network (CNN) was shown to be the most effective model for financial forecasting by Sohangir et al. Another research found that the accuracy of financial risk predictions and the optimization of investment portfolios were critical, so he compared the performance of Long Short-Term Memory (LSTM) and Recurrent Neural

*Corresponding author

E-mail addresses: ozcan.dilan@gmail.com (Dilan Özcan Yaylalı), imasker@eng.ankara.edu.tr (İman Askerzade)

www.icp.az/2026/1-02.pdf <https://doi.org/10.54381/icp.2026.1.02>

2664-2085/ © 2026 Institute of Mathematics. All rights reserved

Network (RNN) in classification; the results indicated that the LSTM had a better performance, which was of great significance for maximizing investment profits [1, 3]. Based on the existing studies, the paper proposes to use LSTM which is a specialized RNN technique and CNN as deep learning techniques to build a risk assessment model for investment projects and analyze investment risks scientifically and effectively. Using these deep learning techniques, we will assess the risk analysis of the investment projects that businesses may identify and mitigate investment risks. CNN is distinguished by its focus on prominent elements throughout the visual field, making it extensively utilized in feature engineering. LSTM possesses the ability to expand in accordance with temporal sequences and is extensively utilized in time series analysis. A stock forecasting model is created based on the properties of CNN and LSTM. Figure 1 illustrates the model structure diagram, which mostly comprises CNN and LSTM components, including the input layer, one-dimensional convolution layer, pooling layer, LSTM hidden layer, and fully connected layer.

2. Description of the processing system

Architecture of The Machine Learning Method

CNN is distinguished by its focus on prominent elements throughout the visual field, making it extensively utilized in feature engineering. LSTM possesses the ability to expand in accordance with temporal sequences and is extensively utilized in time series analysis. A stock forecasting model is created based on the properties of CNN and LSTM. Figure 1 illustrates the model structure diagram, which mostly comprises CNN and LSTM components, including the input layer, one-dimensional convolution layer, pooling layer, LSTM hidden layer, and fully connected layer.

CNN is a model introduced by LeCun et al. in 1998. CNN is a type of feedforward neural network that demonstrates strong efficacy in image processing and natural language processing. It may be efficiently utilized for time series forecasting. The local perception and weight sharing of CNN significantly diminish the parameter count, hence enhancing model learning efficiency. CNN primarily consists of two components: the convolutional layer and the pooling layer. Each convolutional layer comprises many convolutional kernels, and its computational algorithm is presented in equation (1). Following the convolution operation in the convolution layer, the data's features are extracted; however, the dimensionality of these features is excessively high. To address this issue and mitigate the training costs of the network, a pooling layer is incorporated subsequent to the convolution layer to diminish the feature dimensions.

LSTM is a network architecture introduced by Schmidhuber et al. in 1997. LSTM is a network architecture developed to address the persistent issues of gradient expansion and gradient vanishing in RNNs. It has been extensively utilized in speech recognition, emotional analysis, and text analysis due to its inherent memory and capacity for very precise predicting. Recently, it has been utilized in stock market forecasting to address the persistent issues of gradient expansion and gradient vanishing in RNNs. It has been extensively utilized in speech recognition, emotional analysis, and text analysis due to its inherent memory and capacity for very precise predicting. In recent years, it has been utilized in stock market forecasting, investment predictions and risk management.

