Bank Net Interest Margin Forecasting and Capital Adequacy Stress Testing by Machine Learning Techniques

Raymond Brummelhuis[*] Zhongmin Luo[†]

March 31, 2019

Abstract

The 2007-09 financial crisis revealed that the investors in the financial market were more concerned about the *future* as opposed to the *current* capital adequacies for banks. Stress testing promises to complement the regulatory capital adequacy regimes, which assess a bank's current capital adequacy, with the ability to assess its future capital adequacy based on the projected *asset-losses* and *incomes* from the forecasting models from regulators and banks. The effectiveness of stress-test rests on its ability to inform the financial market, which depends on whether or not the market has confidence in the model-projected asset-losses and incomes for banks. Post-crisis studies found that the stress-test results are uninformative and receive insignificant market reactions; others question its validity on the grounds of the poor forecast accuracies using linear regression models which forecast the banking-industry incomes measured by Aggregate *N*et *I*nterest *M*argin. Instead, our study focuses on NIM forecasting at an individual bank's level and employs both linear regression and non-linear Machine Learning techniques. *First*, we present both the linear and non-linear Machine Learning regression techniques used in our study. *Then*, based on out-of-sample tests and literature-recommended forecasting techniques, we compare the NIM forecast accuracies by 162 models based on 11 different regression techniques, finding that some Machine Learning techniques as well as some linear ones can achieve significantly higher accuracies than the random-walk benchmark, which invalidates the grounds used by the literature to challenge the validity of stress-test. *Last*, our results from forecast accuracy comparisons are either consistent with or complement those from existing forecasting literature. We believe that the paper is the first systematic study on forecasting bank-specific NIM by Machine Learning Techniques; also, it is a first systematic study on forecast accuracy comparison including both linear and non-linear Machine Learning techniques using financial data for a critical real-world problem; it is a multi-step forecasting example involving iterative forecasting, rolling-origins, recalibration with forecast accuracy measure being scale-independent; robust regression proved to be beneficial for forecasting in presence of outliers. It concludes with policy suggestions and future research directions.
Keywords: Machine Learning; time series forecasting; forecast accuracy; bank capital; stress testing; net interest margin; systemic risk.

1 Introduction

The 2007-09 financial crisis revealed that the financial market was more concerned about *future* as opposed to *current* capital adequacy problem for large financial institutions, some of which failed or nearly failed but were judged to be "well-capitalized" by the on-going regulatory capital criteria; see a list of such financial institutions in Schuermann (2014). Also, it was clear that such concerns about future capital adequacies with a few large banks (particularly, the so-called global systemically important banks or G-SIBs) can quickly transform into the collapses of financial and economic systems or into a systemic risk. Schuermann's study found that the stress-test exercise conducted by the US regulators in May 2009 unprecedentedly disclosed important information about the projected capital and net revenue results under the worse-case-scenario for some of the largest banks, which, corroborated by observations from Krugman[1], effectively regained

[*] Dept. of Mathematics & Computer Science, Univ. of Reims, Reims, France, raymondus.brummelhuis@univ-reims.fr

[†] Dept. of Economics, Mathematics & Statistics, Birkbeck, Univ. of London, England, zhongmin.luo@btinternet.com.

[1] Krugman P., 14/07/14, "the stress tests marked the end of the financial panic during the financial crisis; after stress testing, the three key measures of financial disruption, ..., all fell sharply back to normal levels over the first half of 2009"; `https://bit.ly/2rlxJbb`

investors' confidence in the banking industry so that the banks were able to raise needed capitals from less volatile markets, and effectively end the crisis. Also, the study by Schuermann highlighted that, whilst proper stress test requires modelling three components for the banks, namely, *asset losses*, *net revenues* and *balance sheet dynamics*, regulators, banks and researchers have focused mainly on quantifying the asset losses when assessing banks' capital adequacy and that the literature about the latter two is very limited. In this study, we focus on forecasting the second component. Meanwhile, we reserve the third component of stress testing for future research; see for example Hirtle *et al* (2015), which included current practice of assuming constant asset growth rates estimated from historical data.

Stress testing has become a central tool used by global regulators to manage financial stability with the promise to address investors' real concerns about banks' future capital adequacies manifested during the financial crisis. Typically, national regulators use their own stress-test models developed for a representative bank based on data collected about banks and macroeconomic variables to project the future capital needs for a typical bank under a set of forward-looking severely stressed macroeconomic scenarios and baseline (normal) macroeconomic scenarios (see for example Hirtle *et al* for the US). Then they use the results to evaluate the numbers reported by individual banks, which include individual banks' future asset-losses, incomes and future capital plans, projected by the banks' own forecast models under the set of forward-looking economic scenarios; based on the evaluation results, regulators identify those banks which need to replenish their capitals and release such results to the public. However, a recent study by Kupiec (2018), which forecast total incomes (interest incomes plus non-interest incomes) for a representative bank based on regulator-like models, raised serious concerns about the absence of information about how regulators evaluate banks' forecasted numbers and concerns about data overfitting and lack of out-of-sample tests about the forecast accuracies of the stress-test models.

Clearly, the creditability of the stress testing as a regulatory tool for financial stability is founded on its ability to inform the financial market, which, in turn, depends on its ability to establish or maintain the market's confidence in the results projected by the three forecast modelling components, namely, the respective models used by regulators and banks to project banks' future asset-losses, incomes and balance sheet dynamics. However, some of post-crisis literature is very critical about the stress testing. For example, Glasserman and Tangirala (2015) highlighted that the US stress-test results have become predictable and uninformative year after year, and therefore subject to gaming by banks. Some others criticized the soundness of first two modelling components of stress-test. For example, Acharya, Engel and Pierret (2014) criticized that the risk weights used to project banks' asset-losses are not risk-sensitive. Using linear regression models, two post-crisis papers respectively by Guerrieri and Welch (2012) and by Bolotnyy, Edge and Guerrieri (2015) found that the forecasted industry-wide Aggregated NIM, which is defined as, at the banking industry level, the net interest earnings to interest-earning asset ratio, is mostly not as accurate as those forecasted by a simple Random-walk model. Thus, the authors of both papers questioned the credibility of the stress test, which we refer to as the *Predictability or Forecast Accuracy Problem for Aggregate NIM*. All above literature has taken a top-down approach to study the forecast accuracy of NIM models for either a representative bank level (see Hirtle *et al*'s or Kupiec's) or for Aggregate NIM at the banking-industry level (see the above two Aggregate NIM literature); unfortunately, no post-crisis literature available in public domain has adopted a bottom-up approach to examine whether the forecast accuracy problem with Aggregate NIM also exist for the bank-specific NIM forecast model used to predict future NIM for a specific bank (to which we refer as a Bank-specific model); in this study, we take up the challenge.

We next turn to explain why we have chosen to study NIM as the net income metric for banks rather than other metrics such as total income (like Kupiec's) or Return on equity (ROE) or Return on asset (ROA). *First*, non-interest incomes for banks such as the capital gains from trading, advisory fees, as a percentage of banks' total incomes (which also include interests earned from interest-earning assets such as loans, bonds, etc.) have decreased sharply after the financial crisis according to our study based on data from the US Fed[2]) across the Euro Area, the United States and the world as a whole. The sharp decrease coincides with a period of post-crisis regulatory reforms which have led to increased capital requirements for investment-banking businesses so that banks have moved back to traditional interest-earning businesses such as lending instead of mainly non-interest-earning businesses such as trading. Thus, NIM has become more reflective of banks'

[2] Federal Reserve Bank of St. Louis for NIM for the US, Euro Area and the World https://fred.stlouisfed.org/series

main business activities than before.

Also, according to a ECB's study (2010), the ROE (return on equity) was criticised after the financial crisis for its short-term focus and incentivizing banks for increased risk-taking while neglecting long-term profitability; the ROA (return on asset) tends to be flat across time, containing little information for predicting potential future profit falls.

In our study, we will examine whether the *forecast accuracy* problem for Aggregate NIM raised in the above NIM literature exists for Bank-specific NIM. The forecasting models we will use include both the linear models used in NIM literature and Non-linear regression models from the Machine Learning arena such as the Regression Tree and its Ensemble-version, Gaussian Process Regression, Support Vector Machines and the NARX (or Nonlinear AutoRegressive models with eXogeneous inputs) version of Recurrent Neural Networks (or RNN, see Lin *et al* (1996) for an introduction of NARX version of RNN and see Graves (2014) for a recent introduction for RNN) and which, from a statistician's viewpoint, can be considered as non-linear regression models.

In the remainder of this introductory section, first, we will discuss the relationships between capital adequacy, bank income and stress test; then we will present the essence of bank-specific NIM forecasting by Machine Learning Techniques, including the economic principles followed by our feature selections; finally, we will survey recent literature on NIM study, review some forecasting techniques from the literature, then present a summary of the research gaps which motivated this study and a summary of what we believe to be this paper's main contributions.

1.1 *Capital Adequacy, Bank Incomes and Stress Test*

As indicated in the FCIC report (the US Financial Crisis Investigation Commission), the global financial crisis was caused by a panic about the financial system triggered by the collapses of a few G-SIBs[3], which in its turn resulted from the fear on the part of the creditors and business partners that these G-SIBs might not be able to honour *future* financial obligations, leading creditors and investors to refuse to roll over the existing short-term funding to these G-SIBs such as 90-day Asset-backed Commercial Paper[4]. Thus, our study will focus on the G-SIBs.

According to Basle III's capital adequacy requirements (2010), a bank's accounting capital K_t^A should, at a reference date t, exceed a fixed constant k times the risk-weighted value of its assets or RWA_t at time t:

$$K_t^A \geq k \cdot RWA_t, \tag{1}$$

where k is the so-called *capital adequacy ratio*. The bank's accounting capital is, by definition, the sum of its *equity* E_t^1 (contributed by its shareholders and subject to a bank's own capital planning, e.g., new share issuance and share buyback.), and its *retained earnings* E_t^2, which is the income generated from its business and investment activities minus the dividends paid to its shareholders.

The objective of stress testing is to examine whether or not at some reference date t, (1) still holds at future times $t+h$ ($h := 1, 2, \cdots, H$), given a set of exogenously given economic scenarios denoted by ω_{t+h} related to time $t+h$, which represent normal and or severely adverse future economic conditions contemporaneous with a response variable $y_{t+h|t}$ for which we will take the predicted NIM at $t+h$, given information available at t; symbolically,

$$K_{t+h|t}^A(y_{t+h|t}, \omega_{t+h}) \geq k \cdot RWA_{t+h|t}(\omega_{t+h}) \tag{2}$$

where $K_{t+h|t}^A(y_{t+h|t}, \omega_{t+h})$ denotes h-step ahead projected accounting capital for time $t+h$ as a function of both h-step ahead NIM to be forecasted (denoted by $y_{t+h|t}$) and ω_{t+h} as well as other information at time t, and similarly for the right hand side.

The 2007-09 crisis manifested the significant difference between (1) and (2) in that the former is concerned with banks' current capital adequacy, whereas, the latter is with their future capital adequacy. During the crisis, the fear of future failure severely affected the share prices of banks such as Morgan Stanley, which lost

[3] Our study includes 6 US G-SIBs: `https://bit.ly/2xC8wh1`

[4] Investment banks tend to rely on short-term funding; whereas, commercial banks rely on deposits.

almost 30% of its share values on the day after the Lehman debacle despite the fact that Morgan Stanley had \$181 billion in cash, just posted earning results that defied analysts' expectations and had a healthy capital ratio[5]. This indicates that the real concerns of investors and banks' creditors were about their future as opposed to their current capital adequacy. The purpose of stress testing and that of this study is to address such real concerns.

