

Ứng dụng học máy trong đánh giá rủi ro tài chính của các công ty niêm yết trên thị trường chứng khoán Việt Nam

TÓM TẮT

Quản lý rủi ro tài chính là điều cần thiết đối với các doanh nghiệp vì nó giúp ngăn ngừa tổn thất và tối đa hóa lợi nhuận. Do quá trình này phụ thuộc nhiều vào việc ra quyết định dựa trên dữ liệu, học máy mang lại tiềm năng phát triển các phương pháp và công nghệ sáng tạo. Trong bài báo này, chúng tôi so sánh khả năng dự đoán của các mô hình học máy khác nhau và sử dụng phương pháp LIME để diễn giải cách chúng đưa ra quyết định. Dữ liệu được thu thập từ báo cáo tài chính của các công ty niêm yết từ năm 2009 đến năm 2023. Kết quả cho thấy Gradient Boosting và Random Forest đạt hiệu suất tốt nhất.Thêm vào đó, trọng số LIME chỉ ra rằng các yếu tố ảnh hưởng nhiều nhất đến dự đoán của các mô hình là tỷ lệ thanh khoản hiện hành, tỷ suất lợi nhuận trên tài sản, tỷ lệ nợ và tỷ lệ nợ trên vốn chủ sở hữu.

Từ khóa: *Rủi ro tài chính, công ty niêm yết, mô hình học máy, phương pháp LIME.*

Application of machine learning in assessing financial risk of listed companies on the Vietnam stock market

ABSTRACT

Financial risk management is essential for businesses as it helps prevent losses and maximize profits. Since this process depends heavily on data-driven decision-making, machine learning offers a promising avenue for developing innovative methods and technologies. In this paper, we compare the predictive capabilities of various machine learning models and use the LIME method to interpret how they make decisions. Data was collected from the financial statements of listed companies from 2009 to 2023. The results show that Gradient Boosting and Random Forest achieved the best performance. Additionally, LIME weights indicate that the most influential factors affecting the models' predictions are the current ratio, return on assets, debt ratio, and debt-to-equity ratio.

Keywords: *Financial risk, listed companies, machine learning models, LIME method.*

1. INTRODUCTION

Financial risk arises when there is a chance that an event will cause a company to underperform relative to its planned financial targets or established metrics.¹ Examples of such financial metrics or values encompass earnings per share, return on equity, and cash flows. Financial risks encompass categories such as market risk, credit risk, market liquidity risk, operational risk, and legal risk. Financial risk assessment is critical for investors, regulators, and corporate managers to identify potential challenges and mitigate their impacts.

Financial risk is often associated with the risk of bankruptcy or insolvency of a business. Traditional methods of financial risk assessment often rely on expert judgment and statistical models. Experts can leverage their domain knowledge to identify potential risks, assess the impact of external factors, and interpret the results of statistical models. However, expert judgment can be subjective and prone to bias, particularly when dealing with complex financial scenarios. Numerous statistical models have been proposed, such as Z-score, S-score, O-score, X-score, H-score, B-score,...²⁻⁷ In Vietnam, researchers have tested the Z-score model in forecasting corporate failure⁸ and bankruptcy⁹, applied the B-score in analyzing factors influencing financial risk¹⁰, compared various models in measuring financial distress^{11,...} Statistical models are straightforward in design, offer strong explanatory power, and require relatively short training time. These methods,

however, rely on several rigid preconditions that frequently prove to be impractical in real-world situations. These preconditions include, for example, the existence of linear relationships, consistent variance across data, and variable independence. If these preconditions are not met, the effectiveness of these statistical approaches in prediction can be diminished.¹²

In recent years, machine learning (ML) has emerged as a powerful tool for overcoming the limitations of traditional methods. ML algorithms can automatically learn complex patterns from large datasets, without relying on strict assumptions. This makes them well-suited for financial risk assessment, where data is often noisy, incomplete, and high-dimensional. Algorithms such as support vector machine (SVM), decision tree, and artificial neural network are applied to enhance the efficiency of traditional methods in volatility forecasting, bankruptcy prediction, credit scoring,...¹³⁻¹⁶ Ensemble learning and hybrid models have been widely studied in this field. Research suggests that random forest algorithms may surpass other single or hybrid classifiers.^{17,18}

In this article, we will construct and compare the performance of several advanced machine learning models, such as SVM, neural networks, random forests, gradient boosting,...in forecasting the financial risks of listed companies on the Vietnamese stock market. Additionally, we also assess the importance of features using LIME to identify the key factors influencing

financial risk and propose solutions to mitigate these risks.

