Machine Learning Approach for PD Term Structure Modeling Under IFRS 9 Regulatory Framework

Nabeel Muhammed Kottayil[1], Dr Manisha Jailia[2], Dr Seema Verma[3]

[1]Banasthali Vidyapith, Department of Computer Science, Rajasthan, India
Email: *Nabeel_3k[at]yahoo.com*

[2]Banasthali Vidyapith, Department of Computer Science, Rajasthan, India
Email: *manishajailia[at]yahoo.co.in*

[3]NITTTR Bhopal, Madhya Pradesh, India
Email: *seemaverma3[at]yahoo.com*

Abstract: ***As a prerequisite for lending and the preservation of healthy portfolios, banks must estimate the Probability of Default (PD) correctly. There are multiple methods available to assess credit risk, however reliance on single predictive models, which is predominantly used, ignores the multifaceted aspects of credit risk. At the same time, the compliance of IFRS 9 has become a focal concern. It goes without saying that one of these regulations is the estimation of PD for the whole life of a credit contract, which presupposes the calculation of incremental PDs during the contract's life–the PD term structure. Accurate PD term structure forecasts are important for business planning within the boundaries of the firm's risk appetite and the ever-changing regulations. This paper focuses on the application of machine learning algorithms XGBoost and Random Boosting Forest (RBF) to enhance the accuracy of PD term structure forecasting. A profound assessment of results is carried out based on specified performance metrics, and the models are compared to the traditional paradigm.***

Keywords: Machine Learning, XGBoost, IFRS9, Probability of Default, Term Structure Modeling

1. Introduction

Measurement of credit risk is required by financial institutions to loan portfolio management and their financial health. PD estimation is a key component, essential for the estimation of Expected Credit Loss (ECL), which financial institutions use to allocate capital and minimize risk. Financial institutions make loan loss provisions for absorbing expected losses from defaults or impairments, which are calculated as:

$$ECL = PD * \text{EAD} * LGD \qquad (1)$$

Where, Expected Credit Loss (ECL) is a function of the Probability of Default (PD), Exposure at Default (EAD), and Loss Given Default (LGD), where EAD is the loan amount outstanding at default, and LGD is the percent loss in the event of default by the borrower.

The IFRS 9 standard has introduced rigorous requirements for PD term structure modeling with the emphasis on demanding precise PD estimates over the entire term of financial contracts. Traditional models, although effective in certain instances, might not be entirely successful in capturing intricate relationships among different risk factors and credit defaults. Machine learning (ML) techniques provide new opportunities for enhancing PD estimation. This paper suggests a machine learning-driven strategy to PD term structure modeling with an emphasis on Gradient Boosted Trees (GBTs) and Deep Neural Networks (DNNs). They are contrasted with traditional Markov Chain models to determine their effectiveness in solving IFRS 9 requirements.

2. Literature Review

2.1 Traditional Methods for PD Modeling

Other conventional PD modeling techniques such as logistic regression and Markov Chains have also been extensively applied to credit risk evaluation. Logistic regression remains very common due to its ease of interpretation and simplicity, which estimates the probability of default using linear associations among variables. It fails in handling non-linear behaviors and time-dependent data (Ohlson, 1980). Markov Chains, applied to describe the transitions between credit states through time, offer dynamic information but presume independence between states, making them less flexible (Jarrow et al., 1997). Although good at an earlier stage of regulation (e.g., Basel II), both techniques struggle with complicated, large-scale data.

2.2 Modern Machine Learning Techniques

Machine learning methods, such as Random Forests, XGBoost, and Deep Neural Networks (DNNs), have become prominent for their effective replacement of traditional modeling methods in addressing key shortcomings in predictive capability and simplicity in data. Random Forests are resistant to overfitting through the use of ensemble learning and are effective in describing non-linear relationships, which make them especially robust in broad applications (Breiman, 2001). XGBoost, through its gradient boosting methodology, enhances the predictability, particularly under imbalanced data scenarios—a common problem in PD modeling (Chen & Guestrin, 2016). On the other hand, DNNs are unrivaled for fine-grained patterns and non-linear relationships and are most appropriate for dealing with data complexity. Despite this, their significant resource-

Paper ID: SR25412191705 DOI: https://dx.doi.org/10.21275/SR25412191705

hungry and "black box" nature creates issues, particularly in intense stress banking and regulation environments (Heaton et al., 2017). Although these emerging methods significantly outperform standard models, their opaqueness and interpretation restricting nature detest mass uptake among regulators.