Reliable stress-test rests on finding good models for the two sides of (2). A lot of research efforts (see for example McNeil, Frey and Embrechts, 2015) has gone into the modelling side of $RWA_{t+h|t}(\omega_{t+h})$, and in particular of the valuation of the different assets which constitute the bank's portfolio. As regards the portfolio itself, quantitative models for banks' balance sheet dynamics are unrealistically simple in practice; see examples of assuming constant growth rate for assets in Hirtle *et al's* and Kupiec's. Modelling balance sheet dynamics is an interesting and challenging topic[6]: the future equity depends on strategic decisions such as share buyback or new share issuances, which will depend on future capital projection and make predicting future equity holdings a "nested forecasting problem". Modelling the balance sheet dynamic is outside of the scope of the present paper. In this paper we will concentrate on the left-hand side of (2), and in particular on *forecasting retained earnings* component.

1.2 *Essence of Bank-specific NIM forecast by ML-techniques*

Given a set of exogenous economic scenarios contemporaneous with time $t + h$ and conditional on other information at time t, the goal of this study is to forecast h-step ahead NIM, which will then be used to check whether or not the capital adequacy condition (2) holds for a specific bank on on-going basis in stress testing. To that end, as indicated in section 2, we include classical statistical and more recent Machine Learning regression techniques in our study. We will refer to both of these below as *Machine Learning Regression Techniques* or simply as *ML-Regression Algorithms*. In contrast with the Classification models used in Brummelhuis and Luo (2019a), which are used to predict *discrete variables*, the aim of Machine Learning Regression is to devise computerised algorithms which predict the value of one or more *continuous response variables*, on the basis of a vector-valued *feature variable* with values in some finite dimensional vector space $\mathbb{R}^d$. Since our forecast models involve non-linear Machine Learning techniques, these inevitably contain elements of parameterization choices, see Brummelhuis and Luo (2019b) for a reference in the context of classification. We refer to Hastie, Tibshirani and Friedman (2008) for an overview of modern Machine Learning and Tashman (2000) for a comprehensive review of multi-step forecasting including some forecasting techniques used in this study. In this study, we also compare the forecast accuracy of our regression models with those of a benchmark model; see Brummelhuis and Luo (2019b) for a benchmarking exercise in the context of classification. At the beginning of forecasting, it is important to determine whether to formulate a problem as a classification or a regression one and a wrong decision at this point can lead to nonsensical results; see an example of potential nonsensical probabilities among other implications as a result of using cross-sectional regression as opposed to classification for constructing CDS proxy rates in Brummelhuis and Luo (2018c).

Denoting NIM at time t by y_t, we can express the models used for this study schematically:

$$y_{t+h} = f(y_{t+h-1}, \mathcal{X}_{t+h}, \theta_h) + \epsilon_{t+h}, \tag{3}$$

Where y_{t+h-1} denotes an AR(1) term for NIM which we introduce for reasons discussed below. $\mathcal{X}_t$ is the vector of exogenous feature variables representing the macroeconomic variables or bank-specific microeconomic variables (see a list of feature variables in Appendix A.1), both contemporaneous with y_{t+h}; f is a function depending on a set of parameters denoted by h-specific θ_h and ϵ_{t+h} is the forecast error term measured as the gap between $\widehat{y_{t+h}}$ forecasted *ex-ante* and y_{t+h} observed *ex-post*.

Equation (3) applies to both training and forecasting: during training, $h = 0$, we train equation (3) on training samples; during forecasting, $h > 0$, we can form forecasts based on the learned functional form with parameters $\widehat{\theta_h}$.

[5] Wharton; `https://whr.tn/2St2kiF`

[6] Balance sheet modelling requires granular predictive models, e.g., product-specific account balance origination, roll-over rates, attrition rates, prepayment options for products like non-matured deposits, pre-payable mortgages, etc.

During the training stage, we need to make a few related modelling choices: (1) whether we want to forecast based on a fixed forecasting origin or rolling origins; (2) if the choice of rolling origins is decided, whether we want to just update our training set with new data or we want to recalibrate our models at each roll of forecasting origin before forecasting; (3) whether we want a certain feature variables always included in our final model or not for practical reasons, e.g., the board of a company or the regulator might want to see the response of NIM with regard to CPI index, etc. First, fixed-origin forecasting only produces a horizon-specific forecast and a horizon-specific forecasting error, which makes its results susceptible to data noise and makes it difficult to judge the forecast accuracy; in contrast, rolling-origin forecasting leads to multiple horizon-specific forecasting errors, which permits one to estimate a distribution of forecasting errors. Hence, we choose rolling-origin forecasting as recommended by Tashman in our study. Second, given that both our forecasting object and most of our feature data are quarterly, we recalibrate our model to a rolling training set to allow our models to pick up the potential signals in data between quarters. Third, our feature selection is mainly driven by economic principles, we choose to include a default set of economically motivated feature variables, which we call standard feature selections (see FS1-FS10 in Appendix A).

As a result, we evaluate forecast accuracy based on a series of horizon h-specific forecasting errors defined as $\widehat{\epsilon_h} = y_{t+h} - \widehat{y_{t+h}}$, which are always out-of-sample in that sense that the observed y_{t+h} is not part of the training sample that ends at forecast origin t. Bergmeir and Hyndman (2018) is a recent study which includes a discussion of such practice in the context of Leave-one-out cross validation (LOOCV) as a special case of K-fold cross validation. As discussed further in section 2.2, from each rolling window, we obtain h-specific out-of-sample ϵ_{t+h}; we collect ϵ_{t+h} from all rolling-origin forecasts to arrive at a statistics for forecast accuracies measured by a metrics such as Root-mean-squared-forecast-error or RMSE; see section 2.11 for further details regarding the choice of forecast accuracy metrics.

In the remainder of this section we explain the rationale underlying the 10 different Standard Feature Variable selections as described in Appendix A.1, which are economically motivated; we use these 10 Feature Selections in all models with the exceptions of Principal Component Regression and Stepwise Regression, which, for reasons explained in section 2, are based on their own Feature Selections shown in Appendix A.3 and Appendix **??** respectively. In the same vein, the study by Busch and Memmel (2014) contains a discussion about determinants of NIM.

1. First of all, banks are paid for providing two related financial intermediary services, *Term Transformation* (TTS) and *Interest Rate Risk Management* (IRM). Traditionally, banks generate interest incomes by paying their depositors the low rates on the short end of the interest rate curve in exchange for deposits (which are short-term on average) while earning the high rates on the long end of the curve from activities such as mortgage lending. As a result, customers can transform their asset-maturity profile, for example, from short-term cash into long-term houses, which is called Term Transformation. Meanwhile, to provide such intermediary services, banks have to invest in managing interest rate risk, for example by hedging interest-rates using derivatives such as interest rate swaps and swaptions for risks related to interest rate level and volatility respectively. Clearly, both TTS and IRM are influenced by interest rate level and interest rate volatility. However, empirical experience from experimenting with the so-called "Merrill Lynch Option Volatility Index" or MOVE, which is a the standard index for tracking implied interest rate volatility, suggests that interest rate level movements are more important for the bank-specific NIM models of our study. Meanwhile, for loans of revolving exposures such as revolving lines of credits, as expected, our experiences show that MOVE contributes significantly to explain the variations of NIM at asset class level, which is out of the scope of this study. In our study, we therefore naturally use the *Slope of the Interest Rate Curve*, defined as the gap between 10-year and 3-month treasury yields, as one of our feature variables; see Covas, Rump and Zakrajsek (2014) as an example of such a practice in the NIM literature, and Appendix A for further details.

2. Secondly, banks are paid for assuming the *credit risk* resulting from the losses due to potential defaults of their many borrowers. To quantify this *collective credit risk*, we need a *broad* market index with sufficiently long time series data which reflects the aggregate credit risk. Two widely used CDS indices, CDX and ITraxx were considered but they mainly reflect the default risks associated with a few large corporates. Furthermore, their historical data are not sufficiently long for our purpose. Motivated by the well-known Structural default risk models (see for example Berndt et al (2005)), we

link this *collective credit risk* to two broad equity market indices, the S&P 500 and the VIX, the implied volatility index for equity options, which we use as feature variables. In addition, we have included a bond spread index, which is also indicative of Credit Risk: see Appendix A for details.

3. Thirdly, empirically, the NIM shows a certain persistence, which can be explained by the fact that banks often have large volumes of fixed-rate loans. Furthermore, banks' operational costs such as staff, properties and equipment are relatively stable between two consecutive reporting periods and the efficiency of a bank's management team and its market position, as indicated by metrics such as the banks' own credit standing, are relatively stable. Our feature selection needs to reflect this persistence of bank incomes, and we have done so by including a 1-period lagged Autoregressive term of NIM as a feature, for which Appendix B.2 provides a justification in the context of linear models, by examining the autocorrelations. We recognize that for the non-linear models such an argument is not conclusive and that, ultimately, the question of how many lags have to be included can only be answered empirically, for each algorithm separately. We note that basically all post-crisis NIM literature only use AR(1), without further justification[7].

1.3 *Literature, Research Gaps and Contributions*

NIM and Forecasting Literature

Based on their forecast objectives, the NIM literature can be divided into papers focused on forecasting NIM for a specific bank using bank-specific NIM models and those on forecasting NIM at banking-industry level using Aggregate NIM models. In this section, we review the techniques used by both types of literature so that we can include them in our study of Bank-specific NIM forecasting.

The paper by Ho and Saunders (1981) is the first studying Bank-specific NIM using classic Multiple Linear Regression techniques, including both cross sectional and time series regression, to try and understand the determining factors of a bank's NIM, by linking this NIM to a number of microeconomic and macroeconomic variables.

As for Aggregate NIM, Grover and McCracken (2014) find support for its predictability, using Factor-based NIM models via Principal Component Regression. Another Aggregate NIM study is the one of Covas, Rump and Zakrajsek (2014), who present a Fixed-Effect Quantile Autoregressive Regression (FE-QAR) model which significantly outperforms its Fixed-Effect Linear counterpart. Contrary to other papers on NIM, they investigate a non-linear model by forecasting the conditional density for NIM instead of the conditional mean as done in linear forecast models, but their study differs from ours in following respects: first, the modelling object is the NIM for a representative or "average" bank (which is still aggregate NIM) and the approach taken is a top-down approach by assuming the same set of feature variables across banks to forecast the NIM for a representative bank while, in this study, we take bottom-up approach to forecast the NIM for a specific bank, which allows for bank-specific feature selections and potentially leads to more accurate forecasting; second, they use the Linear Fixed Effect model as the benchmark (against which forecasting performance of other models has to be tested) while we will use the Random Walk, the same benchmark-model shared by most of the literature including two important NIM studies on Aggregate NIM to be discussed next. Similar to the above authors, Kupiec's study focused on forecasting a representative bank's total income calculated as the average total income across all banks in sample; in that sense, it is a top-down Aggregate NIM study. We note that, Kupiec's is the only study on bank-income forecast which includes a Machine Learning technique, specifically, the so-called least absolute shrinkage and selection or Lasso for its feature variable selection (see Hastie, Tibshirani and Friedman), which, however, is a technique for variable selection rather than a forecasting technique; in addition, the target variable in Kupiec's is bank's total income rather than Bank-specific NIM, on which we choose to focus for reasons discussed above.

We next turn to the two post-crisis Aggregate NIM studies of Guerrieri and Welch (2012) and Bolotnyy, Edge and Guerrieri (2015). Using RMSE as the forecast accuracy metric, both papers found that the forecast accuracies of their Linear Regression models, which included an AR(1) term for NIM as one of its explanatory variables, were indistinguishable from those of the random-walk benchmark. As a result, both papers raised serious concerns about the relevance and effectiveness of stress tests, which we referred to

[7] Hirtle *et al*, Kupiec and two other NIM literature use AR(1) for their bank-income forecast models.

as the *predictability or forecast accuracy Problem for the Aggregate NIM*. As regards feature variable selections, Bolotnyy, Edge and Guerrieri used only treasury-yield variables as features, whereas Guerrieri and Welch included a wide range of macroeconomic variables. We note in passing that interest-rate only variables may not be very useful in a flat-rate environment such as the one prevailing after the 2008-09 crisis. As regards the data samples, the main results from Bolotnyy, Edge and Guerrieri's are based on data sample ranged from 1989Q4 until 2008Q3, and thus did not completely cover the financial crisis up till its end in 2009Q2, particularly, the post-crisis expansionary periods[8], which prevents their results from being extrapolatable to forecasting NIM for post-crisis expansionary periods. Similarly, the Aggregate NIM study conducted by Guerrieri and Welch was based on data up to 2009Q4.

