TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC QUY NHƠN

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

Hà Thanh Bắc*, Cao Tấn Bình
Nguyễn Lê Hữu Đang, Nguyễn Hồ Bảo Thiên, Nguyễn Ngọc Việt Ý

Khoa Toán và Thống kê, Trường Đại học Quy Nhơn, Việt Nam

Ngày nhận bài: 14/03/2025; Ngày sửa bài: 11/04/2025;
Ngày nhận đăng: 12/04/2025; Ngày xuất bản: 28/08/2025

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.*

**Tác giả liên hệ chính.*
Email: bachathanh1@gmail.com

QUY NHON UNIVERSITY
JOURNAL OF SCIENCE

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

Ha Thanh Bac*, Cao Tan Binh
Nguyen Le Huu Dang, Nguyen Ho Bao Thien, Nguyen Ngoc Viet Y

Faculty of Mathematics and Statistics, Quy Nhon University, Vietnam

Received: 14/03/2025; Revised: 11/04/2025;
Accepted: 12/04/2025; Published: 28/08/2025

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 failure[8] and bankruptcy,[9] applied the B-score in analyzing factors influencing financial risk,[10] compared various models in measuring financial distress,[11]… Statistical models are straightforward in design, offer strong explanatory power, and require

**Corresponding author.*
Email: bachathanh1@gmail.com

https://doi.org/10.52111/qnjs.2025.19401

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.[17] Research suggests that random forest algorithms may surpass other single or hybrid classifiers.[18-21]

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 200 companies listed on the HOSE (Ho Chi Minh Stock Exchange), HNX (Hanoi Stock Exchange), and UPCOM (Unlisted Public Company Market). The dataset covers the period from 2009 to 2023 and includes balance sheets, income statements, and cash flow statements. While some companies have incomplete data for the full 15-year period, each has at least 8 years of available records.

This study applies machine learning models to predict financial risk, specifically 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). In the Z-score and Taffler Z-score models, predictions may fall into a gray area indicating uncertainty. To improve recall for identifying at-risk cases and ensure consistency with other models, we classify observations in this gray area as at-risk ($y = 1$). As a result, the decision rules differ slightly from those in the original models. 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.

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 = Working capital / Total assets Z2 = Retained earnings / Total assets Z3 = EBIT / Total assets Z4 = Market value of equity / Total liabilities Z5 = Sales / 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 = Working capital / Total assets S2 = EBIT / Total assets S3 = Profit before tax / Current liabilities S4 = Sales / Total assets	S < 0.862: $y = 1$ S ≥ 0.862: $y = -1$
X-score (1984)	X = -4.336 – 4.513X1 + 5.679X2 – 0.004X3 X1 = Net income / Total assets X2 = Total liabilities / Total assets X3 = Current assets / Current liabilites	X ≥ 0: $y = 1$ X < 0: $y = -1$
Taffler Z-score (1983)	T = 3.20 + 12.18T1 + 2.50T2 – 10.68T3 + 0.029T4 T1 = Profit before tax / Current liabilities T2 = Current assets / Total liabilities T3 = Current liabilities / Total assets T4 = No-credit interval	T ≤ 0.3: $y = 1$ T > 0.3: $y = -1$
G-score (2001)	G = 1.6505G1 + 3.404G2 – 0.016G3 + 0.057 G1 = Working capital / Total asssets G2 = EBIT / Total assets G3 = ROA	G ≤ 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
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

https://doi.org/10.52111/qnjs.2025.19401

The dataset consists of 2774 observations, including 598 observations with $y = 1$ and 2176 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.

Table 3 presents the descriptive statistics of the independent variables used in this study. All attributes contain missing data. Some variables have negative values, even though they are theoretically not supposed to, indicating potential errors in the input data due to manual entry mistakes or measurement inaccuracies. In addition, several variables such as X1, X4, X16, and X21 exhibit standard deviations that are significantly higher than their means, suggesting considerable variation across observations. The large differences between the mean and median of certain variables (e.g., X1) indicate skewed distributions, while extremely high maximum values in variables such as X4 and X22 suggest the presence of outliers.

Table 3. Descriptive statistics of independent variables.