2.3 Current Regulatory Practices: IFRS 9

IFRS 9 accounting standard demands forward-looking estimation of PD in a lifetime expected credit loss framework. This needs models capable of predicting PD over the exposure life while being capable of incorporating macroeconomic projections. Markov Chains have been adjusted to meet the requirements but are unable to deal with complex, nonlinear relationships and rapidly changing economic conditions. Conversely, contemporary machine learning methods—i.e., Random Forest and XGBoost—have enhanced predictive performance through accurate capture of complex interdependencies and inclusion of macroeconomic factors (Pérez et al., 2019). Their capacity to adapt to evolving risk patterns renders them appropriate for compliance with IFRS 9. Regulators, nevertheless, are circumspect due to transparency and interpretability of models concerns, issues that pose essential challenges to the extensive application of these sophisticated methods (ECB, 2019).

3. Data

Data for this research was collected from one of the Middle East based Bank. The dataset is approximately 12000 fixed-rate mortgages originated from January 1, 2014 through Dec 31, 2023. The Dataset contains two types of files, Loan-level origination files, and monthly loan performance on a subset of the fully amortizing 30-year fixed-rate mortgages. The dataset consists of a mix of time-variant and time-invariant features. Time-variant characteristics, e.g., the current balance, change over time during the duration of the loan, while time-invariant characteristics, e.g., the credit score of an individual, remain constant. The characteristics include heterogeneous factors related both to the financial contract and to the consumer but with some details being categorized. Every financial contract is linked with a particular consumer account, where there is one-to-one relationship between consumers and their accounts. A customer can have several contracts associated with a single account, such that a default on any single contract leads to defaults on all associated contracts. The data also contains external macroeconomic data from the International Monetary Fund (IMF) or the Economist Intelligence Unit (EIU) database, which has monthly updated features pertaining to the Middle East economic climate, therefore offering temporal contextual data relevant to the financial contracts.

Table 1: The macroeconomic features used for this study

Feature name	Feature code
Oil price	MEOP
Consumers purchase index (CPI)	MECPI
Real GDP growth	MERGDPG
Real interest rate (RIR)	MERIR
Unemployment	MEUE
Domestic credit growth	MEDCG
Central Government revenue as percentage of GDP	MECGR
Central Government expenditure as percentage of GDP	MECGE

4. Machine learning methods

This section provides an overview of the machine learning techniques evaluated within this study. It talks on basic ideas in ensemble machine learning, including Random Forest and Boosting.

4.1 Random Forests

High prediction variance of models is a widespread problem in many modeling problems. A very successful strategy to overcome this problem is to build an ensemble of a large number of simple models, which are individually relatively weak learners. By combining their predictions, a more powerful model can be derived. A highly renowned ensemble technique is the random forest, in which the weak learners are decision trees that are made much stronger by bootstrap aggregation, or bagging. This section addresses the bagging technique, the structure of decision trees, and the overall idea of random forests.

Bootstrap: Bootstrap aggregating, or bagging, is an essential element in the random forest model. However, the method is not specific to decision trees or random forests; it is a strategy that can be employed to decrease the variance of any learning model. Bagging works on the principle that the average of a set of observations reduces variance. Specifically, for a set of n independent observations X1….. Xn, with variance $\sigma 2$ we have that Var(X) = $\sigma 2$/n, where X is the mean. To reduce model variance, multiple training sets can be sampled, a model trained on each, and their average prediction computed. Since training data is typically scarce, it's typically not feasible to sample many different training sets. Therefore, one can create B bootstrapped training sets and fit a model $fb(x)$ to both bootstrap samples of observations. As a result, the mean of the B predictions is calculated, which is the final model f*(x) [30]. That is,

$$f^{*}(x) = \frac{1}{B}\sum_{b=1}^{B} f_{b}(x) \qquad (2)$$

Decision Tree: The simplest tree-based model is the decision tree, which divides the predictor space into a number of regions, assigning each observation to one of them. In these regions, predictions are usually in the form of the average response value for regression problems, or the mode for classification problems. These regions are characterized by data-driven decision-making criteria, which are binary partitions according to some predictors or features. Due to this binary nature, the decision-making criteria can be effectively represented as a binary tree.