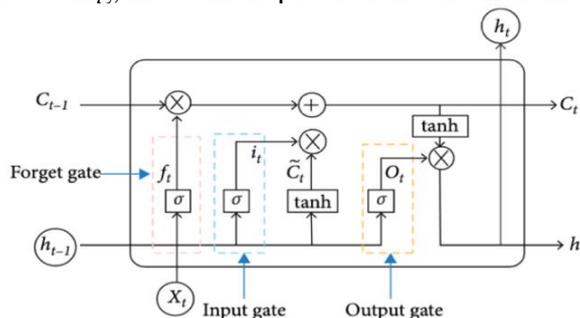


Fig. 1. LSTM Architecture

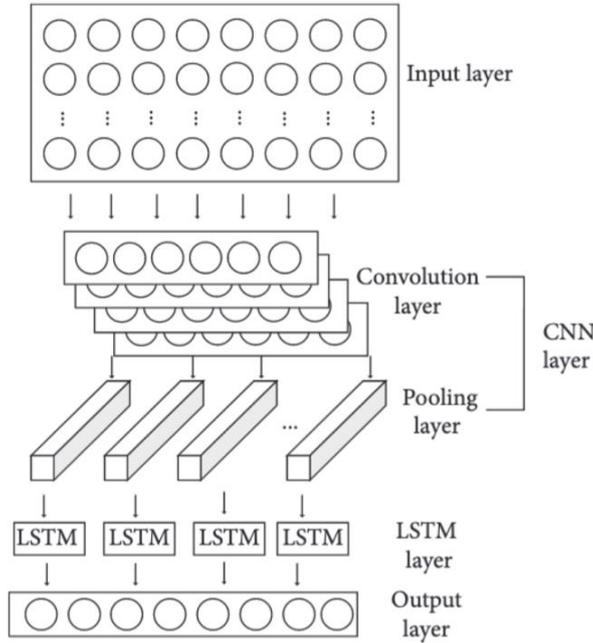


Fig. 2. Architecture of CNN and LSTM Memory Cell

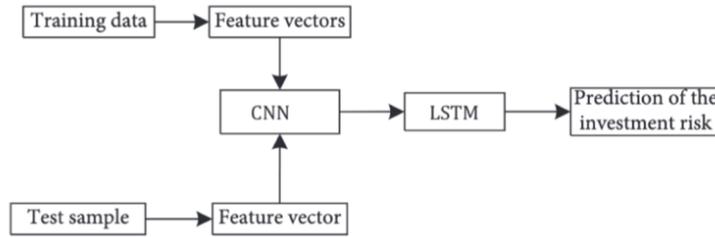


Fig. 3. Prediction Model of Investment Risks

Mathematical Background of CNN and LSTM Models

As the literature indicates, CNN and LSTM models can be used in the training of evaluation and prediction on risk analysis of investment projects. When it comes to solving the gradient disappearance problem, only the LSTM algorithm, which is a specialized recurrent neural network, can handle several original data inputs at once and take into account their relationship over time. In the future, the information in its hidden layer can be communicated to either the output end or the hidden layer itself. Additionally, the weight of the LSTM algorithm can be updated continuously to minimize gradient expansion or disappearance. LSTM has advantages because it has a long-term memory and is easy to implement; the problems of gradient vanishing and gradient explosion that occur during long sequence training are addressed. LSTM algorithm training consists of four stages:

Step 1: Calculate the output of LSTM hidden layer. Let w , o_t , c , and f denote output gate, cell state, forgetting gate, and input gate, respectively, let b denote offset, let w denote weight coefficient matrix, and T ($\tan h$) and S ($sigmoid$) are activation functions; then the expressions of forgetting gate and input layer are shown in the two following equations (1) and (2):

$$f_t = \sigma(w_f[h_t - 1, x_t] + b), f_t \in [0,1] \quad (1)$$

$$i_t = \sigma(w_i[h_t - 1, x_t] + b_i)$$

$$cl_t = \tan h(w_c[h_t - 1, x_t] + b_c) \quad (2)$$

Update the cell state:

$$c_t = f_t * c_{t-1} + i_t * cl_t \quad (3)$$

The model output is obtained by activating the function:

$$\begin{aligned} o_t &= \sigma(w_o[h_t - 1, x_t] + b) \\ h_t &= o_t * \tan h(c_t) \end{aligned} \quad (4)$$

Step 2: The error δ of each LSTM cell is calculated by backpropagation, including the error terms in time and network level. Let δ_t denote the error term at time t , let δ_{t-1} denote the error term at time $t - 1$, and let δ_k denote the error term at time k ; the backpropagation delay of the error term is expressed as:

$$\begin{aligned} \delta_t^T &= \frac{\partial E}{\partial h_t} \\ \delta_{t-1}^T &= \frac{\partial E}{\partial h_{t-1}} \\ \delta_k^T &= \prod_{j=k}^{t-1} \delta_{o,j}^T w_{oh} + \delta_{f,j}^T w_{fh} + \delta_{i,j}^T w_{ih} \end{aligned} \quad (5)$$

Error terms are reversely transmitted to the upper layer along the network:

$$\delta = (\delta_{f,t}^T w_{fx} + \delta_{i,t}^T w_{ix} + \delta_{o,t}^T w_{ox} \cdot f'(net_t^{l-1})) \quad (6)$$

Step 3: Calculate the weight gradient according to the error.

$$\frac{\partial E}{\partial w_f} = \sum_{j=1}^t \delta_{fj} \quad (7)$$

w is the error with the fastest value at the point. In the decision of the company in the multi-index decision for error calculation, it finds out the index with the fastest change.