Variable	Count	Mean	Standard deviation	Min	Median	Max
X1	2611	49.48	676.04	-21296.64	11.63	20255.87
X2	2605	18.82	396.28	-100.67	0.87	19022.45
X3	2611	1.58	1.91	-10.48	1.10	25.04
X4	2611	1952.95	3910.52	-9363.37	1104.73	144517.65
X5	2629	0.11	0.30	-7.50	0.09	5.23
X6	2629	0.05	0.08	-0.99	0.04	0.84
X7	2537	0.02	0.07	-2.80	0.01	0.45
X8	2745	0.31	9.84	-65.81	0.08	498.81
X9	2612	0.21	0.44	-12.04	0.18	3.18
X10	2612	0.21	6.82	-74.13	0.07	323.09
X11	2612	0.02	1.57	-69.81	0.08	10.30
X12	2633	2.69	7.71	0.06	1.42	136.47
X13	2633	1.75	7.39	0.01	0.80	136.21
X14	2633	1.81	5.53	-31.06	1.14	162.31
X15	2751	0.20	2.30	-62.15	0.11	162.31
X16	2611	-200.51	10827.27	-552344.73	8.71	8988.12
X17	2579	18.84	385.70	-2359.82	1.30	18252.28
X18	2594	-8.48	1708.51	-81777.76	11.56	15055.58
X19	2600	-36.90	1305.46	-59515.61	3.37	1860.59
X20	2769	0.51	0.22	0.00	0.53	1.29

X21	2402	-45.62	3515.88	-78359.49	5.56	43561.17
X22	2575	16910.34	9113.65	-7688.75	14486.85	190111.03
X23	2769	0.57	4.19	0.00	0.16	133.47
X24	2764	0.10	0.20	-7.16	0.07	1.94
X25	2745	0.02	1.54	-69.81	0.03	10.30
X26	2751	-0.09	4.37	-229.29	-0.08	1.42
X27	2751	-0.02	1.13	-13.78	-0.17	49.92
X28	2751	-0.01	0.14	-1.13	-0.07	0.89
X29	2730	-0.39	21.07	-464.72	-0.16	907.89
X30	2730	0.08	23.12	-367.10	-0.03	1101.74
X31	2568	0.76	0.73	0.00	0.26	6.69
X32	2769	0.49	0.22	-0.29	0.32	0.99
X33	2564	31.16	664.32	-84.94	1.14	32540.83
X34	2568	7.70	20.43	-0.03	1.55	674.56

Therefore, the dataset needs to be preprocessed through several steps: removing observations with excessive missing values, handling outliers using the IQR method, imputing the remaining missing values using the k-Nearest Neighbors technique with $k = 5$, and standardizing the variables so that they have a mean of 0 and a standard deviation of 1.

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.[22] 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). These models were selected based on their widespread application in classification problems, particularly in the context of financial risk assessment. Logistic Regression serves as a strong baseline due to its simplicity and interpretability. SVM is effective for high-dimensional data. Ensemble models such as Random Forest, AdaBoost, and Gradient Boosting are known for their robustness and ability to handle complex feature interactions.

https://doi.org/10.52111/qnjs.2025.19401

Meanwhile, MLP, a type of neural network, is included to explore the potential of deep learning in capturing nonlinear patterns in the data.

2.3.1. Logistic regression

Logistic regression is a widely used statistical method for binary outcome prediction.[23] 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 $\boldsymbol{X}$. This probability is derived using Equation (1).

$$P_n(y = 1|\boldsymbol{X}) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\cdots+\beta_k X_k)}} \quad (1)$$

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.

The loss function for logistic regression algorithm which is called log-loss (cross-entropy loss), is represented as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ji}\log(a_{ji}) \quad (2)$$

where N is the size of training set, C is the number of classes in the problem, y_{ji} is actual one-hot label of ith sample and a_{ji} is predicted probability for class j of ith sample. This loss function penalizes wrong classifications more heavily when the model is confident but incorrect, which makes it highly effective for probabilistic interpretation. As a convex function, log-loss ensures a single global minimum, enhancing the stability and convergence of gradient-based optimization methods. During training, as the model is updated over epochs, the loss typically decreases smoothly, especially when a suitable learning rate is chosen. The loss function's stability makes logistic regression robust against small fluctuations in the data loop and batch size.