In contrast with the above two NIM studies, which used rolling-origin forecasting, Kupiec's study, however, was based on fixed-origin forecasting; thus, the assessment of his forecast accuracy using RMSE was based on only 12 sample points for forecast errors while assuming constant forecast errors across forecasting horizons. The training set used in Kupiec's study included data from March 1993 to June 2008, but excluded the Lehman's default (15 September 2008), its ensuing periods and the post-crisis expansionary periods; its out-of-sample test was based on the 12 quarters following June-2008. In our view, the sample size seems to be small and rolling-origin forecasting is desirable.

Regarding multi-step forecasting techniques, Bolotnyy, Edge and Guerrieri found that the *iterated* multi-step approach outperforms the *direct* multi-step one. This is in line with other papers from the forecasting Literature such as Macellino, Stock and Watson (2005), who compared the forecast performances of iterated multi-step and direct Multi-step forecasting using 170 different time series of US macroeconomic variable data, finding that the former outperforms the latter although the former is more vulnerable to misspecification errors.

Regarding the empirical comparisons of Machine Learning algorithms for forecasting purposes, we mention the paper by Ahmed, Gayar and El-Shishiny (2010), who found the following ranking in terms of forecast accuracy for the three algorithms studied in this paper: Neural Network, Gaussian Processes Regression and Support Vector Regression. We note however that their empirical study was restricted to Single-step forecasting only and was not related to banking.

We finally comment on the issue of forecast accuracy metrics. Based on the study of 90 annual and 191 quarterly economic time series data, Armstrong and Collopy (1992) found that the RMSE as performance criterion is not reliable for comparing forecasting performances for different time series of different scales. As an alternative, they propose to use the so-called *Rel*ative RMSE as a measure of forecasting errors. Hyndman and Koehler (1992) proposed Mean Absolute Squared Error (MASE) as a general measure for forecasting accuracy. A more recent study by Chen, Twycross and Garibaldi (2017) proposed an alternative error metric called the Unscaled Mean Bounded Relative Absolute Error or *UMBRAE*, and compared a long list of metrics including the above ones, finding from their empirical comparison study that UMBRAE is superior among a list of error measures for Time Series forecasting; see further discussion in section 2.

Research Gaps

Following the preceding literature survey, we identified the following research gaps which motivated our study in this paper:

1. Literature on stress-test have raised various concerns about the accuracies of the forecasting models for bank income used in these tests, and in particular about the accuracy of Aggregate NIM forecasts; however, despite the fact that the stated objective by regulators is to gain confidence of investors and creditors in the banks and financial system, no literature is available in public domain on the accuracies of NIM forecast models used by banks for stress-tests. Without such information, the regulators and banks may leave investors and creditors in doubt about stress-test results and even existing confidence in them could be diluted, which might explain the failure of some stress-test disclosures to receive significant market reactions as highlighted in some studies.

2. As regards the modelling techniques, existing NIM literature only use classical Linear Regression

[8] The US National Bureau of Economic Research shows that its economy resumes its expansion from June 2009, 1 month after the stress-test. `https://bit.ly/1IJjPDM`.

techniques in their forecasting models although non-linear Machine Learning techniques can be used to construct time-series forecast models and might bring out new insights.

3. Empirically, NIM data are known to be non-Gaussian and heavy-tailed; the results from existing literature are typically based on Linear regression models, which are typically estimated by Ordinary Least Square methods; LS estimators are known to be vulnerable to outliers. Alternative regression techniques might bring new insights or even lead to different results to those from existing literature.

4. With regard to the data sampling, post-crisis bank-income studies cited above only covered part of the latest economic cycle without including the economically expansionary period after the financial crisis; since now we have a bigger data sample, increased forecasting accuracy based on out-of-sample test from bigger sample size is possible.

5. Using the RSME to compare the forecasting performances of different models applied to different banks with varying data scales can be problematic. Arguably, the use of scale-independent forecast accuracy measures such as the Rel-RMSE or the UMBRAE (see 2.11) is more appropriate for such situations and may lead to different conclusions.

6. As a time-series forecasting study, ours include non-linear Machine Learning techniques, which might bring out insights about time series forecasting performance comparison. Except for one study, which only examined Single-step prediction for non-financial data, existing empirical performance-comparison literature rarely include any models from the Machine Learning area. Such models may be serious competitors to the classical models (and we will indeed find this to be the case for at least one class of ML-models).

Contributions of the paper

The paper makes the following contributions to the existing literature:

1. It is the first systematic study on the forecast accuracy problem for Bank-specific NIM. As indicated in section 3, in contrast with the problem raised in the literature about Aggregate NIM, we find that Bank-specific NIM can be predicted with much better accuracy than does the random-walk benchmark by both linear and non-linear regression models, which suggests that Stress Test should not be discredited simply because of the forecast accuracy problem with Aggregate NIM.

2. We show that non-linear Machine Learning regression techniques achieve superior forecast accuracies, notably the NARX version of RNN (cf. section 2), followed by the Gaussian Process Regression (or GPR) and Support Vector Regression (or SVR); the rank-ordered forecast accuracies in our study as shown in Section 3 are in line with those reported in a separate study by Ahmed, Gayar and El-Shishiny (2010) as discussed in section 1.3; however, our studies are conducted in the context of banking data and of Multi-step forecasting.

3. Within Recurrent Neural Network, Bayesian Neural Network achieved the best performances out of different backpropgations, which is consistent with findings from the literature, such as Zhang, Patuwo and Hu (2003).

4. The study represents an application of Robust Regression in both Linear and Factor-based regression models; in presence of outliers in NIM data, Robust Regression is shown to significantly improve forecasting results, as indicated by the performance improvements in Section 3 shown by MLR vs ML, PCR-MLR vs PCR-ML.

5. The paper is an example of applying the following forecasting techniques: multi-step forecasting involving rolling origins, model recalibration, Leave-one-out cross validation for time series modelling, choice of forecast accuracy metrics between scale-dependent RMSE and scale-independent metrics such as UNMBRAE, RelRMSE, etc.

6. Feature Selections can not only contribute to boost forecasting performances but also contribute to link modelling results to sound economic principles as explained in Section 1.2.

7. As for policy suggestions, we recommend that regulators take steps to disclose forecast accuracy information based on literature-recommended forecasting techniques such as out-of-sample tests, rolling-origin forecasting, model recalibration as well as non-linear Machine Learning techniques as part of the forecast models used by banks and regulators themselves, which should contribute to improve confidence on part of investors and creditors in financial markets and to maintain the financial stability. Furthermore, banks should be encouraged to conduct research into forecast models at more granular levels such as asset classes, which will help regulators and banks to identify the vulnerabilities in business areas in G-SIBs at build-up stages before they transform into a crisis.

The rest of the paper is organized as follows:

In section 2, we briefly review linear regression and Machine Learning regression techniques, which we have used to investigate potential forecast accuracy problem with Bank-specific NIM. Section 3 summarizes our empirical results; we do an inter-model comparison, and identify the best performing ones, which is followed by discussions about the empirical results for each model type separately and the conclusions which we believe can be drawn from this paper's results. Three appendices, A, B and C, respectively list the precise feature selections, present data description and summary statistics for the NIM as well as a collection of the different figures and tables which form the basis of the discussion for model-specific results presented in section 3.

2 Regression models via ML-techniques

Classical statistics include a wide range of regression techniques for forecasting; in addition, modern Machine Learning arena also provides regression models that are based on Machine Learning techniques. In our study, we follow the literature (see Hastie, Tibshirani and Friedman, 2008) by referring to both as Machine Learning Techniques or simply *Machine Learning-regression algorithms* to include all regression models of distinct types; we reserve *ML model* for representing Multiple Linear regression models only. As mentioned above, our objective is to forecast the multi-step ahead NIM values for a given bank in response to a set of exogenously given economic scenarios in the context of stress-test; thus, a multi-step time series forecast model as indicated by equation (3) needs to be constructed, trained before being used to forecast NIM. Given that a wide range of model choices and feature variable selections are available, we need to start with a Random Walk benchmark, which is a practice shared by two of the post-crisis NIM forecasting literature discussed above and advocated by forecasting literature in order to compare forecast accuracy amongst competing models based on out-of-sample tests; see a discussion by Hyndman[9].

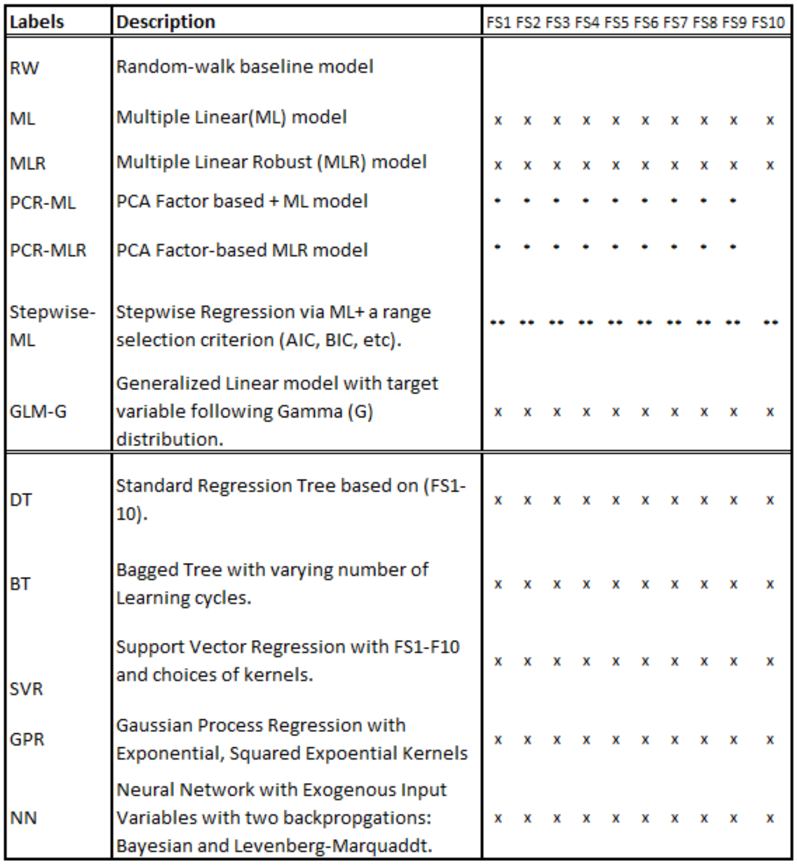
In this section, we will present a brief introduction for each of the Machine Learning-regression algorithms used in our investigation. Table 1 summarizes some of the Machine Learning-regression algorithms that we have tested using a list of so-called standard feature variable selections abbreviated as FS1-FS10 (marked by an "x"), for which detailed descriptions are available in Appendix A.1; in addition, we present a list of feature selections in Appendix A.2, which we investigated but found to achieve lower accuracies than the standard ones. Excluding the random-walk model (labelled as "RW"), we tested a total of 162 models for predicting bank-specific NIM from 11 different Machine Learning-regression algorithms, including 108 models in Table 1 and additional 54 models in Table 5 in Appendix A.2. In addition to their individual sections below, Appendix A.3 and **??** contain more details about Principal Component Regression and Stepwise Multiple Linear Regression respectively. Regarding the types of *ML-Regression algorithms* investigated in the study, we choose popular ML-Regression ones based on a survey of NIM literature (*cf.* Section 1.3), which is not meant to be exhaustive.