2. METHODOLOGY

2.1. Data collection and preprocessing

In this study, we utilize data extracted from the financial statements of companies listed on the HOSE (Ho Chi Minh Stock Exchange), HNX (Hanoi Stock Exchange), and UPCOM (Unlisted Public Company Market). The data spans the period from 2009 to 2023 and includes balance sheets, income statements, and cash flow statements.

This study applies machine learning models to predict financial risk, specifically

Table 1. Bankruptcy prediction models for defining the target variable.

Model	Formula	Conclusion
Z-score (1968)	$Z = 1.2Z1 + 1.4Z2 + 3.3Z3 + 0.6Z4 + 1.0Z5$ $Z1 = \text{Working capital} / \text{Total assets}$ $Z2 = \text{Retained earnings} / \text{Total assets}$ $Z3 = \text{EBIT} / \text{Total assets}$ $Z4 = \text{Market value of equity} / \text{Total liabilities}$ $Z5 = \text{Sales} / \text{Total assets}$	$Z < 2.99: y = 1$ $Z \geq 2.99: y = -1$
S-score (1978)	$S = 1.03S1 + 3.07S2 + 0.66S3 + 0.4S4$ $S1 = \text{Working capital} / \text{Total assets}$ $S2 = \text{EBIT} / \text{Total assets}$ $S3 = \text{Profit before tax} / \text{Current liabilities}$ $S4 = \text{Sales} / \text{Total assets}$	$S < 0.862: y = 1$ $S \geq 0.862: y = -1$
X-score (1984)	$X = -4.336 - 4.513X1 + 5.679X2 - 0.004X3$ $X1 = \text{Net income} / \text{Total assets}$ $X2 = \text{Total liabilities} / \text{Total assets}$ $X3 = \text{Current assets} / \text{Current liabilities}$	$X \geq 0: y = 1$ $X < 0: y = -1$
Taffler Z-score (1983)	$T = 3.20 + 12.18T1 + 2.50T2 - 10.68T3 + 0.029T4$ $T1 = \text{Profit before tax} / \text{Current liabilities}$ $T2 = \text{Current assets} / \text{Total liabilities}$ $T3 = \text{Current liabilities} / \text{Total assets}$ $T4 = \text{No-credit interval}$	$T \leq 0.3: y = 1$ $T > 0.3: y = -1$
G-score (2001)	$G = 1.6505G1 + 3.404G2 - 0.016G3 + 0.057$ $G1 = \text{Working capital} / \text{Total assets}$ $G2 = \text{EBIT} / \text{Total assets}$ $G3 = \text{ROA}$	$G \leq 0.01: y = 1$ $G > 0.01: y = -1$

Table 2. Financial ratios (features) for assessing financial risk.

Symbol	Ratio name	Symbol	Ratio name
X1	Price-to-earnings ratio	X18	EV-to-EBIT ratio
X2	Price-to-sale ratio	X19	Price-to-operating-cash-flow ratio
X3	Price-to-book ratio	X20	Debt ratio
X4	Earnings per share	X21	Price-to-cash-flow ratio
X5	Return on equity	X22	Book value per share
X6	Return on assets	X23	Cash ratio

bankruptcy risk, formulated as a classification problem. To identify companies at risk, we utilize five widely recognized bankruptcy prediction models: the Altman Z-score, Springate S-score, Zmijewski X-score, Taffler Z-score, and Grover G-score (Table 1). A company is labeled as 1 (at risk) if the majority of the five models classify it as being at risk, and -1 otherwise. Regarding independent variables, based on several studies, we use 34 financial ratios as inputs for the machine learning models, as presented in Table 2. These ratios reflect various aspects of the company, such as liquidity, profitability, efficiency, and leverage.