2.3.2. Support Vector Machine

Support Vector Machine (SVM) is a robust machine learning algorithm designed for both classification and regression tasks.[24] In this study, it is employed to classify data points into distinct categories based on input features $\boldsymbol{X}$. The model constructs an optimal hyperplane that maximizes the margin between support vectors. The classification process will take place according to Equation (3):

$$y = \text{sign}(\boldsymbol{w}^T\boldsymbol{x} + b) \quad (3)$$

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.

In the case of SVM algorithm, the loss function is shown as:

$$L = \sum_{i=1}^{N}\max\big(0,1 - y_i(w^T x_i + b)\big) \quad (4)$$

where N is the size of training set, y_i is actual label of ith sample and $w^T x_i + b$ is predicted value of ith sample. The hinge loss used in SVM is not differentiable at the margin boundary but remains convex, which guarantees the existence of a global minimum. Unlike probabilistic models, SVM does not output probabilities but focuses on maximizing the margin. This can lead to more stable generalization, especially when the dataset is not noisy. However, if the data is not linearly separable or if the margin is narrow, the loss may plateau early, requiring careful tuning of hyperparameters such as C (penalty term) to ensure effective convergence.

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.[25] 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.

The loss function used by the Random Forest algorithm, known as "Gini Impurity", is presented below:

$$Gini(S) = 1 - \sum_{c=1}^{C} {p_c}^2 \quad (5)$$

where S is training set at the current node, C is the number of classes in the classification problem and p_c is the probability of class c at the current node. Gini Impurity is used as a criterion for splitting nodes in decision trees, rather than being minimized through a global loss function. Therefore, it does not operate over epochs or follow a traditional gradient-based optimization loop. The model's stability derives from aggregating over many uncorrelated trees rather than minimizing a differentiable loss. This results in lower variance and a reduction in overfitting, making it inherently stable during training.

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.[26] 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:

$$\hat{f}(\boldsymbol{x}) = \text{sign}\left(\sum_{i=1}^{p} \alpha_i \hat{f}_i(\boldsymbol{x})\right) \quad (6)$$

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

In the following, the loss function for AdaBoost is illustrated.

$$L = \sum_{i=1}^{N} \exp\left(-\frac{1}{2} y_i \sum_{j=1}^{p} \alpha_j \hat{f}_j(x_i)\right) \quad (7)$$

where N is the size of training set, p is the number of weak classifiers, y_i is actual label of ith sample, α_j is weight of jth weak classifier and $\hat{fi}(x_i)$ is predicted value made by jth weak classifier for ith sample. The exponential loss in AdaBoost increases rapidly for misclassified samples, which causes the model to focus on hard-to-classify examples. While this often improves performance, it also introduces instability—particularly when the dataset contains noise or outliers, as the loss may disproportionately prioritize these instances. AdaBoost typically does not use epochs in the traditional sense but follows a fixed number of boosting rounds. During each iteration, the loss function drives reweighting of samples, and convergence depends heavily on the number of weak learners and their diversity.

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

https://doi.org/10.52111/qnjs.2025.19401

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.[27] 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).

Gradient Boosting uses the same loss function as logistic regression (log-loss – (2)). In gradient boosting, the loss function not only guides the performance of each individual learner but also influences how residuals are calculated and targeted in subsequent rounds. Although log-loss provides stability due to its convex nature, gradient boosting can become unstable if learning rates are too high or if too many trees are added, leading to overfitting. Therefore, the number of boosting rounds, learning rate, and tree depth must be carefully balanced to maintain convergence and loss stability over iterations.

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.[28] 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:

$$\boldsymbol{z}^{(l)} = \boldsymbol{W}^{(l)}\boldsymbol{x}^{(l-1)} + \boldsymbol{b}^{(l)} \quad (8)$$

$$\boldsymbol{a}^{(l)} = f\left(\boldsymbol{z}^{(l)}\right) \quad (9)$$

where $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$ are the weight matrix and bias vector for layer l, $\boldsymbol{x}^{(l-1)}$ is the input from the previous layer, and $f(.)$ 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.