2.1 *Random Walk: the benchmark model*

We follow common practice and two prominent NIM forecasting literature (*cf.* section 1.3) by adopting the Random-walk model as our benchmark model so that we can compare our forecast performances with those

[9] Benchmarks for Forecasting: `https://bit.ly/2QgkF5R`.

Table 1: 108 Regression Models with 10 Feature Selections; see further 54 models in Appendix A.2.



Labels	Description	FS1	FS2	FS3	FS4	FS5	FS6	FS7	FS8	FS9	FS10
RW	Random-walk baseline model										
ML	Multiple Linear(ML) model	x	x	x	x	x	x	x	x	x	x
MLR	Multiple Linear Robust (MLR) model	x	x	x	x	x	x	x	x	x	x
PCR-ML	PCA Factor based + ML model	•	•	•	•	•	•	•	•	•	
PCR-MLR	PCA Factor-based MLR model	•	•	•	•	•	•	•	•	•	
Stepwise-ML	Stepwise Regression via ML+ a range selection criterion (AIC, BIC, etc).	••	••	••	••	••	••	••	••	••	••
GLM-G	Generalized Linear model with target variable following Gamma (G) distribution.	x	x	x	x	x	x	x	x	x	x
DT	Standard Regression Tree based on (FS1-10).	x	x	x	x	x	x	x	x	x	x
BT	Bagged Tree with varying number of Learning cycles.	x	x	x	x	x	x	x	x	x	x
SVR	Support Vector Regression with FS1-F10 and choices of kernels.	x	x	x	x	x	x	x	x	x	x
GPR	Gaussian Process Regression with Exponential, Squared Expoential Kernels	x	x	x	x	x	x	x	x	x	x
NN	Neural Network with Exogenous Input Variables with two backpropgations: Bayesian and Levenberg-Marquaddt.	x	x	x	x	x	x	x	x	x	x

in the literature using the same benchmark. The random-walk model provides a naïve but often hard-to-beat benchmark which is popular in financial forecast literature; see an attempt by Kilian and Taylor (2003) to provide an explanation for why random-walk is hard-to-beat benchmark in forecasting exchange rates.

Denoting the information set at time t by I_t, the h-step-ahead forecasted NIM can be expressed as $E[y_{t+h}|I_t]$ or simply $E_t[y_{t+h}]$, leaving the information set understood. If we assume that the y_t follows a simple Random Walk model, with independent and identically distributed increments $y_{t+1} - y_t$, $t = 0, 1, \ldots$, then the multi-step forecast simply becomes

$$\widehat{y}_{t+h|t} := E_t[y_{t+h}] = y_t, \tag{4}$$

for all $h > 0$. This just says that the best estimate for the NIM at time $t + h$ is its value at time t. Random Walk based forecasting is also called the ”No-change” model, for obvious reasons.

2.2 *Multiple Linear & Multiple Linear Robust Regressions*

For the rest of the paper, to simplify the notations, we index time by i instead of t; for quarterly data (if not, quarterly averages are taken), $i = 1, \cdots, n$ means that data are sampled from $Q_1, Q_2, \cdots, Q_n$. The general Linear Regression model takes f in equation (3) to be an affine function of y_{i-1} and X_i, respectively representing the AR(1) term for NIM and exogenously given economic variables. As ML-model is familiar to most people, we now formally define some forecasting techniques in the context of a ML-model.

For reasons discussed above, we conduct rolling-origin forecasting, which starts with splitting our time series data, which ends at time period indexed by n (e.g., Q_n for the last quarter), into a series of training sets based on a rolling window with a fixed length $\ell < n$. As discussed, the ending period (e.g., ending quarter) of one training set is referred to as the forecasting origin indexed by r (e.g., for the first training set, $r = \ell$), where $r = \ell, \cdots, R$ and the rolling window rolls until $R = n - H$ in order to leave H number of observations for the out-of-sample tests at the last roll.

In our study, our data set are from $2000Q_1$ to $2016Q_4$ inclusive, thus, $n = 68$; our rolling window has a fixed lengths $\ell = 36$, leaving last $H = 8$[10] quarters for out-of-sample test, so $R = 60$. $\forall r, r = \ell, \cdots, R$, altogether the number of rolls is 25, leading to the construction of 25 training sets denoted by D^{T_r}. We choose $\ell = 36$ such that each training set contains the quarter, i.e., 2008Q3, when Lehman's default took place, which is desired.

For each training set D^{T_r} with $m = \ell$ observations, we train a model as

$$y_i = \alpha y_{i-1} + \sum_{j=1}^{d} \beta_j x_{i,j} + \beta_0 + \epsilon_i, \qquad i = 1, \ldots, m \tag{5}$$

where the $x_{i,j}$, $j = 1, \ldots d$, are the components of the d-dimensional exogenous feature vector $\mathcal{X}_i$, α and the β_j are the coefficients which are to be learned from the training set D^{T_r} for all r; ϵ_i stands for zero-mean distributed error term at time i. We note that ϵ_i is the in-sample training error rather than the desired forecast error, which we calculate as the gap between the forecasted h-specific NIM $\widehat{y_{i+h}}$ ex-ante and the observed h-specific NIM ex-post: $\epsilon_h = \widehat{y_{i+h}} - y_{i+h}$. Corresponding with each training set indexed by r, we have a set of h-specific ϵ_h. We collect ϵ_h from all training sets to arrive at a sample distribution of ϵ_h, thus, the basis for *forecast accuracy*.

As discussed above, after each roll, we recalibrate regression model to pick up signal from refreshed data; furthermore, for $h > 1$, we plug forecasted $\widehat{y_{i-1}}$ into equation (5) iteratively as discussed further below.

2.2.1 Multiple Linear Regression or ML

Continuing with the notations above for a training set D^{T_r} of sample size ℓ, we further introduce the k-dimensional feature vector (where $k = d + 2$), denoted by $\xi_i := (y_{i-1}, x_{i,1}, \ldots, x_{i,d}, 1)$ and associated k-dimensional parameter vector, in post-transposition form, denoted by $b^T := (\alpha, \beta_1, \ldots, \beta_d, \beta_0)$, the Linear Regression model can be written as

$$y_i = \xi_i b + \epsilon_i, \qquad i = 1, \ldots, n,$$

where y_i stands for the data point i observed for the target variable and ϵ_i for the error term $\forall i \in D^{T_r}$; in particular, to distinguish from the total sample size n, we denote the sample size for D^{T_r} by m, where $m = \ell$. The classical Least Squares or LS estimator for b is then given by

$$\widehat{b} := \widehat{b}(D^{T_r}) := (\Xi^T \Xi)^{-1} \Xi^T y, \tag{6}$$

where y stands for the vector $(y_1, \cdots, y_n)$ of observed NIM; Ξ is the $\ell \times k$-matrix made up of the components $\xi_{i,j}$ $(j = 1, \cdots, k)$ of the vectors ξ_i. When conducting Multi-step ahead NIM forecasting, under *Direct Multi-step forecast*, one trains h-specific NIM model for each h, thus, $h = 8$ in our study, we need to train 8 regression models, from each of which we forecast h-specific NIM; whereas, under *Iterated forecast*, one train one model and forecast 1-step ahead NIM, which is then plugged into the trained model to forecast the 2-step ahead NIM. In the training, we also recalibrate the parameters in our study based on refreshed data as mentioned above.

This Least Squares estimator is known to be BLUE (Best Linear Unbiased Estimator), in the sense that when the estimation errors can be assumed to be uncorrelated and homoscedastic, it minimizes the variance of the estimation error amongst all linear estimators, by the Gauss-Markov theorem: see for example Greene (1997). When the errors are i.i.d and normal, it coincides with the Maximum Likelihood Estimator. However, for finite samples it is sensitive to large outliers, an issue that we can address by using Robust Regression.

In our study of NIM forecasting models, we examine 10 Linear Regression models with the Standard Feature Selections labelled by FS1-FS10 in Table 1; in addition, we looked into 6 other feature selections, which are labelled by $A1 - A6$ in Table 5 and which correspond with the quadratic terms of FS5-FS10 respectively as described in details in Appendix A.2. Altogether, we investigated 16 ML models for studying forecast accuracy by benchmarking against the random-walk baseline.

[10] We choose $H = 8$ for convenience; the choice of $H = 9$ in practice does not affect our results.

2.2.2 Multiple Linear Robust Regression Model or MLR

As highlighted in literature (e.g. Fox and Weisburg, 2013), the LS estimator can behave badly when error distributions are not normal, particularly when the probability distribution of the errors is heavy-tailed. Outliers can be informally defined as data points which are inconsistent with a (presumed) general trend. Outlying feature- or dependent variables can significantly influence the LS estimator, particularly for finite samples of moderate size, since residuals are equally weighted and squared. In the presence of outliers, one strategy is to remove them entirely from the sample, but this requires some criterion of when a data point can be considered to be an outlier. Another strategy is to use Robust Regression with the aim to down-weight the influences of possible outliers. Huber (1964) introduced a general so-called *M*-estimator (owing to its similarity with Maximum Likelihood Estimation or MLE), which includes the LS estimator as a special case. In our study, we apply Robust Regression to the Linear Regression model, to which we refer as the MLR Model.

If we define the residual term e_i for data point i as $e_i := y_i - \xi_i\widehat{b}$, the general *M*-estimator proposed by Huber minimizes a loss function $L := \sum_{i=1}^{n} \rho(e_i)$ across all n data points for some given function ρ. In particular, the LS estimator becomes a special case of the *M*-estimator by setting $\rho(e_i) = e_i^2$. The function $\rho(u)$ should have the following properties:

- ρ is twice continuously differentiable, symmetric: $\rho(-u) = \rho(u)$, and strictly increasing when $u > 0$ with $\rho(0) = 0$.
- In particular, $\rho(u)$ has a unique global minimum equal to 0 when $u = 0$, and its derivative $\phi(u) := \frac{\partial \rho}{\partial u}(u) > 0$ when $u > 0$ while $\phi(u) < 0$ when $u < 0$.

Clearly, LS estimator satisfies the above properties, we can minimize $L = \sum_i (y_i - \xi_i b)^2$ by setting its partial derivatives with respect to each component of b equal to 0, which leads to the system of k equations (where k coincides with the dimension of parameter vector b, in particular, $k = d + 2$ as indicated above):

$$\sum_{i=1}^{m} \phi(e_i)\xi_{i,j} = \mathbf{0} \qquad\qquad \text{for } j = 1, \ldots, k. \tag{7}$$

Introducing weights w_i by $w_i := \frac{\phi(e_i)}{e_i}$ if $e_i \neq 0$ while setting $w_i = \rho''(0)$ if $e_i = 0$, we can rewrite this system as

$$\sum_{i=1}^{m} (w_i e_i)\, \xi_{i,j} = \sum_{i=1}^{n} w_i (y_i - \xi_i b) \xi_{i,j} = \mathbf{0} \qquad\qquad \text{for } j = 1, \ldots, k. \tag{8}$$

Note that, on account of the sign of $\phi(u)$ for positive and negative u, $\phi(u)/u > 0$, the weights are positive.