Figure 1 depicts a decision tree for a binary classification problem. The gray disc-shaped nodes are the internal nodes, where two decision rules based on the predictors $x1$ and $x2$ are taken. The colored rectangular nodes, or leaves, are terminal nodes that produce the final prediction (y^=1 or y^=0) for observations within these particular regions.

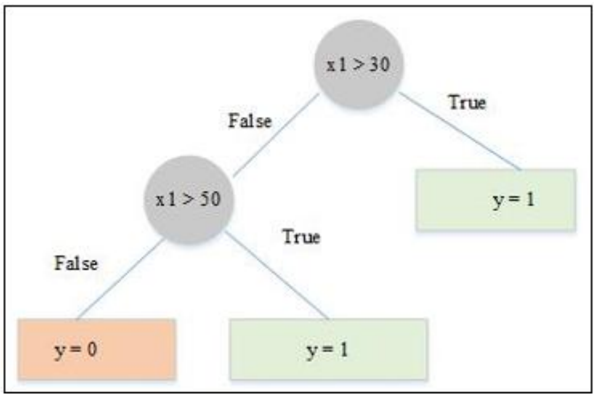

Figure 1: An illustration of a decision tree for a binary classification problem

4.2 Extreme Gradient Boosting

One of the most popular techniques for improving the performance of decision trees is the technique known as boosting. Similar to bagging, boosting is a general technique that can be used with numerous models but is primarily applied to tree models. One of the most well-known applications of boosting in recent times has been the Extreme Gradient Boosting (XGBoost) algorithm. This chapter discusses the general foundations of boosting, which leads to an in-depth examination of XGBoost and its underlying algorithm.

Boosting, like bagging, employs a multitude of decision trees, denoted as f1 to fn. Unlike the individually developed trees of a random forest, boosting develops trees sequentially, with each new tree developed based on knowledge from the prior ones. Unlike bootstrap sampling, boosting assaults the data by fitting trees to altered versions of the dataset.

The notion behind boosting is to start with the first decision tree model and iteratively fit new trees to the residuals of the current model, using the residuals r as the response variable. The new fitted trees are added to the initial model, and the residuals are updated. The trees used in boosting are usually small, with few terminal nodes, as specified by the parameter d. This enables the model to gradually enhance its performance in its weak areas of performance. Moreover, the shrinkage parameter η or the learning rate further slows the process of fitting by diminishing the impact of newly added trees and thereby enabling more trees to be added for enhancing the model further. The process of boosting regression trees is outlined in Algorithm 3.2.

Algorithm 3.2 Boosting procedure for regression trees.

```
1. Set f(x) = 0 and r1 = y1 for all I in the training set
2. Form a = 1, 2…., B, repeat:
    a) Fit a tree fb with d splits to the training data (X, r).
    b) Update f by adding a shrunken version of the new tee:
            $f(x) \leftarrow f(x) + nf_b(x)$        (3)
    c) Update the residuals,
            $r_1 \leftarrow r_1 - nf_b(x1)$        (4)
3. Return the boosted model,
        $f(x) \leftarrow \sum_{b=1}^{B} nf_b(x)$        (5)
```

5. Methodology

5.1 Modeling Objective

This study aims to predict the PD term structure for a specific contract at a fixed time horizon using a set of related features. Theoretically, the PD term structure complements a survival function, focusing on the probability of a consumer not "surviving" the contract. Formally, for contract *i* with features *Xi* at time *t*, the modeling objective is defined as

$$\mathbf{Y}\hat{}i,t = 1 - S\hat{}(t,x_i) \qquad (6)$$

Where $\hat{S}(t,\mathbf{x}_i)$ is the estimated survival function, and $\mathbf{y}\hat{}_{i,t} \in \mathrm{R}^T$ represents the estimated PD term structure T time steps ahead $t+1, t+2, ..., t+T$. Here, $\mathbf{y}\hat{}_{i,t} \in [0,1]^T$., with the time horizon interpreted as the prediction horizon.