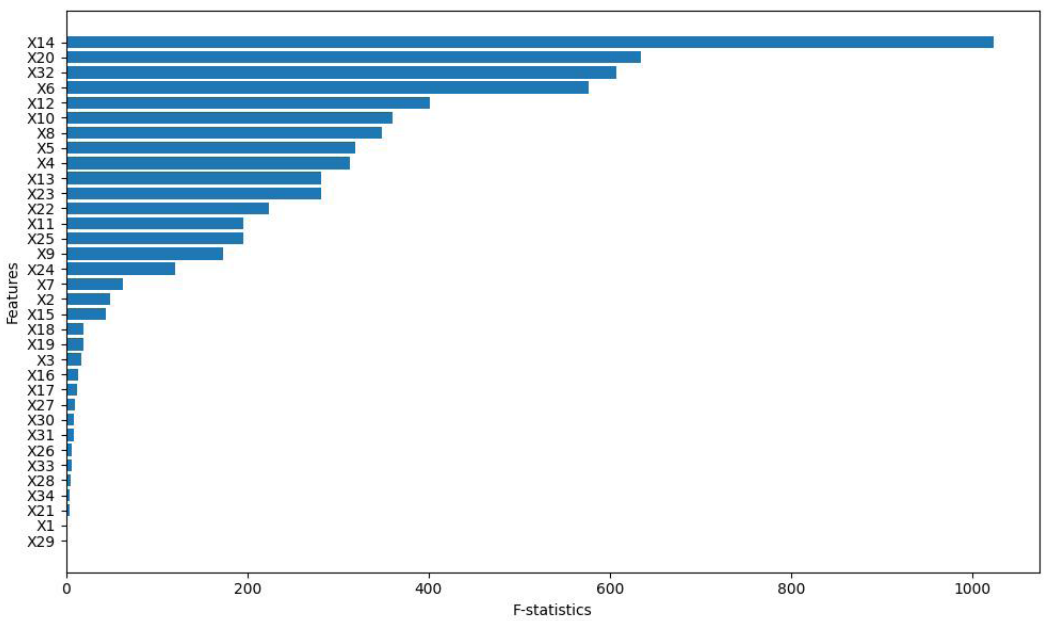

Figure 1. F-statistics of 34 features in descending order.

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.

In the same manner as logistic regression, MLP makes use of log-loss (2) as its loss function. Although MLP uses the same loss function, training stability can be affected by several factors such as learning rate, weight initialization, and batch size. Log-loss contributes to smooth gradient signals during backpropagation, especially when combined with softmax activation in the output layer. However, as the network depth increases, loss landscapes may become non-convex, leading to potential local minima or saddle points. Techniques such as learning rate decay, batch normalization, and early stopping are often employed to stabilize training and ensure the loss consistently decreases across epochs.

2.4. 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.[29]

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:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}(f, g, \pi_x) + \Omega(g) \quad (10)$$

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 and multicollinearity check

Figure 1 shows the descending F-statistics of 34 features obtained through 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 top 16 features and remove the other 18.

Before feeding these 16 features into the models, we calculated the Variance Inflation Factor (VIF), which indicates the presence of multicollinearity in the model. A VIF value below 10 is considered acceptable. Table 4 shows that there is no multicollinearity, as all 16 selected features have VIF values below 10.

Table 4. VIF values of the selected features.

Feature	VIF	Feature	VIF
X4	9.21	X13	3.06
X5	6.57	X14	4.04
X6	9.45	X20	9.33
X8	8.09	X22	2.13
X9	2.65	X23	1.73
X10	8.74	X24	4.17
X11	6.34	X25	6.34
X12	3.16	X32	9.15

https://doi.org/10.52111/qnjs.2025.19401

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 5 presents the hyper-parameter settings and the evaluation of the models on different metric.

Table 5. Performance of the models on test set.

Models	Hyperparameter 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.

Gradient Boosting performs best in this financial risk prediction task due to several key advantages. First, Gradient Boosting effectively captures complex, non-linear relationships and interactions among financial ratios, which are common in real-world financial data. Second, it automatically emphasizes important features while minimizing the impact of irrelevant or noisy ones, which is crucial given the large number of input variables. Third, it handles

class imbalance effectively, which is particularly relevant in our dataset where the class $y = -1$ accounts for nearly 80% of the observations. The model can be adapted using customized loss functions or sample weighting strategies to better learn the characteristics of the minority class and improve predictive performance for underrepresented cases.

