Original Research Article

PREDICTION OF FINANCIAL DISTRESS USING DYNAMIC ARTIFICIAL NEURAL NETWORK FOR EARLY WARNING SYSTEM

Abstract

In Kenya's economic landscape, financial hardship is a growing concern, leading to the closure of organizations as they are unable to meet their financial obligations and expectations. This study presents a financial distress prediction dynamic model using Artificial Neural Networks (ANN) to address this issue. By analyzing financial ratios, the ANN model achieved a 94% accuracy, with strong recall, precision, and an ROC-AUC score of 0.99, demonstrating its effectiveness in distinguishing between distressed and non-distressed instances. The model's balanced F1 score of 87% further highlights its value as an Early Warning System (EWS) for financial management, helping organizations make informed decisions and avoid financial crises.

Keywords: *Artificial neural networks, financial ratios, Early warning Systems, financial distress.*

Introduction:

In today's economic environment, the responsibility for assessing a company's financial health increasingly falls on external parties like investors, creditors, auditors, and regulators, who bear significant risks in financial crises. In Kenya, Microfinance Institutions (MFIs) play a crucial role, contributing approximately 7% to the economy according to the Central Bank of Kenya (CBK, 2023). Despite their success in promoting financial inclusion, MFIs face challenges such as poor management, competition, and unfavorable business conditions, which can lead to financial distress—a situation where a firm cannot meet its obligations, potentially leading to bankruptcy (Quadir &Jahur, 2012; Parkinson, 2018).
The IMF(2024) notes increasing medium-term vulnerabilities in global financial stability, especially in micro-organizations. Thus, assessing the financial health of institutions, as emphasized by (Schumacher et al., 2020), is vital. Financial distress is often misunderstood, with short-term cash flow issues mistaken for deeper problems (Charalambakis, 2014).An effective Early Warning System (EWS) can help monitor and predict financial risks, enabling proactive measures to prevent crises.

Financial ratios derived from financial statements are key indicators of a company's financial strength and are widely used in predictive models (HIdayat, 2018; Hery, 2018). Predicting financial status is crucial for economic planning and safeguarding institutions. Kenya has seen several companies, such as Resolution Insurance and Uchumi Supermarket, face financial distress and collapse. Early detection of financial distress allows for timely intervention, helping to avoid bankruptcy.

This study focuses on leveraging machine learning, particularly supervised networks, to develop a scalable and adaptable EWS model that accurately predicts financial distress. Traditional models like Beaver's FDP (1968), Altman's Z-score (1968), and Ohlson's Linear Regressive Analysis (1980) have limitations in capturing the complexity of financial data (Inglada-Perez, 2020; Khan & Khan, 2018; Buigut& Cherotich, 2018). Modern techniques like Support Vector Machines (SVM) and Artificial Neural Networks (ANN) offer better accuracy by handling non-linear relationships and real-time data (Derbentsev et al., 2020; Shie et al., 2012). However, challenges remain, particularly in dynamic environments like Kenya's micro-financial sector. Developing a robust, dynamic EWS model is crucial for enhancing stability and resilience in this sector.

Methodology

The methodology details the study's design, data processing, ANN model specifications, and performance metrics. It describes the multilayered perceptron Backpropagation Artificial Neural Network (MLP-ANN) used, including parameter optimizations across its three layers. Data was normalized using a min-max scaling approach, setting values between 0 and 1 for better model convergence. The model predicts financial distress using financial ratios related to liquidity, profitability, and leverage, derived from the financial statements of 12 microfinance institutions over a 5-year period (2019-2023). Summary statistics for these ratios are provided in ***Table1***.

Table1: Financial Ratios Related to liquidity, Profitability, and Leverage (2019-2023)

Ratio	Mean	SD	Kurtosis	Skewness
Working Capital to Total Assets	0.8872	1.8167	9.1960	3.0243
Retained Earnings to Total Assets	1.3057	5.9551	47.132	6.8574
Earnings Before Interest and Tax to Total Assets	0.1067	0.3493	4.2624	0.07216
Market Value of Equity to Total Liabilities	0.4121`	1.8359	10.574	2.7279

Sales-to-Total Assets Ratio	0.2422	0.4453	0.99003	0.1833
Return On Equity	0.4914	1.4951	5.3787	2.1979
Current Ratio	1.8554	1.6817	4.2855	2.2323
Gross Profit Margin	0.2135	0.5254	0.97213	0.3602
Debt Ratio	0.7568	0.3648	0.5097	0.5113
Asset Turnover	0.8782	1.8219	9.1501	3.0035

The Dynamic Artificial Neural Network Model

It is a supervised machine learning approach with three layers: an input layer receiving features, hidden layers, and an output layer for binary classification (Figure 1). Sequential algorithms stack these layers, using activation functions for optimization. Backpropagation adjusts model parameters by transferring error terms from the output to the input layer, refining features along the way. The output layer then classifies instances as either distressed (1) or healthy (0).