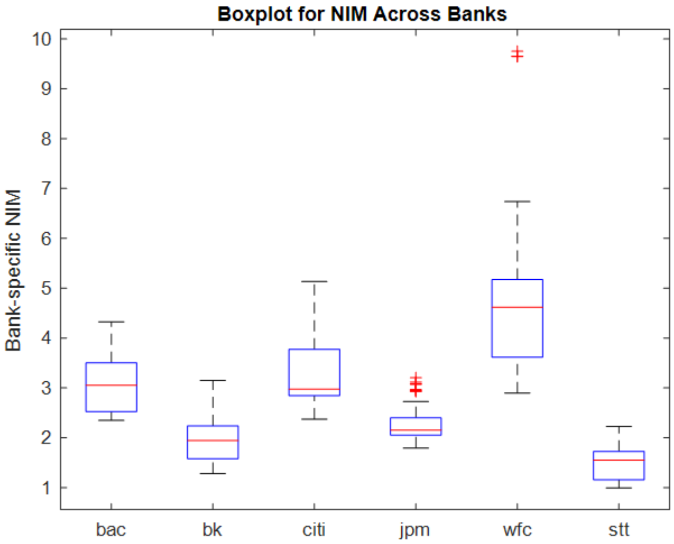
After some manipulations and letting W be the diagonal matrix with the weights w_i on the diagonal, equation (8) is equivalent to

$$b = [\Xi^T W \Xi]^{-1} \Xi^T W y, \tag{9}$$

where Ξ is the $n \times k$-matrix $(\xi_{i,j})$ and y is the response vector $y = (y_1, \ldots, y_n)$. Equation (9) resembles the solution of a Weighted LS estimator problem minimizing $L = \sum (w_i e_i)^2$, but note that the matrix W depends on the residuals e_i, and therefore on b, so this is actually a system of non-linear equations for the components of b. This system can be solved iteratively as follows: let us define $w(u) := \phi(u)/u$ for $u \neq 0$, and $w(0) = \phi'(0)$, so that $w_i = w(e_i)$. Then:

1. Initialize $b = b^0$ by taking b^0 the LS estimator (6).
2. Using b^0, compute the regression errors $e_i^0 = y_i - \xi_i b^0$ and from that the weight matrix $W^0 := \mathrm{diag}(w(e_i^0))$ and $b^1 := [\Xi^T W^0 \Xi]^{-1} \Xi^T W^0 y$.
3. Iteratively, for $\nu \geq 2$, given $b^{\nu-1}$,
    - For $i = 1, \ldots, m$, compute $w_i^{\nu-1} := w(e_i^{\nu-1})$ where $e_i^{\nu-1} := y_i - \xi_i b^{\nu-1}$, and put $W^{\nu-1} = \mathrm{diag}(\, w_i^{\nu-1}\,)$.

Figure 1: Outliers for 6 Bank-specific NIMs in Data Sample



- Define b^ν by

$$b^\nu = [\Xi^T W^{\nu-1} \Xi]^{-1} \Xi^T W^{\nu-1} y. \tag{10}$$

Step 3 is repeated until meeting some tolerance criterion, at which point we set $\widehat{b} \approx b^\nu$.

The above algorithm starts with LS estimator, then iteratively weighs the error terms until the estimator converges. Consequently, it is known as the *Iteratively Reweighed Least Square or IRLS* method. For our study we used Tukey's *bi-square weight function*, defined by

$$w(e) = \begin{cases} [1 - (\frac{e}{K})^2]^2 & \text{for } |e| \leqslant K; \\ 0 & \text{for } |e| > K. \end{cases}$$

with a tuning constant $K = 4.685$, which is shown to be able to retain 95% efficiency of LS Estimator by Tukey (1977).

In Figure 1, we use boxplot to show the empirical distributions for the NIMs of the six G-SIBs investigated in our study; we note that each bank is distinguished by its stock ticker, which one can refer to Table 3 for the corresponding names associated with their stock tickers. Clearly, the NIMs for both 'jpm' and 'wfc' (indicating JP Morgan Chase and Wells Fargo banks respectively) have outliers indicated by "+" highlighted in red colours. In this exercise, we adopt the definition of *outlier* according to Tukey (1977): outliers refer to data lying outside the so-called "Tukey Fences", i.e., the range defined as $[Q_1 - k(Q_3 - Q_1), Q_3 + k(Q_3 - Q_1)]$ where Q_1 and Q_3 denote the 25^{th} and the 75^{th} quantile respectively, and k is a constant. Tukey recommends to take $k = 1.5$. Clearly, the NIM distributions for "citi", "jpm" and "wfc" are all positively skewed while only "jpm" and "wfc" have outliers.

We implement 10 MLR models corresponding with feature variables FS1-FS10, as shown in Table 1. Additional 6 models based on derived feature variables from standard feature variables can be found in Appendix A.2. As shown in our empirical section 3, the use of Robust Regression is conducive to improve the resistance against the influence on forecast accuracies from data outliers.

2.3 *Principal Component Regression*

As mentioned in Section 1.3, two Aggregate NIM studies using Principal Component (PC) Regression (or PCR) come up with different findings, probably due to, among other reasons, a different choice of feature variables: the principal components from Bolotnyy, Edge and Guerrieri's were extracted from interest-rate only variables, whereas those of Grover and McCracken's study were extracted from a diverse range of Macroeconomic variables, which might explains the divergence in their findings. One objective of our study is to investigate whether or not PCR-based Bank-specific NIM models can forecast significantly better than the Random Walk model. We also investigate the predictive performances with regard to varying the number m of retained of Principal Components or PCs in the PCR model. Brummelhuis and Luo (2019a) uses Principal Component Analysis (or simply PCA) as a tool to investigate the potential impacts of Feature Correlations on Classification for a wide range of Classifiers.

More formally, we apply PC analysis to the ML or MLR model of sections 2.2.1 and 2.2 by replacing the feature variables $\mathcal{X}_t^d$ with a certain number m of retained PCs extracted from $\mathcal{X}_t^d$: see Appendix A.3 for details. The resulting models will be denoted by PCR-ML and PCR-MLR respectively. In our study, we applied these two classes of models to NIM forecasting and investigated their forecasting performances when varying the number m of PCs between 3 to 11 (the maximum number of raw feature variables). It is well-known that for yield-curve data, PC analysis with $m = 3$ gives reasonably good results with a good interpretation for the extracted Principals. For illustration, we present in Appendix A.3 the PC components extracted from the US and the UK government yield curves based on data from 2000Q1 and 2016Q4. As indicated by our empirical results in section 3, increasing m does not necessarily lead to improved forecasting performances for PCR-based NIM Models because a larger m only means that a higher percentage of variance is explained by the retained PCs, but does not necessarily lead to higher forecasting accuracy. A similar phenomenon has been observed in Brummelhuis and Luo (2019a).

2.4 *Generalized Linear Model or GLM*

A linear model with response variable y and vector of explanatory variables ξ can be formulated as

$$\mathbb{E}(y|\xi) = \xi b,$$

where b is the vector of parameters which have to be estimated. It has the property that y is Gaussian if ξ is multi-variate Gaussian. Such a model would not be appropriate if, for example, the values of the response variable would have to be restricted to some subdomain of the real numbers. An example in Finance is the modelling of default probabilities, whose values have to lie in the interval $[0, 1]$. In such situations a Generalized Linear Model or GLM might be more suitable. For such models one assumes that y, conditional on ξ, has a distribution in some given class of distributions, and that some suitable non-linear function g of the conditional mean of y is a linear function of ξ:

$$g\left(\,\mathbb{E}(y|\xi)\,\right) = \xi b.$$

The function g, which is called the link-function of the GLM, is chosen appropriately depending on the chosen class of distributions. As for us relevant example is the GLM-Gamma model, for which $y|\xi$ is assumed to follow a Gamma-distribution,

$$y|\xi \sim \frac{1}{\theta^k \Gamma(k)} y^{k-1} e^{-y/\theta},$$

with parameters k and θ and mean is $k\theta$. The choice of link function $g(\mu) = 1/\mu$, leads to the conditional pdf:

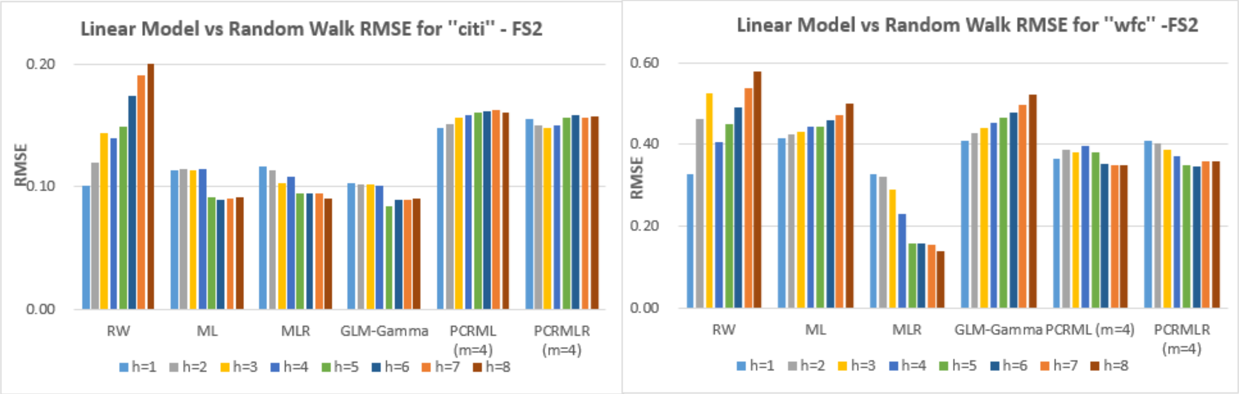
$$\frac{k^k (b^T \xi)^k}{\Gamma(k)} e^{-k(\xi b) y}.$$

Given a sample (y_i, ξ_i) and suitable distributional assumptions on ξ, one can then write down the likelihood function and determine the parameters by Maximum Likelihood, in particular the vector b (but also the k.) Forecasting is then simply done by inverting the regression equation above:

$$\mathbb{E}(y|\xi) = g^{-1}(\xi b).$$

We note in passing that for writing down the likelihood, we have to take into account the $AR(1)$-structure of our model, in which y_{i-1} occur as a component of $\xi_i = (y_{i-1}, \mathcal{X}_i, 1)$. We have tested the GLM - Gamma model for forecasting the NIM.

Figure 2: Summary for Performances by RMSE for Linear Models



2.5 *An example: Using Linear models for NIM forecasting*

We end our discussion of linear models with an illustrative example. Figure 2 presents the root mean square (forecasting) errors (RMSE) for two banks, Citi Group Bank (”citi”) and Wells Fargo Bank (”wfc”), under the five linear models, ML, MLR, PCR-ML, PCR-MLR and GLM-Gamma, alongside with the benchmark RW model. All models were trained with the feature variable selection FS2 of Appendix A, and we used 4 principal components (or $m = 4$) for the PCR models. We see that the PCR-ML and PCR-MLR did not significantly outperform the RW model; while the MLR did in both cases of ”citi” and ”wfc”, the ML only outperformed in the case of ”citi”. As shown in Figure 1, the NIM data for ”citi” have no outliers, whereas those for ”wfc” do, which suggests that the use of Robust Regression can significantly enhance forecasting performance in the presence of outliers.

The choice of the RMSE as a metric to compare forecasting performances for different banks could be criticized on the grounds that it is scale-dependent. Figure 1, however, shows that the NIMs of ”citi” and ”wfc” are of comparable sizes. We have nevertheless also used scale-independent error measures for this study, which will be further discussed in section 2.11 below.

2.6 *Stepwise Regression*

Stepwise Regression is a technique for selecting the ”best” subset of feature variables out of a list of ”candidate” feature variables (and their transformations) based on a pre-specified criterion as a result of training a sequence of regression models. Stepwise Regression can lead to different choices of Feature Selections depending on the choices of criteria; in this study, we explore the choices of criteria such as Bayesian Information Criterion (BIC), Akaike information criterion (or AIC) and Sum of Squared Errors (or SSE), for which definitions are available from Hastie, Tibshirani and Friedman.

In this study, we focus on using Stepwise Regression in combination with ML model from the list of Feature Selections available in Appendix **??**. Stepwise Regression can be used in combination with GLM and other models but it is not in our scope. Therefore, although we present the performance for Stepwise Regression together with the rest of other models in Section 3, the fair performance comparison that one should conduct is between Stepwise and ML models because both are based on the same modelling set-up, i.e., ML.