X7	Return on invested capital	X24	Return on capital employed
X8	Operating margin	X25	Return on sales
X9	Gross margin	X26	Cash return on invested capital
X10	Net margin	X27	Cash return on equity
X11	EBIT margin	X28	Cash return on assets
X12	Current ratio	X29	Free cash flow margin
X13	Quick ratio	X30	Operating cash flow margin
X14	Debt-to-equity ratio	X31	Total asset turnover ratio
X15	Operating cash flow ratio	X32	Equity ratio
X16	EV-to-EBITDA ratio	X33	Fixed asset turnover ratio
X17	EV-to-sales ratio	X34	Receivables turnover ratio

The dataset consists of 2614 observations, including 557 observations with $y = 1$ and 2057 observations with $y = -1$. Before performing preprocessing steps, the data is split into training and testing sets at an 8:2 ratio to prevent data leakage. Data leakage in machine learning arises when a model, during its training phase, utilizes data that would not be accessible when making actual predictions. This form of leakage creates a deceptive appearance of model accuracy, which is only revealed to be false upon deployment. In practice, such models produce unreliable outcomes, resulting in flawed decision-making and misleading conclusions. The dataset is then cleaned by removing outliers and imputing missing values.

2.2. Dimensionality reduction

Dimensionality reduction involves decreasing the number of features to enable efficient model development. It has two main methods: feature selection and feature extraction. Feature selection chooses the most important original features. Feature extraction makes new features by combining or changing the originals.

Here, we will use the feature selection to retain the original meaning of the variables in the dataset. Our data has numerical attributes, and the target variable is categorical, so we will use the ANOVA F-test technique.¹⁹ ANOVA, or “analysis of variance”, is a parametric test to check if means of two or more samples come from the same distribution. It's an F-test, a type of statistical test that compares variances, like variance across samples or explained versus unexplained variance in ANOVA. This method is

particularly useful when one variable is numerical and the other is categorical, such as numerical input features and a categorical target variable in classification tasks. The results of ANOVA can be applied in feature selection by identifying and removing features that are independent of the target variable, helping to refine the dataset for better model performance.

2.3. Machine learning models to predict financial risk

In this study, we implement and compare the effectiveness of statistical and machine learning models, including Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), Adaptive Boosting (AdaBoost), Gradient Boosting, and Multi-layer Perceptron (MLP).

2.3.1. Logistic Regression

Logistic regression is a widely used statistical method for binary outcome prediction.²⁰ In this study, it is applied to determine financial risk status. The model produces an output P_n , which represents the probability of being at risk based on the input variables \mathbf{X} . This probability is derived using Equation (1).

$$P_n(y = 1|\mathbf{X}) = \frac{1}{1+e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_k x_k)}} \quad (1)$$

Logistic regression often serves as a baseline in studies designed to measure the performance of alternative forecasting approaches. Its primary strength lies in the simplicity and clarity of its results, making them accessible and easy to interpret for most users. This high level of interpretability makes logistic

regression a popular choice in practical applications, particularly within financial institutions.

2.3.2. Support Vector Machine

Support Vector Machine (SVM) is a robust machine learning algorithm designed for both classification and regression tasks.²¹ In this study, it is employed to classify data points into distinct categories based on input features \mathbf{X} . The model constructs an optimal hyperplane that maximizes the margin between support vectors. The classification process will take place according to Equation (2):

$$y = \text{sign}(\mathbf{w}^T \mathbf{x} + b) \quad (2)$$

SVM is particularly effective in handling high-dimensional data and is often combined with kernel functions to address non-linear problems. Its main advantage lies in its ability to generalize well, even with limited data, making it a standard choice in applications like image classification, bioinformatics, and text categorization.

2.3.3. Random Forest

Random Forest is a highly effective ensemble algorithm frequently employed for both classification and regression. This method builds a collection of decision trees during training and then synthesizes their outputs to improve prediction accuracy and limit overfitting.²² Each tree is trained on a random subset of data, and only a random subset of features is considered for splitting at each node, enhancing diversity among the trees. The final prediction is made through majority voting (for classification) or averaging (for regression). Known for its robustness and ability to handle high-dimensional, non-linear data, Random Forest is widely applied in areas like financial risk assessment, medical diagnosis, and image classification.

2.3.4. Adaptive Boosting

Boosting constructs a model on training data, then creates another model to fix the first model's errors. This technique is repeated until errors diminish and data prediction is accurate. Boosting combines multiple weak models into a strong model for the final result.

AdaBoost works by initially assigning equal weights to all samples in the training dataset.²³ The algorithm then iterates for a predefined number of iterations or until a stopping criterion is met. In each iteration, a weak classifier \hat{f}_i (e.g., a one-level decision tree) is

trained on the data. The weights of the samples are updated, giving higher weights to misclassified examples to focus more on them in subsequent iterations. The weak classifiers are evaluated based on their errors, with lower-error classifiers receiving higher weights. The sample weights are then normalized to sum up to 1. The final prediction is made by combining the predictions of all p weak classifiers using a weighted majority vote:

$$\hat{f}(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^p \alpha_i \hat{f}_i(\mathbf{x})\right) \quad (3)$$

This process repeats until the specified number of iterations is completed or the stopping criterion is satisfied.