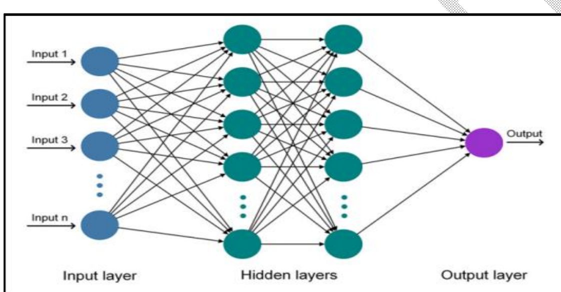

Figure 1: ANN Model (Source SafronEdge)

Input Layer:

Let $x \in R^n$ denote normalized financial ratios which are defined at the initial state of the model with *n* number of nodes and passed to the first hidden layer without any modification.

Hidden Layers:

This next model component receives features from the output layer, learning patterns and establishing non-linear relationships through supervised learning.It uses hyper-parameter tuning and regularization, with optional dropout to prevent over fitting. The Rectified Linear Unit (ReLU) activation function is employed in the hidden layers for its computational efficiency and ability to avoid the vanishing gradient problem. Let S^l represent the state of features from the output layer to the first hidden layer (l = 1), with w_{ij} as the weight matrix and b_i as bias vectors. These are optimized to create the next pre-state P_i through ReLU activation, which then serves as input for subsequent hidden layers dynamically obtained $l - hidden\ layers$.

For $layer$ = 1

$$S^1 = \sum_{i=1}^{n}(w_{ij}^1 x_i + b_i) = h\sum_{i=1}^{n}(w_{ij}^1 x_i + b_i^1) = P^1$$ (Error! Bookmark not defined.)

For $layer$ =2

$$S^2 = \sum_{i=1}^{n}(w_{ij}^2 P^1 + b_i) = h\sum_{j=1}^{n}(w_{ij}^2 P^1 + b_i^2) = P^2$$ (Error! Bookmark not defined.)

For last settled layer

$$S^l = \sum_{j=1}^{n}(w_{ij}^{l-1} P^{l-1} + b_i) = h\sum_{j=1}^{n}(w_{ij}^{i-1} P^{l-1} + b_i^{i-1}) = P^l$$ (Error! Bookmark not defined.)

Basically, the S^l forms the linear combinations for the inputs to the j^{th} neuron while P^l forms the linear combinations of the outputs from the i^{th} neuron.

Output layer:

This final component of the network computes a weighted sum of outputs from the last hidden layer, adds a bias term, and applies the sigmoid function (σ) to convert the output into a probability between 0 and 1, ideal for binary classification. Let P^o represent the output linear combinations from the last hidden layer and $\hat{y}$ the predicted probabilities, calculated through repeated iterations until error minimization. The predicted probability is given by equation **(**Error**!** Bookmark not defined.**)**

$$\hat{y} = \sigma\sum_{i}^{n} w^o P^o + b^o = \frac{1}{1 + e^{P^o}}$$ (Error! Bookmark not defined.)

where σ is the sigmoid activation function, and w^o and b^o are the output layer's weight matrix and bias vectors. Sigmoid activation is effective for binary classification, as noted by (Chollet & Allaire , 2021).