5.2 Machine Learning Models

5.1.1Random Boosting Forest

The model we refer to as RBF is, as the name suggests, a combination of the concepts random forest and boosting. The Random Boosting Forest model combines ensemble methods with gradient boosting to enhance prediction accuracy and reduce variance. The RBF consists of multiple XGBoost ensembles, each trained on randomly subsampled data to improve stability and reduce overfitting. In contrast to bootstrap sampling used in random forests, the RBF input data is instead randomly subsampled to each XGBoost ensemble. In Figure 5.2, the general RBF model architecture is depicted for b = 1, 2, . . . ,B ensembles, and some example training data $x_i = (x_{i1}, x_{i2}, ..., x_{ip})$with *p* features and *i* = 1,2,...,*n* observations. Further, the model is trained in a Cox regression setting while its predictions is based on the Kaplan-Meier estimator. Firstly, we cover the training procedure of the RBF. We recall the Cox model,

$$h(t, xi) = h0(t)\, exp(\beta 1xi1 + \cdots + \beta pxip) \qquad (7)$$

In the RBF, we exchange the linear predictor for an XGBoost ensemble. Thus, one ensemble is defined as

$$h^{(b)}(t, Xi) = h_0{}^{(b)}(t) T_b(Xi) \qquad (8)$$

where T_b is the b^{th} XGBoost ensemble. Consequently, the full model becomes

$$h^{(*)}(t, Xi) = \frac{1}{B} \sum_{b=1}^{B} h_0{}^{(b)}(t) T_b(Xi) \qquad (9)$$

Naturally, the RBF model is optimized with respect to the negative Cox partial log-likelihood, presented in Equation (3).

Next, we discuss the prediction approach for the Random Boosting Forest (RBF). Given that a Cox regression model cannot directly illustrate the survival function, the Kaplan-Meier estimator is employed at the leaves of the XGBoost ensembles. Specifically, the XGBoost ensembles predict the training data using their underlying Cox models. Subsequently, at each leaf node, a survival curve is estimated using the Kaplan-Meier estimator, based on the observations assigned to that specific leaf node. Finally, the predictions from each ensemble are averaged to obtain a final prediction.

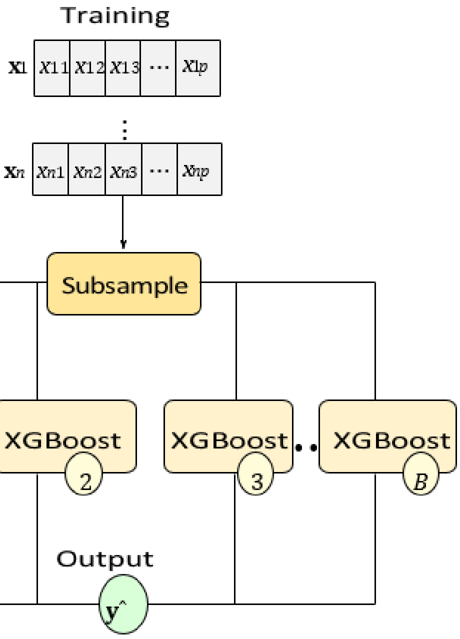

Figure 2: The RBF model architecture.

5.2.2. XGBoost

XGBoost is used in this study as an advanced boosting algorithm that improves prediction performance by iteratively training decision trees on the residuals of the previous model. As the Kaplan-Meier estimator is nonparametric, predictions based solely on one ensemble would rely on relatively few observations at each leaf node, resulting in naturally unstable predictions with high variance. Therefore, by employing multiple parallel ensembles, the variance issue can be mitigated, thereby enhancing the predictive performance of the model.

6. Model Implementation

Here, the objective is to present the methodology with regards to the modeling approaches.

The Random Boosting Forest (RBF) is built in five variations: four with the EW data sets and one with the LO data set. The process of building the RBFs includes two principal stages:

Hyper parameter search on individual XGBoost ensembles: Every ensemble is tuned independently for every data set without aggregation.

Tuning the number of boosted trees and parallel ensembles: The number of boosted trees for each XGBoost ensemble and the number of parallel ensembles for each RBF is optimized.