2.3.5. Gradient Boosting

Gradient Boosting, a powerful boosting algorithm, creates strong learners by combining weak ones. It trains each new model to minimize the previous model's loss—like mean squared error or cross-entropy—using gradient descent. In each step, the algorithm calculates the loss function's gradient against the ensemble's predictions and trains a new weak model to reduce this gradient.²⁴ The ensemble is built incrementally by adding predictions from each new model, a process that continues until a stopping point is reached. Different from AdaBoost which adjusts sample weights, Gradient Boosting focuses on training each new predictor to target the residual errors from the previous one, using these residuals as the learning objective. Gradient Boosted Trees, a well-known implementation, is based on CART (Classification and Regression Trees).

2.3.6. Multi-layer Perceptron (MLP)

Multi-layer Perceptron (MLP) is the most common neural network architecture, composed of input, hidden, and output layers.²⁵ For each neuron in a hidden layer, the operation involves taking a weighted sum of its inputs. This sum is then subjected to a non-linear activation, examples of which include the Rectified Linear Unit (ReLU), Sigmoid, and Hyperbolic Tangent (Tanh).

During training, MLP utilizes a two-step learning process: forward propagation and backpropagation. In forward propagation, the output of a neuron is computed as follows:

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)} \quad (4)$$

$$\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)}) \quad (5)$$

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the weight matrix and bias vector for layer l , $\mathbf{x}^{(l-1)}$ is the input from the previous layer, and $f(\cdot)$ is the activation function. The backpropagation algorithm then updates the network's weights by computing gradients of the loss function with respect to the weights using the chain rule. The gradient descent optimization technique, often with variations such as Stochastic Gradient Descent (SGD) or Adam, is applied to minimize the loss iteratively.

MLP is widely used in classification and regression tasks due to its ability to learn complex patterns in data. It serves as a foundation for more advanced deep learning models and is particularly effective in applications such as image recognition, speech processing, and time series prediction.

2.3. Local Interpretable Model-agnostic Explanations (LIME)

Local Interpretable Model-agnostic Explanations (LIME) is an algorithmic approach designed to elucidate the predictions of any classifier or regressor. It achieves this by creating a locally faithful approximation using an interpretable model.²⁶

Often classified as a “surrogate model” approach, the LIME explainability model is constructed through a step-by-step procedure. First, to generate a substitute dataset, the LIME algorithm subtly alters the feature values of the original dataset – the very data that trained the black-box model. Next, these newly created samples are assigned weights that reflect their similarity to the particular instance under explanation. Lastly, an inherently understandable model, like a decision tree or logistic regression, is employed as a surrogate machine learning model and trained on this weighted, artificially created dataset. The learned model should be a good approximation of the machine learning model predictions locally, but it does not have to be a good global approximation. This kind of accuracy is also called local fidelity. The explanation produced by LIME is obtained by the following:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}(f, g, \pi_x) + \Omega(g) \quad (6)$$

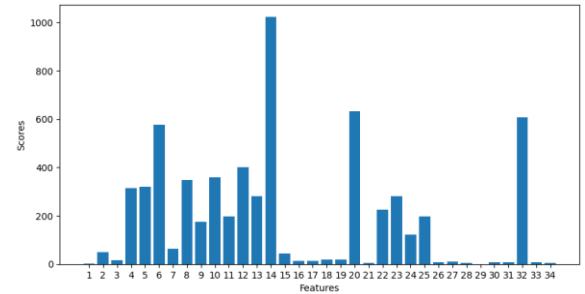
LIME explains instance x with a model g (like linear regression) that minimizes a loss \mathcal{L} (e.g., mean squared error). Loss \mathcal{L} measures how well g mimics the original model f (e.g., AdaBoost) predictions, while keeping g simple (low $\Omega(g)$).

G is the set of possible g models, for example, all linear regressions. π_x defines the neighborhood size around x used for explanation.

3. RESULTS AND DISCUSSION

3.1. Dimensionality reduction results

Figure 1 illustrates the F-statistics of 34 features when performing ANOVA. X14 (debt-to-equity ratio) appears to be the most relevant, and 16 out of 34 features have significantly higher scores



compared to the rest. We will retain these 16 features and remove the other 18 to proceed with building machine learning model.