2.7 *Tree-based Regression Models*

From this section onward, we turn to the non-linear Machine Learning models that we have used for forecasting the NIM, starting with the Regression Tree model and its Ensemble version, the Bootstrap Aggregating, or Bagged Regression Trees. As a general comment, due to sample size constraints, the parameters of the different non-linear regression models are in this paper tuned by trial and error rather than by using K-fold Cross Validation as we did in Brummelhuis and Luo (2019a), our data only being quarterly.

2.7.1 Regression Tree Models

Compared with the Linear Models discussed before, the Regression Tree model constructs a piecewise constant predictor function in which at each node of a decision tree, a constant sample mean of the target variable is fitted. Classification and Regression Tree or CART algorithms were introduced by Breiman, Friedman and Stone (1984). Classification Trees were for example used in Brummelhuis et al (2019a) for the construction of proxy CDS Rates.

In this paper, we adopt the Binary Regression Tree, in which the objective is to minimize the Sum of Squared (Forecasting) Errors or SSE, where the Forecasting Error is defined as the difference between the observed NIM and forecasted NIM within different partitioned regions of Feature Space. The forecasted NIM for each region will simply be the mean of the target variable y_i for the data whose feature variables lie in that region. The different regions are rectangular boxes in feature space with faces parallel to the axes, and are obtained recursively by performing a succession of *splits*: given a feature variable, we can split our set of training data into two subsets, one where the chosen feature variable is smaller than some number c, and one where it is bigger. Only finitely many such splits are possible, since the set of training data is finite. Calling the two subsets L_s and R_s, where the subscript s indicates the split, the SSE associated to the split s is defined by

$$\epsilon_s = \sum_{i \in L_s} (y_i - \mu_{L_s})^2 + \sum_{i \in R_s} (y_i - \mu_{R_s})^2, \tag{11}$$

where $\mu_{L_s} = (\#L_s)^{-1} \sum_{i \in L_s} y_i$ and similarly for μ_{R_s}. We then construct our tree by using a "greedy" search algorithm, starting off with the entire training set at the base node, with its associated mean and SSE, and recursively choosing splits which minimize the total SSE, creating two new nodes of the tree for each such split. The algorithm stops splitting if the resulting tree meets certain constraints, such as the SSE becoming smaller than some pre-assigned number or by imposing a constraint on the maximum number of splits. This is known as *Standard CART*. There are two other variants, *Curvature Test CART* and *Interactive Test CART*, which we briefly describe.

Standard CART treats all predictor splitting homogeneously in terms of their influences on the target variable. The *Curvature Test* and the *Interactive Test*, as proposed by Loh (2002), allow users to select a subset of feature variables based on (Pearson) Chi-square tests while respectively taking into account the heterogeneous Feature Influence on the target variable and the interactions between feature variables when they choose predictor splitting.

For this study, we tested a total of 10 Standard Regression Tree models, one for each Feature Selection (FS1-FS10) and implemented six additional models with feature selections FS5-FS10 to investigate heterogeneity of feature variable influence and its impact on NIM prediction, using the *Curvature Test* and the *Interaction Test* described above. Table 2 presents an illustrative example for NIM forecasting for the Bank of America (bac) using Standard CART with Feature Selection FS1, where we refer Appendix A.1 for the notations used.

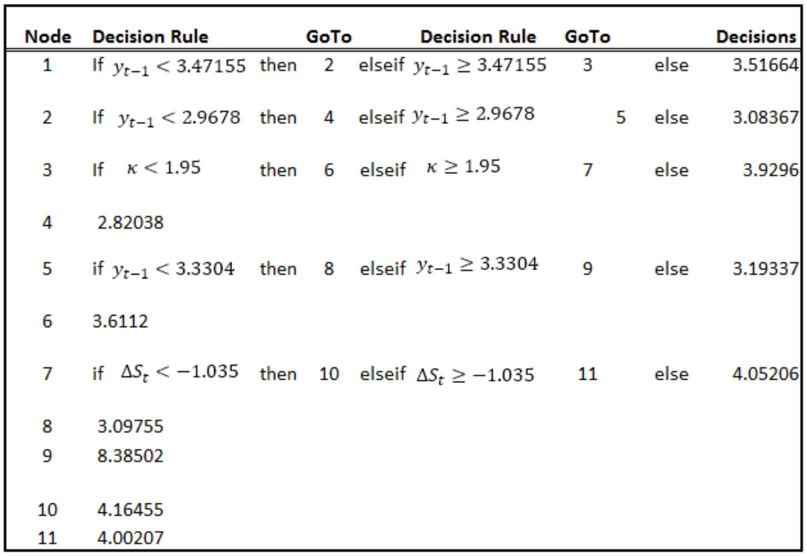
It is well-known that the Regression Tree is a weak learner, in the sense that it tends to over-fit a given set of training data, and that its forecasting performance on other data sets then is not very good. This is sometimes called the generalization problem. With the hope to improve the robustness of a Regression Tree's forecasts we use Ensemble learning, where we train an ensemble of Regression Trees instead of a single one.

2.7.2 Ensemble / Bagged Regression Trees or B-Tree

Ensemble Learning is a general Machine Learning philosophy in which, using techniques such as Bootstrapped Aggregation (or Bagging), Boosting or Random Forest, one combines the outcomes from several weak learners to arrive at a final learning outcome which is expected to be stable in the sense of reducing the variance. We focus on Bagged Regression Trees and refer the interested reader to Hastie, Tibshirani and Friedman for a detailed discussion of the other two techniques, which we haven't employed for this study.

The Bagged Regression Tree algorithm runs as follows:

Table 2: A Simple Illustrative Example for Regression Tree results ('bac' learned under FS1)



Node	Decision Rule		GoTo	Decision Rule	GoTo		Decisions
1	If $y_{t-1} < 3.47155$	then	2	elseif $y_{t-1} \geq 3.47155$	3	else	3.51664
2	If $y_{t-1} < 2.9678$	then	4	elseif $y_{t-1} \geq 2.9678$	5	else	3.08367
3	If $\kappa < 1.95$	then	6	elseif $\kappa \geq 1.95$	7	else	3.9296
4	2.82038						
5	if $y_{t-1} < 3.3304$	then	8	elseif $y_{t-1} \geq 3.3304$	9	else	3.19337
6	3.6112						
7	if $\Delta S_t < -1.035$	then	10	elseif $\Delta S_t \geq -1.035$	11	else	4.05206
8	3.09755						
9	8.38502						
10	4.16455						
11	4.00207						

1. First, we draw random samples with replacement from the training data D^T to create a new training sets by D_b^{T*}, to which is often referred as a bag, where $b = 1, \ldots, B$ and each D_b^{T*} can have some rows repeated or some rows missing from D^T.
2. Second, for each $b = 1, \ldots, B$ we train a Regression Tree based on D_b^{T*}, thereby obtaining a predicted value denoted as $\widehat{y_b^*}$ associated with bag b.
3. Finally, we define the forecasted value for the Bagged Regression Tree as the average of the forecasted values from B bags.
$$\widehat{y} = \frac{1}{B}\sum_{b=1}^{B} \widehat{y_b^*} \tag{12}$$

In our study, we use the Bagged Trees (or called B-Tree) by growing Regression Trees as described in Section 2.7.1. The motivation to use Bagged Regression Trees is to minimize the variance of prediction errors in least square sense across different Regression Trees and mitigate the so-called generalization problem associated with single Decision Tree.

2.8 *Support Vector Regression or SVR*

Support Vector Machines (SVM), introduced by Cortes and Vapnik (1995), can be used for both classification and regression. For a two-class classification problem with linearly separable data, the basic idea is to construct a separating hyperplane which is at maximum distance of both classes, equivalently, which creates a maximal margin, to ensure stability with respect to generalization. The same idea underlies SV Regression: see for example Schölkopf and Smola (1998). In linear ε-SV regression one looks for a linear function $f(x) = \beta^T x + \beta_0$ such that $|y_i - f(x_i)| \leq \varepsilon$, for all pairs (x_i, y_i) in the training set and such that the distances of the (x_i, y_i) to the hyperplane defined by f are as large as possible. Since this distance can be shown to be equal to $|y_i - (\beta^T x_i + \beta_0)| \,/\, \sqrt{1 + ||\beta||^2}$ with $||\beta||^2 = \beta^T \beta$ the Euclidean norm, and the numerator is bounded by ε, one therefore wants to minimize $||\beta||^2$. This leads to a constrained minimization problem. In the literature one often encounters the phrase that one is looking for a linear function f which is "as flat as possible", a terminology which may be confusing (since any hyperplane is after all flat) unless this is interpreted as "as horizontal as possible". Another motivation for wanting to have $||\beta||$ as small as possible is to avoid situations where a component of β dominates all others, similar to Ridge Regression (see Hastie, Tibshirani and Friedman).

As it stands, this problem may not be feasible, for there may not exist a (β, β_0) such that the constraints $|y_i - f(x_i)| \leq \varepsilon$ are satisfied for all elements of our training set. We therefore allow, as for the "soft-margin" version of the SVM classifier, that these constraints can be violated but at a cost proportional to the size of the violation. This leads to the following constrained minimization problem (Vapnik, 1995):

$$\begin{cases} \min_{\boldsymbol{\beta}} \frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ \text{subject to} \\ \forall i : y_i - \boldsymbol{\beta}^T x_i + \beta_0) \leqslant \epsilon + \xi_i, \\ \forall i : (\boldsymbol{\beta}^T x_i^T + \beta_0) - y_i \leqslant \epsilon + \xi_i^*, \\ \forall i : \xi_i^* \geqslant 0, \\ \forall i : \xi_i \geqslant 0. \end{cases} \tag{13}$$

In Non-linear SV Regression one first transforms the feature variables via an invertible non-linear map $\varphi(x)$ of feature space to some higher-dimensional space, and then performs a linear SV Regression on the transformed data points $(\varphi(x_i), y_i)$. It turns out that the algorithm for linear regression can be formulated completely in terms of the scalar products $x_i^T x_j$, and we therefore only need $k(x_i, x_j)$, where $k(x, z) := \varphi(x)^T\varphi(z)$, the so-called kernel function. The function $k(x, z)$ is positive definite, and any such a positive definite function comes from a suitable transformation φ, by a mathematical result known as Mercer's theorem. It therefore suffices to specify k and keep the underlying transformation φ implicit. Examples of positive definite k's are Gaussian and homogeneous polynomials in $x^T z$. For both Linear and Non-linear SVR we only need to solve a quadratic optimization problem, for which numerical solvers are readily available from a variety of statistical or numerical computational packages.

In our study we set the parameter C in (13) equal to $C := \frac{IQR(y)}{1.3490}$, where $IQR(y)$ stands for the Inter-quartile range[11] or IQR of the y_i, and 1.3490 is the IQR of the standard normal distribution. The loss-tolerance level ϵ was set equal to one-tenth of the standardized IQR $\epsilon := \frac{IQR(y)}{13.490}$. We used the Sequential Minimal Optimization algorithm or SMO, of Platt (1998), which is provided in standard software, as our solver. We experimented with two different Kernel functions, the Gaussian, $k(x, y) = e^{-\|x-y\|^2/\sigma^2}$, and the Linear kernel, $k(x, y) = \langle x, y\rangle$ and used the kernel bandwidth σ chosen heuristically based on the median distance between a training point and its nearest neighbour, as described in Kim and Scott (2012).