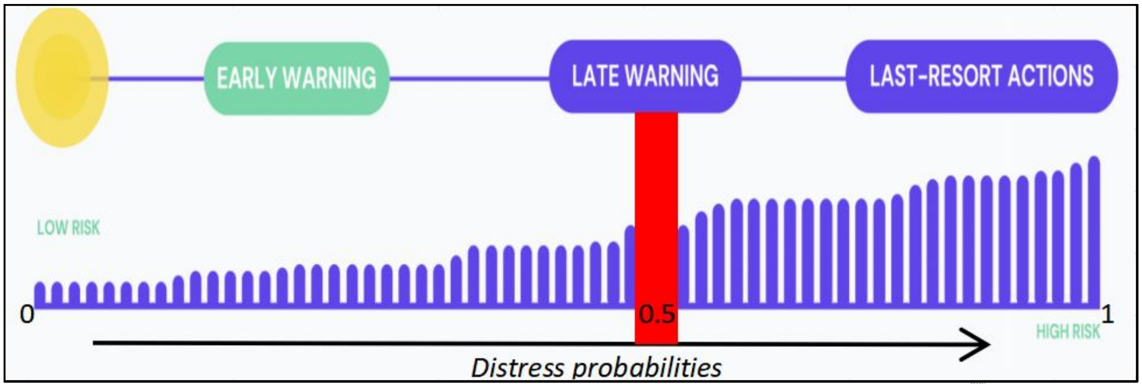

Figure 2: EWS Model

Gradient Descent algorithm and Backprobagation of errors

This algorithm is vital in supervised learning models for minimizing errors during training. It optimizes model parameters (weights and biases) by calculating gradient descent variants to reduce the error or loss function. When an error (the difference between actual and target values) is detected, it is propagated backward from the output layer through the hidden layers to the input layer, a process known as backpropagation.

Let y_i be the predicted values and t_i the target values. The function $F(x_i)$ maps the relationship between output y_i and input x_i and is defined as;

$$F(x_i) = y_i$$ (Error! Bookmark not defined.)

The $F(x_i)$ is determined by an error term *E, which* depends on model parameters (*weights and biases*). It maps the difference between predicted and target values using binary cross-entropy, as shown in equation (Error! Bookmark not defined.).

$$E(W_{ij}b_i) = \sum_{i}^{p} [t_i \log(y_i) + (1 - t_i) \log(1 - y_i)]$$ (Error! Bookmark not defined.)

To minimize the error term *E*, the model adjusts its parameters (weights and biases) by moving in the opposite direction of the loss function's gradient using a gradient descent algorithm with a dynamically chosen learning rate.

Let $w_{ij}^{(l)}$ represent the weight matrix connecting the j^{th} neuron in layer l to the i^{th} neuron in layer $l-1$, $b_i^{(l-1)}$ represent the vector bias associated with the i^{th} neuron in layer $l-1$, and $\propto$ be the models learning rate, then the updated model parameters will be given as;

$$w_{ij}^{(l)} \xleftarrow[\text{Backward move}]{} W_{ij}^{(l)} - \propto \frac{\partial E}{\partial w_{ij}^{(l)}}$$ for the weights,

$$b_i^{(l)} \xleftarrow[\text{Backward move}]{} b_i^{(l)} - \propto \frac{\partial E}{\partial b_i^{(l-1)}}$$ for the biases,

Adam optimizer was ideal to optimize the learning process as it combines the strengths of the Adaptive Gradient Algorithm (AdaGrad) on sparse gradients and Root Mean Squared Propagation (RMSProp)'s effectiveness with non-stationary data. These updates reduce the error by adjusting the weights and biases to minimize the loss function over successive iterations.

Model Specification

The model was built using TensorFlow-Keras sequential API involving specified optimizer, loss function, and evaluation metrics, which collectively defined the model. Adam optimizer was used to update the models' parameters because of of its ability to dynamically adjust the learning rate optimizing how the model updates its weights based on the gradient variations of the loss function.

For binary classification task, binary cross-entropy was used as the loss function, which quantified the error by comparing the predicted probability distribution to the actual distribution, effectively guiding the model to improve its predictions over time. The model architecture was flexible, with the number of hidden layers dynamically determined based on where model was most effective.

Three different learning rates; 0.01, 0.001, and 0.0001 were tested during hyper-parametric tuning to identify the one with the highest accuracy. Debugging reports were systematically generated to monitor training process. Keras Tuner was used with a Random Search algorithm for systematic exploration on the models' space to determine the optimal combination of units, dropout rates, and learning rates that maximized accuracy.