Figure 1. ANOVA F-test result.

3.2. Performance of machine learning models

When training a machine learning model, we fit the model's parameters. However, before the model begins learning, certain parameters are pre-set—these are called hyperparameters. We can improve the model's performance by tuning these hyperparameters. There are several hyperparameter tuning methods, such as grid search, random search, and Bayesian optimization. Among them, grid search is widely used. Grid search works by building a grid of all hyperparameter settings. Then, it trains and tests the model with each setting and picks the best one. This complete search of hyperparameters makes sure every option is checked. Moreover, grid search is typically used with cross-validation, specifically k-fold cross-validation. Here, the training set is divided into k parts. In each iteration, $k - 1$ parts are used to train the model, while the remaining part is used for validation. The best set of hyperparameters is the one that yields the highest average performance. Finally, the models with the optimal set of hyperparameters are tested on the test set using various metrics.

Table 3 presents the hyper-parameter settings and the evaluation of the models on different metric

Table 3. Performance of the models on test set.

Models	Hyper-parameter settings	Accuracy	Precision	Recall	F1 Score	AUC
Logistic Regression	C=1, max_iter=300, penalty='1', solver= 'saga'	0.9331	0.8861	0.8952	0.8901	0.9693
SVM	C=1, degree=2, gamma='scale', kernel='rbf'	0.8642	0.7799	0.8729	0.8097	0.9534
Random Forest	bootstrap=False, max_depth=10, max_features='sqrt', min_samples_split=20, n_estimators=100	0.9484	0.9133	0.9166	0.9149	0.9836
AdaBoost	learning_rate=1, n_estimators=500	0.9331	0.8904	0.8873	0.8888	0.9780
Gradient Boosting	learning_rate=0.5, loss='log_loss', max_depth=7, max_features='sqrt', min_samples_split=20, n_estimators=100	0.9579	0.9276	0.9344	0.9309	0.9870
MLP	activation='relu', alpha=0.01, hidden_layer_sizes=(100,), learning_rate='adaptive', solver='adam'	0.9312	0.8785	0.9020	0.8896	0.9788

Gradient Boosting achieved the best performance across all metrics, indicating high predictive accuracy and a good balance between precision and recall. Random Forest ranked second with high accuracy and AUC, demonstrating strong and consistent classification ability. MLP also showed good results across all metrics, particularly in AUC.

AdaBoost and Logistic Regression had similar performance with accuracy but showed lower precision and recall compared to Gradient Boosting and Random Forest. SVM had the lowest performance across all metrics, particularly in precision and F1 score, indicating difficulties in accurate classification and balancing precision and recall.

Overall, Gradient Boosting is the most suitable model for this problem, followed by Random Forest and MLP, while SVM performed the worst.

3.3. Interpretations of results

We used LIME to interpret the two best-performing models: Gradient Boosting and Random Forest. A random instance from the test set was selected to generate a local explanation for this specific instance (Figure 2).

The chosen instance has a true label of $y = -1$, indicating no risk. Both models identified features X12 and X24 as the most influential. Specifically, X12 contributes to the model's prediction of $y = -1$, while X24 influences the prediction in the opposite direction. For Gradient Boosting, the impact of features decreases noticeably from top to bottom, highlighting the model's tendency to focus on the most important features. In contrast, Random Forest distributes influence more evenly across features, reflecting its nature of aggregating predictions from multiple independent decision trees.

Local explanations are valuable for understanding the reasoning behind individual predictions. However, analyzing a single instance does not provide a comprehensive understanding of the model's overall behavior. To gain deeper insights into the model's decision-making process, we can aggregate local explanations across multiple predictions. Specifically, by combining the LIME weights of numerous instances and visualizing them through various types of charts, we can better capture the model's general patterns and feature importance.