Step 4: Update the weights based on the previous step. An additional deep learning model used in this paper is a hierarchical CNN layer. A hierarchical CNN is made up of numerous CNN layers that can be interconnected and covered by CNN architecture. Additionally, non-linearity is incorporated into each layer of the model, which helps the model zero in on the most critical aspects of the sequence as a whole.

Gated linear units (GLU) is used to generate the hierarchical CNN layer's output and can be calculated theoretically as follows:

$$v([XY]) = X \odot \sigma(Y) \quad (8)$$

in which $X, Y \in \mathbb{R}^d$ represent GLU's inputs, and \odot denotes the element-wise production function σ is utilized, and this paper uses sigmoid in practice. When CNN layers are stacked, some information may be lost at the higher layers. In order to connect the input and output layers, this work relies on the residual connections. Adding a residual mechanism to the equation results in the following:

$$h_t^k = v\left(W^l \left[h_{t-\frac{1}{2}}^{k-1}, \dots, h_{t+\frac{1}{2}}^{k-1} \right] + b_w^k\right) + h_t^{k-1} \quad (9)$$

Afterward a fully connected layer is adopted to transform the outputs of CNN to hidden state h^k , so that the self-attention mechanism can be adopted. After hidden state, the optimization layer will work to further process the output of the hierarchical CNN layer, which is able to calculate the weight of each subindicator. The equation that calculates the weights based on h^k could be represented as:

$$f_h = \tan h(hW_a h^T + f - b_a) \quad (10)$$

$$A = softmax(f_h) \quad (11)$$

$$g = Ah = (g_1, g_2, \dots, g_n) \quad (12)$$

where f_h is the generated rating score of each risk indicator, A is the subindex's weight, and g is the output of the whole self attention layer which will be processed by CRF layer and classification layer afterwards. W_a denotes the attention parameter matrix, while b_a denotes bias parameter. Then CRF layer will be compiled. The input layer is the output of the optimization layer. For a sequence S , the input matrix I is denoted as $I = [I_1, I_2, \dots, I_n - 1]^T$, in which $I \in n \times d$. Let $I_{i,j}$ denote the probability score of the j^{th} score of the i^{th} element in the sequence. Given the sequence $S = \{x_1^k, x_2^k, \dots, x_n^k\}$ the CRF score could be calculated as follows:

$$f(X, y) = \sum_{i=0}^{s+1} O_{y_i, y_{i+1}} + \sum_{i=1}^s I_{i, y_i} \quad (13)$$

in which T denotes the transition matrix, and $O_{i,j}$ is the corresponding transition score. To obtain the final risk assessment, the classifier layer is used. Such a process can be illustrated mathematically as follows:

$$p_t^i(\theta) = \frac{\exp(y_t^i)}{\sum_{j=1} \exp(y_t^j)} \quad (14)$$

where θ represents all parameters, T is the label set, and $p_t^i(\theta)$ is the output of the risk analysis.

3. Experiments

To demonstrate the effectiveness of CNN-LSTM, a comparative analysis was conducted against CNN, RNN, LSTM, and CNN-RNN. This comparison utilized the same training and test set data within a consistent operating environment.

Data

In this experiment, Oxygen Prices of Producer Pricer Index of Oxygen Industry from Federal Reserved Economical Data (FRED) database by the Research division of the Federal Reserve Bank of St.Louis is used as the main source of the experimental data. The daily trading dataset consists of 10228 data from February 1, 2020, to February 1, 2025. Each piece of data contains seven items, namely, opening price, highest price, lowest price, closing price, volume, ups and downs, and change. We take the data of the first 8401 data as training set and the data of the last 1827 data as test set.