During model training, the best model configuration was identified through hyper-parameter tuning, compiled and evaluated then recompiled with a fixed learning rate for final evaluation

Results and Discussion

This chapter presents a TensorFlow Keras model for predicting financial distress in Kenyan microfinance institutions by analyzing financial ratios from 12 institutions, covering 85% of the sector. The model, optimized and evaluated for accuracy, precision, recall, and F1-score, uses a Z-Score threshold of 1.8 to distinguish distressed (coded as 1) from healthy (coded as 0) institutions. A Z-Score below 1.8 indicates high bankruptcy risk, while scores above 1.8 suggest financial health.

The study calculated Z-Scores annually for ratios including working capital/total assets and market value of equity/total liabilities, using Monte Carlo simulations to add two more years of data for improved model training. The model was trained on pre-processed datasets with batch sizes from 32 to 512 over 50 epochs, with 20% reserved for validation. Key steps included hyper parameter tuning and backpropagation to minimize binary cross-entropy loss

(Figure 2). The architecture, featuring dense and dropout layers, had 36,389 parameters in total, 12,129 of which were trainable, and utilized the Adam optimizer.

Regularization techniques such as dropout layers, checkpoints, and early stopping enhanced model generalization. The Keras Tuner was used to prevent overfitting and under-fitting, with a final learning rate of 0.01 and an optimal epoch count of 20 (Figure 3). Loss curves (Figure 3) showed stable validation loss, indicating strong model generalization.

Predicted probabilities were classified using a 0.5 threshold. The model's performance, summarized in the confusion matrix (Figure 4), showed it correctly predicted 118 healthy instances (True Negatives), 107 distressed cases (True Positives), and incorrectly predicted 11 non-distressed cases (False Positives) and 3 distressed institutions (False Negatives). It achieved a recall rate of 94%, precision of 92%, and an F1 score of 87%. The ROC-AUC (Figure 5) demonstrated a high AUC of 0.99, reflecting excellent performance in distinguishing between distressed and non-distressed institutions.

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 96)	1,056
dropout (Dropout)	(None, 96)	0
dense_1 (Dense)	(None, 512)	49,664
dense_2 (Dense)	(None, 384)	196,992
dense_3 (Dense)	(None, 1)	385

Total params: 744,293 (2.84 MB)
Trainable params: 248,097 (969.13 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 496,196 (1.89 MB)