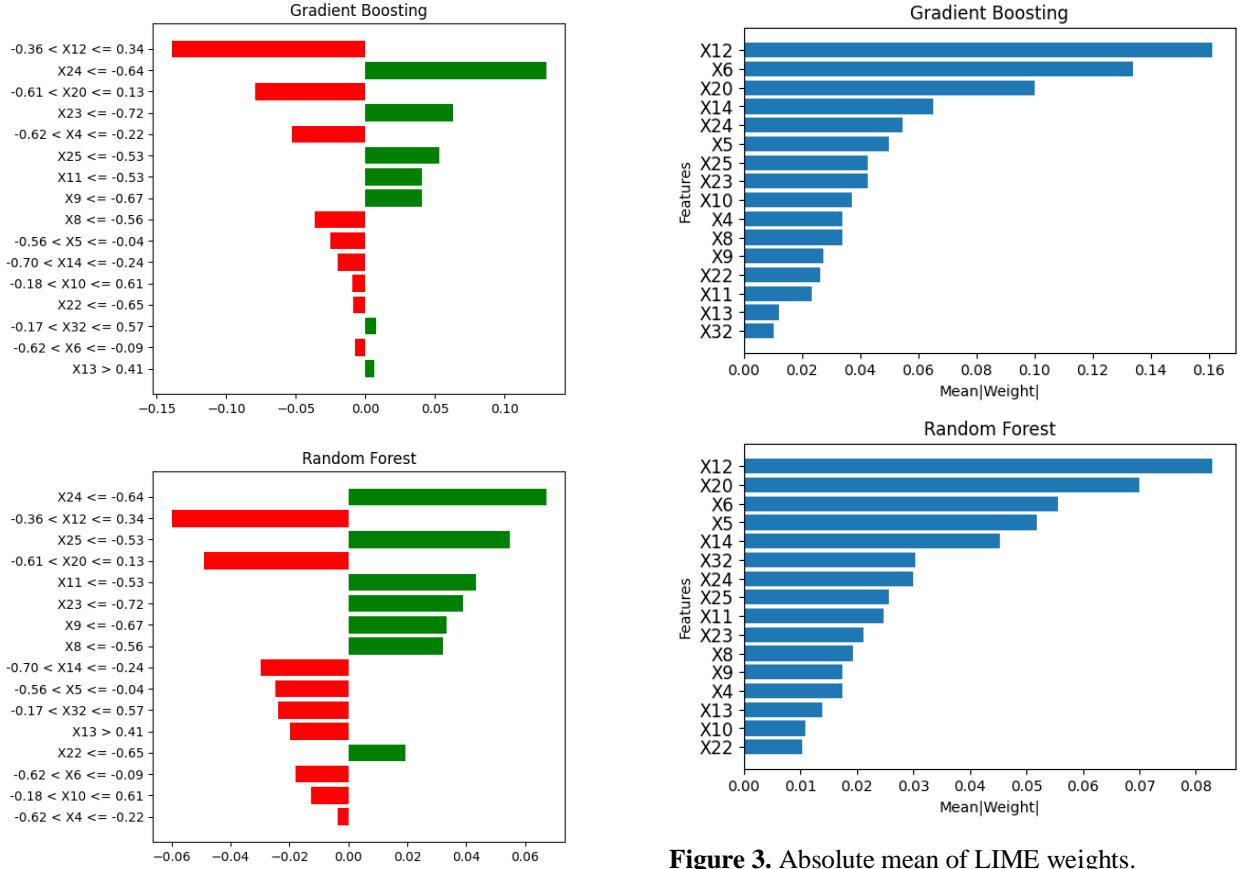


Figure 2. Local explanations of Gradient Boosting and Random Forest.

The first aggregation can help us understand which of the features are most important. Features with either high positive or negative LIME weights had a larger impact on a prediction. For each feature, we take the absolute mean of all the LIME weights. Features with large mean weights have, in general, made large contributions to the predictions. Figure 3 shows the average weights of the features in the two models. It can be observed that the important features are relatively similar across both models. These features are the current ratio (X12), return on assets (X6), debt ratio (X20), and debt-to-equity ratio (X14).

Next, we will examine the trends of these important features by plotting the LIME weights against their values. A high LIME weight indicates that, for a specific prediction, the value of this feature increases the probability of predicting a “At risk” case ($y = 1$).

Figure 3. Absolute mean of LIME weights.

Figure 4 illustrates that as the values of X12 and X6 rise, their LIME weights shift from positive to negative. X12, representing the current ratio, assesses a company's short-term liquidity. A low current ratio suggests potential liquidity problems, which increase financial risk and result in a positive LIME weight. In contrast, a high current ratio indicates a stronger ability to meet debt obligations, reducing financial risk and producing a negative LIME weight. This negative weight decreases the probability of being classified as risky ($y = 1$). Meanwhile, X6, which measures return on assets (ROA), reflects how efficiently a company generates profit from its assets. A low ROA indicates weak profitability and higher financial risk, leading to a positive LIME weight. Conversely, a high ROA signifies effective asset management and lower risk, resulting in a negative LIME weight.