2.9 *Gaussian Process Regression or GPR*

Given a training set $D^T = \{(x_i, y_i) : i = 1, \cdots, n\}$ of data, we want to predict the values y_i^* for a set of feature variables $\{x_i^*\}$ different from the x_i's which appear in the training set. The basic idea of Gaussian Process Regression or GPR, which first appeared in Danie G. Krige's master thesis in the area of Geostatistics (which is why GPR is also called Kriging[12], and was subsequently formalized by the French mathematician George Matheron (1963) is that, rather than doing this by fitting some parametric function f to the data set D^T and subsequently using that function for prediction, we put an appropriate probability measure on the set of all functions, and predict the new values as a conditional expectation:

$$y_i^* = \mathbb{E}\left(f(x_i^*)\,|\,f(x_i) = y_i, i = 1, \ldots, n\right).$$

The probability measure is specified by requiring that for any set of points $x_1, \ldots, x_p$ of feature space, the vector of function values $(f(x_1), \cdots, f(x_n))$ should be multivariate normally distributed (remember that here it is f which is random). A normal distribution is uniquely characterized by its mean, which in this context is usually taken to be 0, and its variance-covariance matrix, which is made up of the two-point covariances $k(x_i, x_j) = \mathbb{E}\left(f(x_i)f(x_j)\right)$ (given the zero-mean assumption), so we only need to specify the latter. In fact, *any* kernel function $k(x, x')$ which is positive definite[13] can be chosen as correlation function. Given such a positive definite kernel function, we define a Gaussian process (or Gaussian field) on $\mathbb{R}^n$ by specifying that

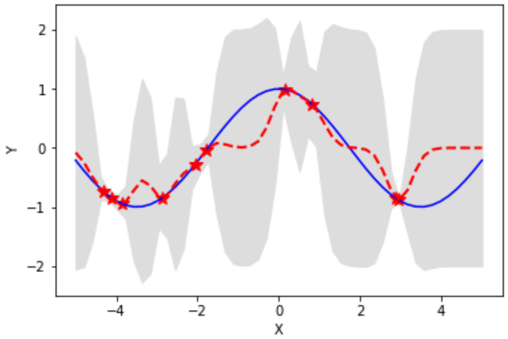
$$(f(x_1), \ldots, f(x_n)) \sim N\left(0, K(X, X)\right), \tag{14}$$

[11] defined as the difference between the upper and lower quartiles

[12] Krige, D., "Gaussian Process Regression", Wikipedia `https://bit.ly/2zHkRkc`

[13] In the sense that $\sum_{i,j=1}^{n} k(x_i, x_j)v_i v_j > 0$ for any choice of points $x_1, \ldots, x_n$ and any non-zero vector $(v_1, \ldots, v_n) \in \mathbb{R}^n$.

Figure 3: Confidence Intervals for GPR Prediction



where $X := (x_1, \dots, x_n)$ and $K(X, X)$ is the $n \times n$ matrix with matrix elements $K(X, X)_{ij} := k(x_i, x_j)$. Given a set of test-points $X^* = (x_1^*, \dots, x_p^*)$, the vector made up of the $f(x_i)$'s and $f(x_j^*)$'s has a similar normal distribution, and standard theory of normal distributions then allows us to explicitly evaluate the conditional expectation as

$$\mathbb{E}(f(X^*) \,|\, f(X) = y) = K(X^*, X)K(X, X)^{-1}y, \tag{15}$$

where$f(X) := \big(f(x_1^*), \dots, f(x_p^*)\big)$, $y = (y_1, \dots, y_n)$ the recorded NIMs in our training set, and $K(X^*, X)$ is the $p \times n$-matrix with elements $k(x_i^*, x_j)$, i being the row index. What is more, we can also compute the conditional variances and covariances of the random vector $f(X^*)$:

$$\mathbb{V}\left(f(X^*) \,|\, f(X) = y\right) = K(X^*, X^*) - K(X^*, X)K(X, X)^{-1}K(X, X^*), \tag{16}$$

which will give a measure of the uncertainty or potential error of our predictions. Observe that the conditional variance-covariance does not depend on y. If we want the process to have a non-zero mean: $f(X) \sim N(\mu(X), K(X, X))$ with $\mu(X) = (\mu(x_1), \dots, \mu(x_n))$, where μ is some given function on feature space, we simply add $\mu(X^*)$ the right hand side of (15) and replace y by $y - \mu(X)$; the formula for the conditional variance remains unchanged. This may be useful if we have some a priori information about the NIM as function of x, e.g. that its values have to lie close to some known function $\mu(x)$.

Figure 3 illustrates prediction under GPR. In this example, the algorithm is trained on a a training set consisting of 10 data points, indicated by a $*$, calculated from a unknown function, represented by the blue line (and which happens to be cosine). Confidence intervals, in grey, are formed by taking the means ± twice the standard deviations. As indicated in the Figure, in regions where a greater number of training points are concentrated the prediction, denoted by the dashed line, is closer to the true function, with a confidence interval which is more narrow.

As regards the choice of kernel function, we have used the *Exponential* and *Squared Exponential* as described in Rasmussen and Williams (2006).

2.10 *NARX Recurrent Neural Network (RNN)*

We start with a brief schematic description of an Artificial Neural Network or ANN. Mathematically, a layer of an ANN sends an input vector $\xi = \mathbb{R}^p$ to a vector of outputs $\eta \in \mathbb{R}^q$ according to

$$\eta_i = \varphi\left(\sum_{j=1}^{p} w_{ij}\xi_j\right), \quad i = 1, \dots, p,$$

where the η_i and the ξ_j are the components of η and ξ, respectively, φ is a non-linear increasing function on $\mathbb{R}$ which is called the *activation function* and the w_{ij} are real numbers which are called the *weights* of the layer. An ANN then simply is a composition of several such layers, with possibly different p's and q's (though the "q" of one layer of course has to be the "p" of the next).

One can depict a layer by the familiar graph, with nodes corresponding to the components of ξ respectively η, and w_{ij} the arrow connecting ξ_j to η_i, where by convention no arrow is depicted if w_{ij} is from the onset chosen to be 0. The linear function $\sum_j w_{ij}\xi_j$ which adds all signals arriving at the node η_i is sometimes called a propagation function. We can and will assume that one of the components of ξ is identically equal to 1, so that the propagation functions all include a constant term. These are sometimes called the bias terms of the neural network. The values of the activation functions are passed through a non-linear filter represented by φ. Observe that we take the same activation function for all components of η and for all layers of the network. Another way to think of an ANN is as a succession of multi-variate linear models interspersed non-linear filters which in a sense cut off the outputs of the linear models when these are too small (this is literally true for some activation functions, such as the step function, but not for others). The choice of the number of layers, the input and output dimensions p and q of each layer and the specification of which weights can be non-zero is known as the *architecture* of the network.

The nodes ξ of the first layer constitute the *Input Layer* and correspond to the components of the feature vector $\mathcal{X}$. The nodes of the other layers are know as the *Hidden Layers*. The outputs of the final layer, possible submitted to a further final transformation known as the *output transfer function*, are the variable y which we seek to model as function of $\mathcal{X}$, and which in our case is scalar. Symbolically, we can write the ANN as

$$y = F\left(\mathcal{X}; \mathcal{W}\right),$$

where $\mathcal{W}$ is a vector representing all of the weights (w_{ij} of all the layers of our network. Training of the network is by minimizing the sum of squared errors, $\sum_i(\widehat{y_i} - y_i)^2$ over $\mathcal{W}$, for a given training set of (x_i, y_i)'s, where $\widehat{y_i} := f(x_i, \mathcal{W})$. This is a in general huge-dimensional optimisation problem, for which efficient algorithms have been developed: see below.

We will be using a time-series (sequential) version of an ANN which is called the Neural Network AutoRegressive model with eXogenous Variables, which was also referred to as *NARX recurrent neural network* in Machine Learning literature (see Lin *et al* (1996) for early introduction about NARX RNN, see Graves (2014) for more recent introduction about RNN). These are neural networks where the input vector of feature variables include lagged values of the output as well as, possibly, lagged values of the exogenous feature variables. The NARX model can be expressed as a non-linear time series regression model including both an autoregressive term of order p for the target variable and a set of lagged exogenous feature vectors $\mathcal{X}_{t-i}$ with d lags:

$$y_t = F(y_{t-1}, y_{t-2}, \ldots, y_{t-p}, \mathcal{X}_t, \mathcal{X}_{t-1}, \ldots, \mathcal{X}_{t-d}, \mathcal{W}) + \epsilon_t, \tag{17}$$

the function F representing a Neural Network with weights $\mathcal{W}$ and ϵ_t representing an error term about which no distributional assumption is made (the training of the model being done by non-linear least squares). In our study, we implemented this model with $p = 1$ and $d = 0$. Moreover,

- The NARX model is implemented using the so-called Closed-loop Feedback architecture, as illustrated by Figure 4, in which the outputs are fed back into the input layers, making it into a dynamical system.

- We chose a simple two-layer network with a single Hidden Layer of 10 nodes; we experimented with other sizes, but found that forecast performances were not significantly enhanced by increasing the Hidden Layer Size.

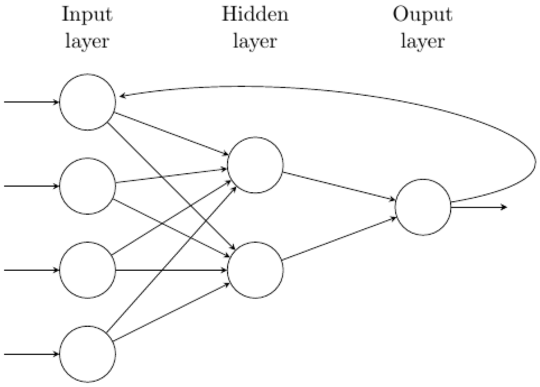
- Unless otherwise stated, we took the *tangent-sigmoid* as our Activation Function and a linear output transfer function.

- Essentially, NARX determines the weight matrices associated to the different layers using non-linear Least Squares, that is, by minimizing the sum of squared errors,

$$\mathcal{E} = \sum_{i=1}^{n} (y_i - \widehat{y_i})^2, \tag{18}$$

  over $\mathcal{W}$, where $(y_i, x_i) : 1 \leq i \leq n\}$ is the training set and $\widehat{y_i} := F(y_{i-1}, x_i, \mathcal{W})$. However, this is modified to guard against over-fitting: see below.

Figure 4: NARX Recurrent Neural Network model



We experimented with two different backpropgation schemes for determining $\widehat{\mathcal{W}}$, the Levenberg Marquardt- or LM scheme and the Bayesian Regularization- or BR scheme. LM is the standard non-convex optimization algorithm used for Neural Networks, but as indicated in Section 3, BR improves upon LM in terms of forecasting performance.

Due to the number of weight parameters used in Neural Network, it can potentially overfit the training data. To mitigate the risk of overfitting, one includes a penalty term for the weights in a Bayesian framework following MacKay (1992).

As indicated in Table 1, we looked at the two different algorithms for backpropogation to investigate their impacts on the NARX based NIM forecast model performance, for all 10 standard feature selections and 6 additional add-on configurations described in Appendix A.

2.11 *Forecast Evaluation and Accuracy Metrics*

Our objective is to estimate the forecast accuracy for the h-step (where $h = 1, \cdots, H$) ahead NIM, which is a random variable, forecasted by a forecasting model. As discussed above, instead of applying a fixed forecast-origin forecasting, we adopt a rolling forecast-origin for a rolling training set indexed by r; then we calculate the associated forecasting errors as the gap between h-step ahead forecasted NIM denoted by $\widehat{y}^r_{i+h}$ and the observed one denoted by y^r_{i+h}. That is,

$$\epsilon^r_h := \widehat{y}^r_{i+h} - y^r_{i+h} \tag{19}$$

It is well-known that scale-dependent forecast accuracy metrics can be used to compare forecasting performances for models learned from datasets of comparable scales, whereas when using datasets living on different scales, scale-independent metrics should be preferred. In the context of NIM modelling, the variations in the NIMs amongst banks may be driven by the differences in their business strategies, competition status in terms of market shares, etc. For example, a bank with its business focus on emerging markets may face less intense competitions and more volatile sales than does another bank which operates in more mature, stable and competitive markets. As a result, the latter may have a more stable NIM than the former, which leads to more subdued *RMSE*. Therefore, comparing the RMSEs in this case may be misleading, and it may be more appropriate to use a scale-independent error metric when comparing the performances of a NIM forecasting model for the two banks.