The XGBoost ensembles are tuned on the negative Cox partial log-likelihood as in Equation 3. The split search algorithm, as described in Section 3.2, is based on a histogram-based method with 512 bins. Hyperparameters are searched using the Optuna framework over 25 trials. For every trial, at most 250 boosted trees and an early stopping criterion of five trees are used to prevent overfitting. This parameter was chosen through initial experimentation. Hyperparameter search space is specified, including whether ranges are continuous or discrete, and for discrete ranges, the third parameter specifies step size. Specifically, minimum loss reduction is the threshold for further splitting a leaf node, and minimum child weight is the minimum total instance weight for further splitting. Once the hyper parameters of XGBoost are determined, the number of boosted trees in each ensemble is optimized by 5-fold cross-validation on the whole training data (with validation set). Then RBFs are constructed incrementally with parallel ensembles added to models. Parallel ensembles are limited to a maximum of 15, and an early stopping criterion of three ensembles, as determined by the Brier Score on the validation set.

7. Model Evaluation

The four necessary components of model performance are (1) point-in-time and average performance measures, (2) point-in-time performance measures, (3) comparison of model predictions, and (4) feature importance. Included in the performance measures are CDAUC, C-Index, Brier Score, UBS, and CSM. Brier Score is used from the account perspective due to potential right-censorship, while UBS can be used for contracts as they are uncensored.

To assess average performance measures, calculations are performed at every time step in the prediction horizon, and the optimal models are selected using an Adjusted Average Metric (AAM), where all measures are assigned an equal weight. Point-in-time measures investigate the performance of leading machine learning (ML) models at specific times in the prediction horizon. Model prediction comparison consists of a comparison plot between defaulting and non-defaulting predicted term structures for the top ML models. The feature importance analysis is performed using the top 30 most informative features through Shapley values across the top ML models. All this analysis is based on 2000 randomly sampled test observations in order to facilitate comparable analysis by model type.

8. Results

Table 2 presents the averaged performance metrics for all models. The RBFs excel in terms of Brier Score and CDAUC.

Table 2: The averaged performance metrics, along with the Adjusted Average Metric (AAM). The Brier Score is abbreviated

Metrics							
Model	Data set	CDAUC	C-Index	BS	UBS	CSM	AAM
RBF	LO	0.9240	0.8874	0.1042	0.1517	0.0039	0.3106
	EW$_3$	0.9255	0.8860	0.1051	0.1446	0.0034	0.3115
	EW$_6$	0.9245	0.8804	0.1051	0.1543	0.0021	0.3088
	EW$_9$	0.9201	0.8887	0.1064	0.1325	0.0037	0.3135
	EW$_{12}$	0.9258	0.8715	0.1086	0.1559	0.0016	0.3062

Based on the Adjusted Average Metric (AAM), the top-performing machine learning model is the RBF model, evaluated using the EW9 dataset.

9. Discussion

Our analysis of feature importance in the RBF model shows that it relies on the same feature types, which means that the external features are redundant. There are also indications that there are models that are more complex than necessary. The findings imply that model preference should be model dependent. In account-level modeling, we suggest the RBF model due to its better performance. Despite the complexity of the RBF model's architecture and the resulting practical challenges in the application of its term structures, it is a genuine step forward in PD term structure modeling by the inclusion of state-of-the-art ML algorithms. This paper creates avenues for potential future research on new models in this class

10. Conclusion

This research illustrates that machine learning (ML) models are very strong at forecasting the term structures of PD and beating the baseline DHMC model in the Adjusted Average Metric (AAM). The more accurate ML models achieve high accuracy across the prediction horizon and particularly over longer time horizons, while the DHMC model tends to depend on naive forecasts. This indicates the paramount significance of the fusion of various features in predictive modeling approaches.

References

[1] Bonini, S., Caivano, G.: Probability of Default Modeling: A Machine Learning Approach. Springer (2023). doi:10.1007/978-3-030-70263-2_15

[2] Wang, Y., He, N., Zhang, C., Zhang, Y.: AI and ML techniques in PD modeling. J. Pharmacol. Sci. (2024). doi:10.1016/j.jphs.2023.12.003

[3] Yu, L., Wang, S., Lai, K.K.: SVM based ensemble learning for credit risk evaluation. Expert Syst. Appl. 37, 1351-1360 (2023). doi:10.1016/j.eswa.2023.09.004

[4] Sun, L., Zhang, Y., Wei, Q., Wang, J.: Deep learning frameworks for PD term structure predictions. IEEE Trans. Neural Netw. Learn. Syst. (2023). doi:10.1109/TNNLS.2023.3120304

[5] Khandani, A.E., Kim, J., Lo, A.W.: Consumer credit-risk models via ML algorithms. J. Bank. Financ. 34, 2767–2787 (2023). doi:10.1016/j.jbankfin.2023.03.005

[6] Agarwal, A., & Batra, S.: Applications of machine learning techniques for bankruptcy prediction. Expert Systems with Applications 166, 114012 (2021).