On the other hand, the LIME weights for X20 and X14 increase as their values grow. X20, the debt ratio, indicates the proportion of a company's assets financed through debt. A high debt ratio suggests significant reliance on borrowed funds, which raises financial leverage and risk due to fixed interest obligations. Similarly, X14, the debt-to-equity ratio, compares total debt to shareholders' equity. A high value for X14 indicates a greater

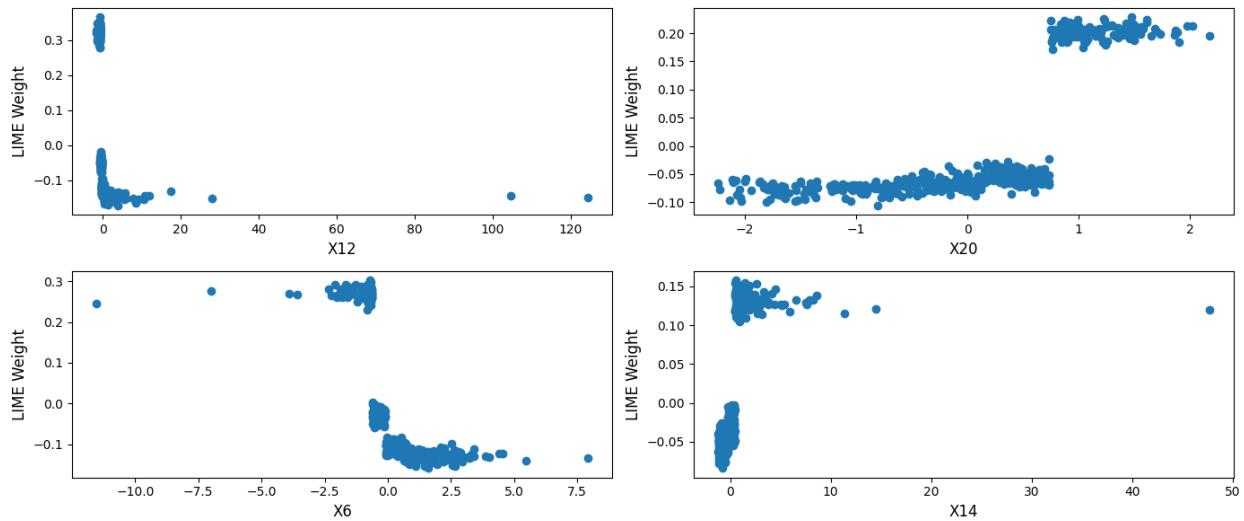


Figure 4. Feature trends for the four most important features.

dependence on debt compared to equity, leading to increased financial burden and risk.

4. CONCLUSIONS

In this study, we developed and compared advanced machine learning models to predict the financial risk of companies listed on the Vietnamese stock market. Based on financial ratios, various models were constructed, hyperparameters were optimized, and evaluations were conducted using different metrics. The two best-performing models were Gradient Boosting and Random Forest, achieving over 94% accuracy and more than 91% recall. This demonstrates the superiority of ensemble learning methods over single models. Furthermore, the LIME method was utilized to explain the models' predictions and the influence of different features on their decisions. The results indicate that to reduce financial risk, businesses should improve their current ratio (X12) by efficiently managing inventory and accelerating receivables collection, thereby reducing the likelihood of liquidity issues. Additionally, enhancing return on assets (ROA - X6) through optimized production processes can lower financial risk. Companies should also closely monitor the debt ratio (X20) and debt-to-equity ratio (X14) by avoiding excessive borrowing and increasing equity financing to reduce interest burdens. Moreover, diversifying funding sources by balancing debt and equity financing will optimize the capital structure and minimize financial risk in the long term. The findings of this study provide a foundation for businesses to manage risks more effectively, make safer business decisions, and optimize their strategies.