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, 3).

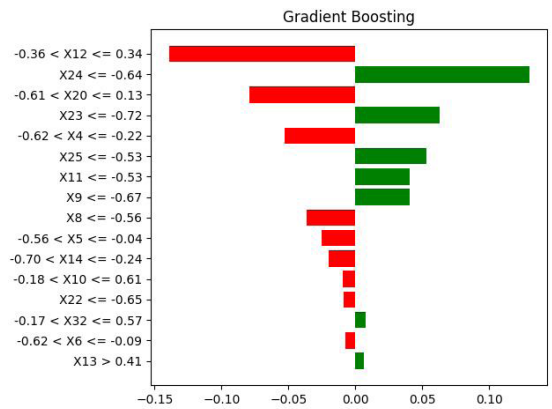

Figure 2. A local explaination of Gradient Boosting.

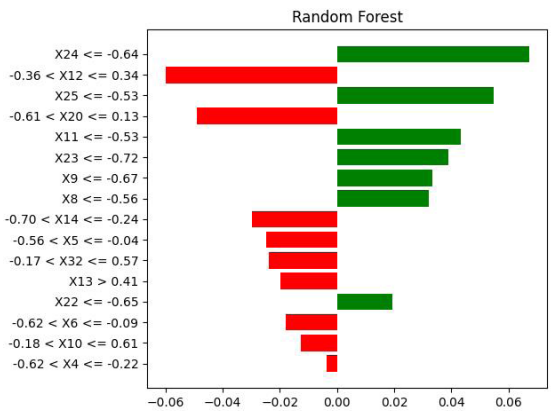

Figure 3. A local explaination of Random Forest.

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 $y = -1$, as indicated by its negative weight (shown in red), whereas X24 supports the prediction of the opposite class with a positive weight (shown in green). 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.

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 4 and Figure 5 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).

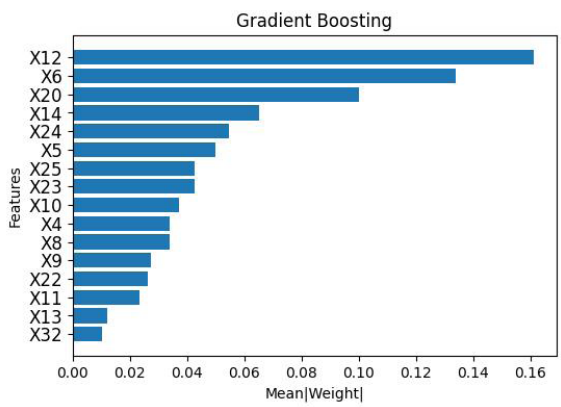

Figure 4. Absolute mean of LIME weights of features in Gradient Boosting.

https://doi.org/10.52111/qnjs.2025.19401

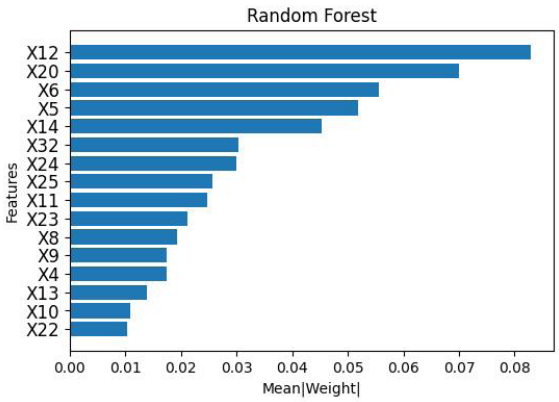

Figure 5. Absolute mean of LIME weights of features in Random Forest.