Evaluation

To assess the forecasting performance of CNN-LSTM, the evaluation criteria employed include mean absolute percentage error (MAPE), root mean square error (RMSE), and R-square (R2).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}, \quad (15)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{(\hat{y}_i - y_i)}{y_i} \right| \times 100\%, \quad (16)$$

$$MA = 1 - MAPE. \quad (17)$$

4. Results

Following the utilization of the processed training set data for training the CNN, RNN, and LSTM models. The model, completed through training using CNN-RNN and CNN-LSTM, is employed to predict the calculated risk. The comparative results of the six methods are presented in

Figures 4–8 and Table I.

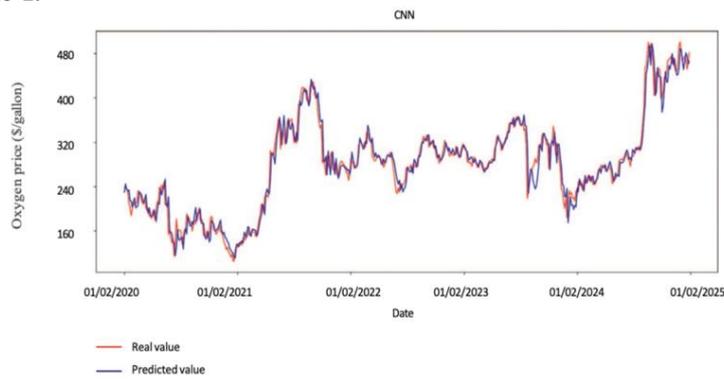


Fig. 4. Comparison of the Predicted Value and the Real value for CNN

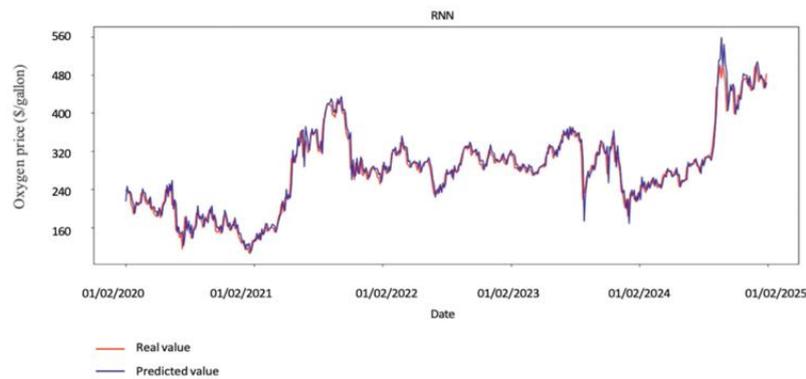


Fig. 5. Comparison of the Predicted Value and the Real Value for RNN

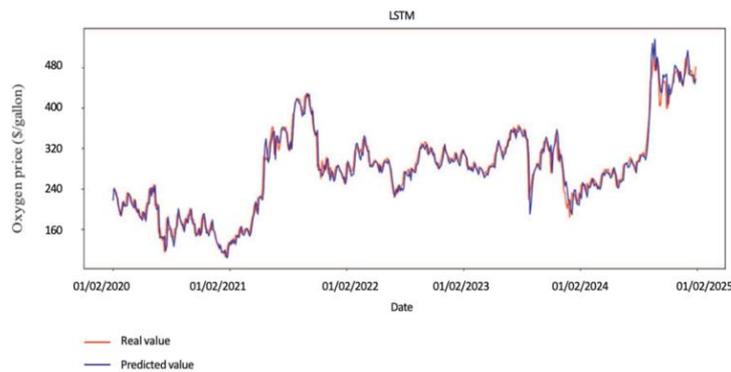


Fig. 6. Comparison of the Predicted Value and the Real Value for LSTM

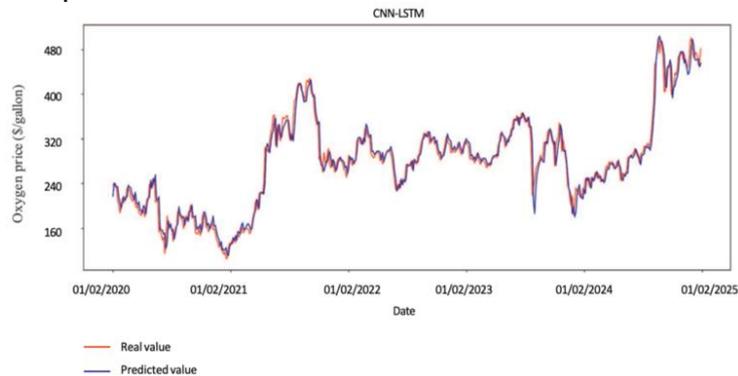


Fig. 7. Comparison of the Predicted Value and the Real Value for CNN-LSTM

Table 1.
Comparison of prediction performance of different machine learning models