REFERENCES

1. P. P. Drake & F. J. Fabozzi. *The basics of finance: An introduction to financial markets, business finance, and portfolio management* (Vol. 192), John Wiley & Sons, New Jersey, 2010.
2. E. I. Altman. Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy, *The Journal of Finance*, **1968**, 23(4), 589–609.
3. G. L. Springate. *Predicting the possibility of failure in a Canadian firm: A discriminant analysis*, Doctoral dissertation, Simon Fraser University, 1978.
4. J. A. Ohlson. Financial Ratios and The Probabilistic Prediction of Bankruptcy, *Journal of Accounting Research*, **1980**, 18(1), 109-131.
5. M. E. Zmijewski. Methodological issues related to the estimation of financial distress prediction models, *Journal of Accounting research*, **1984**, 22, 59-82.
6. J. G. Fulmer, J. E. Moon, T. A. Gavin & M. Erwin. A bankruptcy classification model for small firms, *Journal of commercial bank lending*, **1984**, 66(11), 25-37.
7. A. Bathory. *Predicting corporate collapse: Credit Analysis in the determination and forecasting of insolvent companies*, Financial Times Business Information, 1984.
8. Lê Cao Hoàng Anh, Nguyễn Thu Hằng. Kiểm định mô hình chỉ số Z của Altman trong dự báo thất bại doanh nghiệp tại Việt Nam, *Tạp chí Công nghệ ngân hàng*, **2012**, 74, 3-9.
9. Hoàng Thị Hồng Vân. Vận dụng mô hình Z-score trong dự báo khả năng phá sản doanh nghiệp tại Việt Nam, *Tạp chí Khoa học & Đào tạo Ngân hàng*, **2020**, 217, 43-51.
10. Vũ Thị Hậu. Ứng dụng mô hình B-score phân tích các yếu tố ảnh hưởng đến rủi ro tài chính tại các doanh nghiệp y tế niêm yết trên thị trường chứng khoán Việt Nam, *Tạp chí Khoa học Đại học mở Thành phố Hồ Chí Minh-Kinh tế và Quản trị Kinh doanh*, **2023**, 18(3), 135-146.

11. Lê Hoàng Vinh, Phạm Lê Quang, Bùi Kim Dung. Mô hình nào phù hợp để đo lường kiệt quệ tài chính cho công ty phi tài chính niêm yết tại Việt Nam?, *Tạp chí Quản lý và Kinh tế quốc tế*, 2022, 144, 39-52.
12. Tran Kim Long, Le Hoang Anh, Nguyen Thanh Hien & Nguyen Duc Trung. Explainable machine learning for financial distress prediction: Evidence from Vietnam, *Data*, 2022, 7(11), 160.
13. W. Kristjanpoller, A. Fadic & M. C. Minutolo. Volatility forecast using hybrid neural network models, *Expert Systems with Applications*, 2014, 41(5), 2437-2442.
14. M. Y. Chen. Bankruptcy prediction in firms with statistical and intelligent techniques and a comparison of evolutionary computation approaches, *Computers & Mathematics with Applications*, 2011, 62(12), 4514-4524.
15. L. Yu, W. Yue, S. Wang & K. K. Lai(2010). Support vector machine based multiagent ensemble learning for credit risk evaluation, *Expert Systems with Applications*, 2010, 37(2), 1351-1360.
16. H. S. Kim & S. Y. Sohn. Support vector machines for default prediction of SMEs based on technology credit, *European Journal of Operational Research*, 2010, 201(3), 838-846.
17. J. Kruppa, A. Schwarz, G. Arminger & A. Ziegler. Consumer credit risk: Individual probability estimates using machine learning, *Expert systems with applications*, 2013, 40(13), 5125-5131.
18. M Malekipirbazari & V. Aksakalli. Risk assessment in social lending via random forests, *Expert Systems with Applications*, 2015, 42(10), 4621-4631.
19. N. O. F. Elssied, O. Ibrahim & A. H. Osman. A novel feature selection based on one-way anova f-test for e-mail spam classification, *Research Journal of Applied Sciences, Engineering and Technology*, 2014, 7(3), 625-638.
20. D. Kleinbaum, M. Klein. *Logistic Regression: A Self-Learning Text*, Springer, 2010.
21. L. Wang (Ed.). *Support vector machines: theory and applications*, Springer Science & Business Media, 2009.
22. L. Breiman. Random Forests, *Machine Learning*, 2001, 45, 5-32.
23. Y. Freund, R. R. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997, 55(1), 119-139.
24. J. H. Friedman. Greedy function approximation: A gradient boosting machine, *The Annals of Statistics*, 2001, 29(5) 1189-1232.
25. M. C. Popescu, V. E. Balas, L. Perescu-Popescu & N. Mastorakis. Multilayer perceptron and neural networks, *WSEAS Transactions on Circuits and Systems*, 2009, 8(7), 579-588.
26. M. T. Ribeiro, S. Singh & C. Guestrin. "Why should i trust you?": Explaining the predictions of any classifier, *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, 1135-1144.