Figure 3: Model Parameters

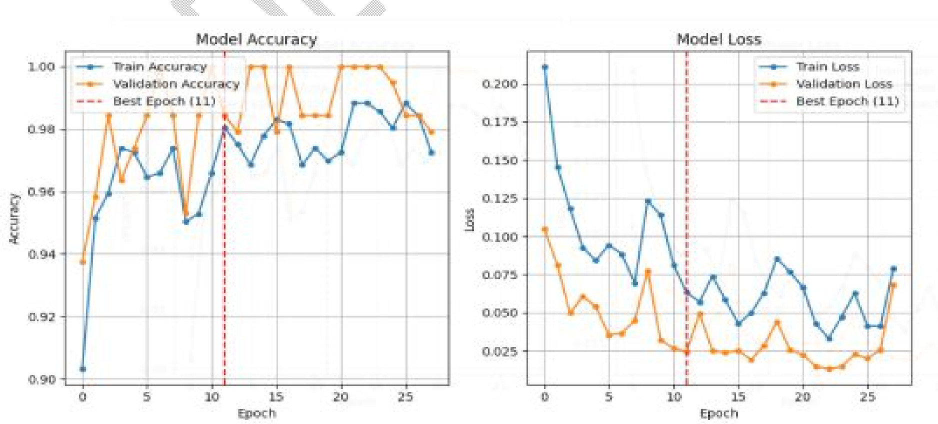

Figure 4: ANN Training Process

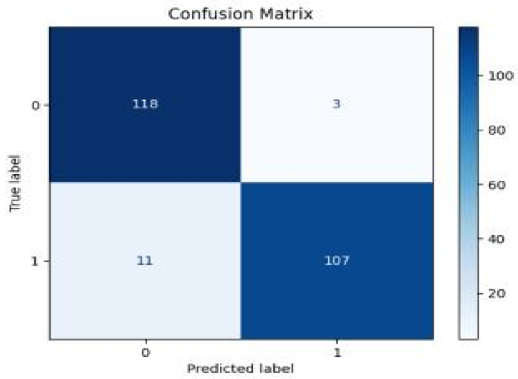

Figure 5: Confusion Matrix

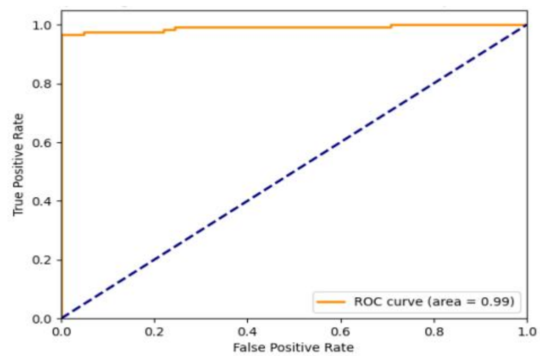

Figure 6: ROC-AUC of the model

Conclusion and Recommendation

The Early Warning System (EWS) developed in this study uses Artificial Neural Networks (ANN) to help organizations make informed decisions in a rapidly changing business environment. ANNs are effective due to their ability to handle complex, non-linear data that traditional methods struggle with. This model, with a high ROC-AUC score of 0.99, accurately predicts financial distress showing enhanced predictions for risk management.

To further improve the EWS, we recommend integrating a dynamic adjustment mechanism within the ANN-GARCH framework. This enhancement aims to capture volatility in predictions and refine accuracy for risk assessment, which will be explored in the future research.

References

Agarap, A. F. (2018). Deep Learning Using Rectified Linear Units (ReLU). arXiv preprint arXiv:1803.08375.

Beaver, W. H. (1968). The Information Content of Annual Earnings Announcements. Journal of Accounting Research, 67-92.

Buigut & Cherotich. (2018). Effect of Corporate Diversification on Capital Structure! Evidence from Listed Firms in Nairobi Security Exchange. American Based Research Journal, Vol. 7 Issue 02.

Charalambakis, E. (2014). On Corporate Financial Distress Prediction: What Can We Learn from Private Firms in a Small Open Economy? Bank of Greece Working Paper No. 188, Available at SSRN: https://ssrn.com/abstract=4184663 or http://dx.

Cheserek, B. K. (2007). The determinants of bank failures: A survey of commercial banks in Kenya.University of Nairobi.

Chollet & Allaire . (2021). Deep Learning with R, 2nd Edition. Manning Publications.

Derbentsev et al. (2020). Machine learning approaches for financial time series forecasting.
Hery, P. (2018). From social network to firm performance: The mediating effect of trust, selling capability and pricing capability", Management Research Review, Vol. 41 No. 6, pp. 680-700. https://doi.org/10.1108/MRR-03-2017-0080. 680-700.
HIdayat. (2018). , Introduction to financial statements, (2018) https://www.scribbr.com/methodology/descriptiveresearch 37.
Inglada-Perez, L. (2020). A comprehensive framework for uncovering non-linearity and chaos in financial markets: Empirical evidence for four major stock market indices. Entropy, 22(12), Article 1435. https://doi.org/10.3390/e22121435.
Khan & Khan. (2018). The Impact of Macroeconomic Variables on Stock Prices: A Case Study of Karachi Stock Exchange.
Outecheva, N. (2007). Corporate Financial Distress: An Empirical Analysis of Distress Risk. Unpublished Doctorate Dissertation, University of St. Gallen.
Ozgul et al. (2010). Combined Detection Model for Criminal Network Detection . In: Chen, H., Chau, M., Li, Sh., Urs, S., Srinivasa, S., Wang, G.A. (eds) Intelligence and Security Informatics. PAISI 2010. Lecture Notes in C.
Quadir & Jahur,. (2012). Financial distressed firms. 21-27.
Robert & Mugo . (2018). Prediction of financial distress in the light of financial crisis: a case of listed firms in kenya, Kenya Egerton University.
Schumacher et al. (2020). Sustainable finance in Japan. Journal of Sustainable Finance & Investment, 213-246.
Shie et al. (2012). Prediction of corporate financial distress: an application of the America banking industry. Neural Compution & Application 21, (2012). https://doi.org/10.1007/s00521-011-0765-5. 1687–1696.
Altman, E. I. (2019). Fifty Years of Z-Scores to Predict the Probability of Corporate Bankruptcy (December 1, 2019). Journal of Investment Consulting, Vol. 19, no. 1, Available at SSRN: https://ssrn.com/abstract=352267. 15-22.
IMF 2024 Global Financial Stability Report https://www.imf.org/en/publications/gfsr