	RMSE	MA (%)
CNN-LSTM	0.13	97.2
CNN	0.20	95.4
RNN	0.18	96.2
LSTM	0.14	96.9

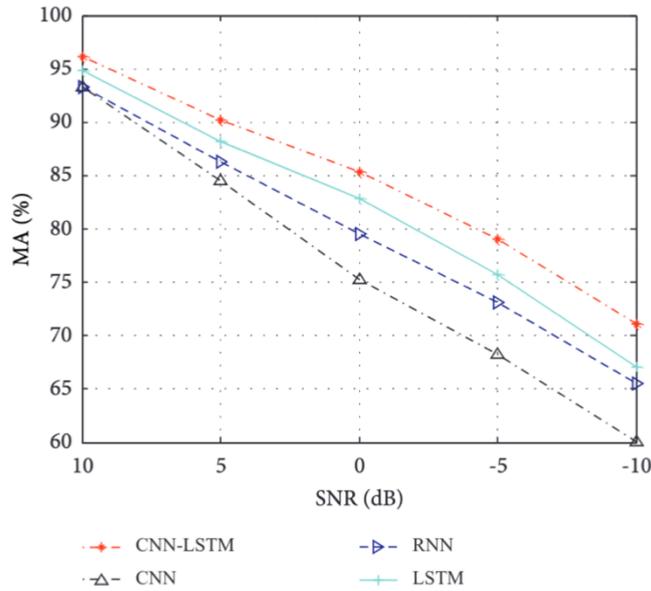


Fig. 8. Comparison of Prediction of Performance under Noises

5. Conclusion

This research proposes a CNN-LSTM model to predict the risk of investment projects based on the chronological aspects of oxygen price data. CNN is employed to extract features from the input data. LSTM is employed to analyze the extracted feature of oxygen data and forecast the oxygen price. This research utilizes data from Oxygen Prices of Producer Pricer Index of Oxygen Industry from Federal Reserve Economics Data (FRED) database by the Research division of the Federal Reserve Bank of St. Louis to validate the experimental findings. The experimental results indicate that the CNN-LSTM exhibits the highest forecasting accuracy and superior performance relative to the MLP, CNN, RNN, LSTM, and CNN-RNN models. MAPE and RMSE are the lowest among all methods, while R^2 approaches 1. The CNN-LSTM model is appropriate for oxygen price forecasting and risk management and offers valuable insights for investors aiming to define risk of investment projects. CNN-LSTM offers valuable insights for researchers studying financial risk management of investment projects.

References

- [1] Networks Expert Systems with Application. 132 (2019) pp.99-109.
- [2] H. Sariaslán, Planning and analyzing foreign direct investment projects, IGI Global, (2019).
- [3] S. Sohangir, D. Wang, A. Pomeranets et al, Big Data: Deep learning for financial sentiment analysis, Journal of Big Data. 5 No.1, (2018) p.3.
- [4] J.T. Song, Construction of corporate investment decision support model based on deep learning, Scientific Programming. (2021) pp.1-7.

- [5] M. Xie, H. Li, Y. Zhao, Blockchain financial investment based on deep learning network algorithm, *Journal of Computational and Applied Mathematics*. 372 (2020) 112723.
<https://doi.org/10.1016/j.cam.2020.112723>
- [6] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*. 86 No.11 (1998) pp.2278-2324.
- [7] S. Hochreiter, J. Schmidhuber, Long short-term memory, *MIT Press*. 9 No.8 (1997) pp.1735-1780.
- [8] D. Ozcan, I. Askerzade, G.E. Bostanci, Gazi Erkan, Risk analysis of an investment project using deep learning techniques, *Proceeding of the International Conference on Control and Optimization with Industrial Applications*. (2022) pp.319-329.