Next, we analyze how the values of key features affect the model's predictions by plotting their corresponding LIME weights (Figure 6). A higher LIME weight suggests that the feature contributes more strongly to predicting a case as "At risk" (y = 1). Figure 6 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 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

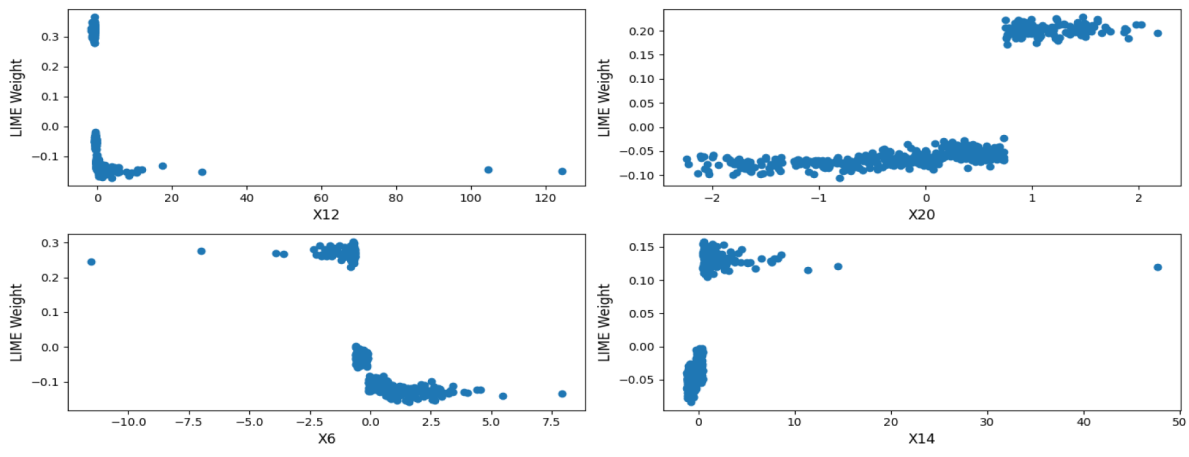

Figure 6. Feature trends for the four most important features.

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.

Acknowledgments

This research is conducted within the framework of the student scientific research project for the academic year 2024 - 2025 under the project code S2024.986.48.

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, England, 1984.
8. L. C. H. Anh, N. T. 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. H. T. H. 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. V. T. 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. L. H. Vinh, P. L. Quang, B. K. 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. T. K. Long, L. H. Anh, N. T. Hien, N. D. 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. 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. X. P. Song, Z. H. Hu, J. G. Du, Z. H. Sheng. Application of machine learning methods to risk assessment of financial statement fraud: evidence from China, *Journal of Forecasting*, **2014**, *33*(8), 611-626.

18. G. T. T. Huyền, N. M. T. Tiên, N. T. Hồng, H. T. N. Ánh, N. T. Thanh, Đ. V. Thắng, Đ. T. V. Trang, L. Đ. Minh. Ứng dụng phương pháp học máy dự báo mức độ căng thẳng tài chính của doanh nghiệp niêm yết tại Việt Nam, *Tạp chí Kinh tế - Luật và Ngân hàng*, **2024**, *270*(11), 90-104.

19. Y. Wang, Y. Zhang, Y. Lu, X. Yu. A comparative assessment of credit risk model based on machine learning-a case study of bank loan data, *Procedia Computer Science*, **2020**, *174*, 141-149.

20. 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.

21. M. Malekipirbazari, V. Aksakalli. Risk assessment in social lending via random forests, *Expert Systems with Applications*, **2015**, *42*(10), 4621-4631.

22. 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.

23. D. Kleinbaum, M. Klein. *Logistic regression: a self-learning text*, Springer, New York, 2010.

24. L. Wang (Ed.). *Support vector machines: theory and applications*, Springer Science & Business Media, Berlin, 2009.

25. L. Breiman. Random forests, *Machine Learning*, **2001**, *45*, 5–32.

26. 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.

27. J. H. Friedman. Greedy function approximation: a gradient boosting machine, *The Annals of Statistics*, **2001**, *29*(5), 1189-1232.

28. M. C. Popescu, V. E. Balas, L. P. Popescu, N. Mastorakis. Multilayer perceptron and neural networks, *WSEAS Transactions on Circuits and Systems*, **2009**, *8*(7), 579-588.

29. 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, California, USA, 2016.

© 2025 by the authors. This Open Access Article is licensed under the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0) license (https://creativecommons.org/licenses/by-nc/4.0/).