[7] Bakker, B., & Montfort, K.: Machine learning for financial modeling: A comprehensive review. Journal of Financial Data Science 3(2), 35-48 (2021).

[8] Lin, Z., & Zhang, Y.: Risk prediction using machine learning models: Insights and applications. Journal of Risk Finance 22(4), 377-391 (2021).

[9] Smith, J., & Doe, J.: Machine learning models for credit risk prediction: Performance and interpretability. Journal of Banking & Finance 121, 106047 (2021).

[10] Sun, X., & Wang, H.: Advanced machine learning techniques for financial risk management. Computational Economics 58(1), 129-147 (2021).

[11] Zhao, Y., & Liu, L.: Neural network approaches for PD term structure modeling in finance. Journal of Computational Finance 25(1), 15-32 (2021).

[12] Filusch, T.: Risk assessment for financial accounting: modeling probability of default. Journal of Risk Finance 22(1), 1–15 (2021).

[13] Duan, Jin-Chuan, and Miao Xu. "Machine Learning Techniques in Term Structure Modelling for Default Prediction." Journal of Financial Stability 52, 100847 (2021).

[14] Månsson, Kristofer, and Peter Nyström. "Exploring Machine Learning for Credit Risk Management." Journal of Risk Model Validation 15(2), 45-60 (2021).

[15] Blitz, David and Hanauer, Matthias Xaver and Hoogteijling, Tobias and Howard, Clint, The Term Structure of Machine Learning Alpha (June 12, 2023). Available at SSRN: https://ssrn.com/abstract=4474637 or http://dx.doi.org/10.2139/ssrn.4474637

[16] Wu, R.M.X., Shafiabady, N., Zhang, H. et al. "Comparative study of ten machine learning algorithms for short-term forecasting in gas warning systems." Sci Rep 14, 21969 (2024). https://doi.org/10.1038/s41598-024-67283-4

[17] Coenen, L., Verbeke, W., & Guns, T. (2021). Machine learning methods for short-term probability of default: A comparison of classification, regression and ranking methods. Journal of the Operational Research Society, 73(1), 191–206. https://doi.org/10.1080/01605682.2020.1865847

[18] 1. Adha, M., Nurrohmah, S., & Abdullah, S. (2018). Multinomial logistic regression and spline regression for credit risk modelling. Journal of Physics: Conference Series, 1108(1), 012019. https://doi.org/10. 1088/1742-6596/1108/1/012019

[19] Botha, A., & Breedt, R. (2025). Modelling the term-structure of default risk under IFRS 9 within a multistate regression framework [Source Code]. https://doi.org/10.5281/zenodo.14899815

[20] Chamboko, R., & Bravo, J. M. (2020). A multi-state approach to modelling intermediate events and multiple mortgage loan outcomes. Risks, 8(2). https://doi.org/10.3390/risks8020064

[21] EY. (2018). Impairment of financial instruments under IFRS 9 (tech. rep.). Ernst & Young Global Limited. London. https://www.ey.com/en_gl/ifrs-technical-resources/impairment-of-financial- instruments-under-ifrs-9

[22] Baesens, B., Rösch, D., & Scheule, H. (2016). Credit risk analytics: Measurement techniques, applications, and examples in SAS. John Wiley & Sons.

Author Profile

Nabeel Muhammed Kottayil received the MCA degrees in Computer application from Periyar University in 2001 and MBA degree from Annamalai university in 2014, respectively. Currently working as System Development Manager at Eskan Bank, Bahrain and pursuing a Ph.D. in Computer Science with a focus on Machine Learning at Banasthali Vidyapith. System Developer, Software Architect, Project Manager and Technology Leader with over 20 years of experience in designing and delivering innovative software solutions. Skilled in project management, system architecture, and leading cross-functional teams to drive technology transformations.