Some examples of scale-dependent forecast accuracy metrics for the h-step Forward NIM are:

1. The widely used *Root-Mean-Squared Error*, which is estimated as the sample forecasting errors obtained from R training sets in our study as:

$$RMSE_h = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\left(\epsilon^{(r)}_h\right)^2} \tag{20}$$

2. The *Mean-Absolute Error* or MAE, defined by

$$MAE_h = \frac{1}{R}\sum_{r=1}^{R}\left|\epsilon_h^{(r)}\right| \tag{21}$$

3. The *Maximum Absolute Error* or ME, given by

$$ME_h = max_r \left|\epsilon_h^{(r)}\right| \tag{22}$$

Obviously, $MAE_h \leq RMSE_h \leq ME_h$. The MAE may be less sensitive to outliers than the RSME. As discussed in Section 1.3, Armstrong et al (1992) found the RMSE to be sensitive to outliers and neither reliable nor appropriate for comparing the performances of time series forecasting models on data-sets of different scales. They proposed to use a measure called the Relative RMSE or RelRMSE. More recently, Chen, Twycross and Garibaldi (2017) conducted extensive studies regarding statistically sound and easily interpretable scale-independent measures. These authors found the so-called *Unscaled MBRAE* or UMBRAE to be the best measure for the purpose of comparing the performances of Time Series forecasting models. We therefore have used this measure, plus with a variant of the RelRMSE. They measure performances relative to those of a benchmark model, which in our case is the Random Walk model, the forecasting errors of which we will denote by $\epsilon_h^{(r)*}$. Note that these depend on the training set D_r^T since, under the Random Walk model, $\widehat{y}_{t+h}^{(r)} = y_{r+\ell-1}$ with $t = r + \ell - 1$.

1. One first defines the *Mean Bounded Relative Absolute Error* or MBRAE for the *h*-step forward forecast of the NIM as

$$MBRAE_h = \frac{1}{R}\sum_{r=1}^{n}\frac{\left|\epsilon_h^{(r)}\right|}{\left|\epsilon_h^{(r)}\right| + \left|\epsilon_h^{(r)*}\right|}. \tag{23}$$

   Observe that $0 \leq MBRAE_h \leq 1$ and that if $MBRAE_h < \frac{1}{2}$, then the forecast is on average better than the benchmark forecast.

2. The *Unscaled MBRAE* or UMBRAE for the *h*-step forecast then is defined by

$$UMBRAE_h = \frac{MBRAE_h}{1 - MBRAE_h}. \tag{24}$$

   We note that the UMBRAE is always positive and that $UMBRAE_h < 1$ is equivalent to $MBRAE_h < \frac{1}{2}$.

3. Finally, assuming we attach the same importance to all forecast horizons, the *Average UMBRAE across Horizons* $h := 1, \ldots, H$ is

$$UMBRAE = \frac{1}{H}\sum_{h=1}^{H} UMBRAE_h \tag{25}$$

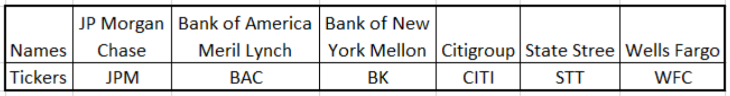
   Since, following Chen, Twycross and Garibaldi (2017), $1 - UMBRAE > 0$ can be interpreted as the model's performance being, on average, better than that of the benchmark Random-Walk model while $1 - UMBRAE < 0$ means its poorer than the benchmark one, it is convenient to use $1 - UMBRAE$ as forecast accuracy metric, rather than the UMBRAE itself, as we do in section 3 below on our empirical results.

   Machine Learning literature often uses ranking of performance measure in empirical comparison studies; for example, Ahmed, Gayar and El-Shishiny (2010) uses the average rank of a model across different time series as a forecast accuracy metric. In our paper, we use *Average Ranking of a Model M* based on the UMBRAE-metric, defined by

$$ARABU_M = \frac{1}{B}\sum_{b=1}^{B} Rank(b, 1 - UMBRAE_M) \tag{26}$$

   where B is the number of banks in sample (in our case, 6), the index b runs over the different banks and M indicates the model, with $UMBRAE_M$ the set of UMBREAs found for these banks when using model M.

Table 3: A list of Banks' Names and Tickers



Names	JP Morgan Chase	Bank of America Meril Lynch	Bank of New York Mellon	Citigroup	State Stree	Wells Fargo
Tickers	JPM	BAC	BK	CITI	STT	WFC

4. Finally, as an alternative to the UMBREA, we use a variant of the Relative RMSE of Armstrong et al. We define the *Average Forecasting Performance Improvement Ratio* or AFPIR, relative to the benchmark, as

$$AFPIR = \frac{1}{H}\sum_{h=1}^{H}\left(1 - \frac{RMSE_h}{RMSE^*_h}\right), \tag{27}$$

where $RMSE_h$ stands for the RMSE calculated for forecast horizon h, $RMSE^*_h$ stands for the RMSE for benchmark model, and where we attach the same importance to all forecasting horizons $h \in \{1, \ldots, H\}$. We recall that the Relative RMSE for the horizon-h forecast is simply $RMSE_h/RSME^*_h$. Considering 1 minus this measure has the advantage that the boundary between better or worse performance than that of the benchmark lies at 0.

In the next section we will compare model performances across banks using these *UMBRAE*-, *AFPIR*- and $ARABU_M$-metrics.

3 Summary of Empirical Results for NIM Models

In this section, we present and discuss our main empirical results from our investigation on Bank-specific NIM forecast models using the 11 Machine Learning-regression algorithms, which include 8 Machine Learning-regression algorithms reported on basis of the same set of feature variables for the purpose of Cross-model comparison (see Figure **??** and 5) and 3 additional regression algorithms (i.e., two PCR-related algorithms and a Stepwise regression algorithm) for the purpose of Robustness study. The empirical results contain investigations of above models across a range of feature selections and parameterization choices and for the following six of all eight G-SIBs US banks[14], for which Table 3 presents their associated stock tickers to be referenced in the discussions below.

This section presents the empirical results pertaining to 11 Machine Learning-regression algorithms in three parts:

- In section 3.1 we present Cross-model forecast performance results for the top-3 performing models to represent each of the 8 different Machine Learning-regression algorithms (while presenting the performance results for the remaining three algorithms as part of Robustness study) trained with our standard feature selections (FS1-FS10), using two scale-independent forecast accuracy metrics, i.e., (1 - UMBRAE) and APFIR (*cf.* section 2.11). With this, we can compare model performance across the different Machine Learning-regression algorithms.
- Next, in section 3.3 we turn to *Intra-model forecasting performance comparison*, comparing the forecasting performances of each of the 8 Machine Learning-regression algorithms separately (while leaving the results of the three algorithms for Robustness study), as function of feature variable selection and of parametrisation choices. Here we will only use the APFIR for comparing performances (relative to the RW model), for reasons explained below.
- Section 3.4 presents the empirical performances from linear regression models with feature variables obtained from Principal Component Analysis in ordinary linear regression or its Robust version as well as those from linear regression models with feature variables obtained from Stepwise regression. Since these Machine Learning-regression algorithms have their bespoke feature choices as opposed to

[14] Following other NIM literature, we excluded Goldman Sachs and Morgan Stanley from our sample because they joined as bank-holding companies during financial crisis, so have had relative short history as a bank.

the feature selections mentioned above, we exclude them from cross-model performance comparison. Instead, they appear at the end of the section under the heading of Robustness Study.

3.1 *Cross-model forecast performance for NIM*

We follow so-called *Maximum Accuracy* practice used by empirical performance comparison in literature such as Delgado and Amorim (2014) to compare different algorithms based on top-3 NIM forecast models (which are the three models with highest forecast accuracies based on RMSE for the same bank out of each Machine Learning-regression algorithm.) to represent each of the 8 Machine Learning-regression algorithms described in Section 2, based on two scale-independent metrics recommended by forecasting literature (*cf.* section 2.11); our empirical results show that the forecast performances based on the top-3 models representing each of the 8 regression model types are consistent for both (1-UMBRAE) and AFPIR measure with the latter showing less volatility for performance. For simplicity of exposition, we only present the results for AFPIR in 5 and with numerical results shown in Table 4. We note that Feature Selections can be found in Appendix A and that summary statistics about Bank NIM data can be found in Appendix B.

3.2 Cross-model type performance comparisons based on AFPIR

As an alternative measure, we present a summary of the overall performances of Bank-specific NIM forecast models based on top-3 performing models measured by an another scale-independent forecast accuracy metric, i.e., AFPIR, as described in Section 2.11. Figure 5 exhibits the top three performing NIM forecast models for all the six G-SIB banks based on their AFPIR average across eight forecast horizons. In both Table **??** and 4, we highlight *performance losses* over the benchmark by a model (thus, negative values) respectively measured by (1-UMBRAE) and AFPIR by pink colour; meanwhile, these cells containing positive values are not highlighted to indicate the relative performance *gain*. X-axis presents the labels for each of the eight Machine Learning-regression algorithms, corresponding with Table 1 as described in Section 2.

We note that

1. Cross-model performance patterns are largely similar; for example, clustered performances for each bank are visible and slightly more so in Figure 5 than those measured by (1-UMBRAE) as indicated by Figure **??**; performances for "jpm" continue to be stable and high, etc. However, AFPIR gives a more consistent performances, which is shown by the fact that greater area is highlighted in pink in Table **??** than that is Table 4 (Pink indicates that models under-perform the benchmark); also, the performances shown in Figure 5 are less volatile than those in Figure **??**.

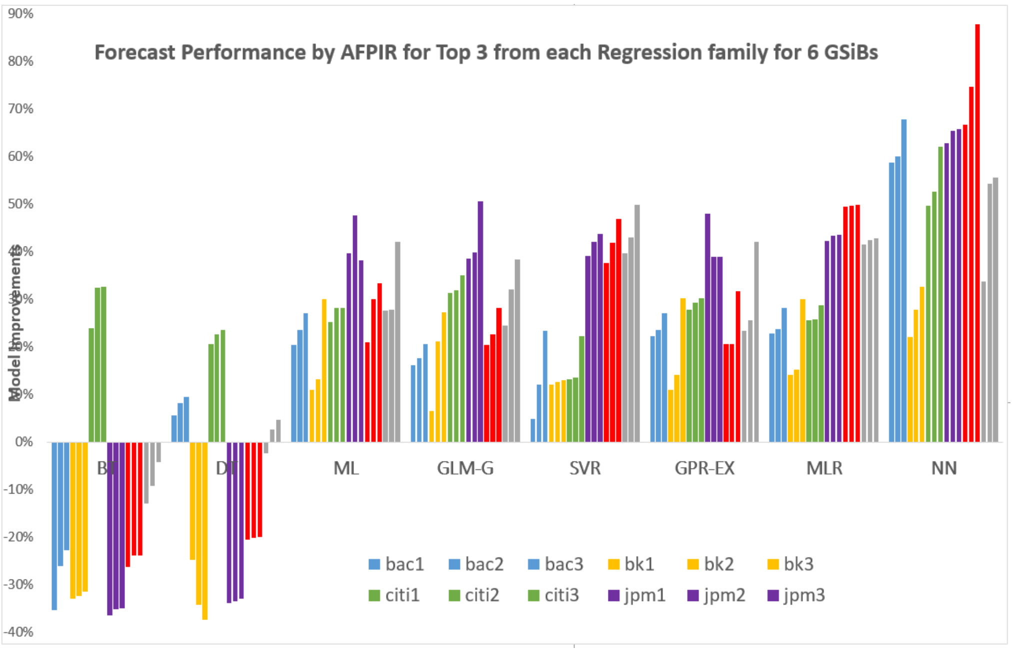
2. Figure 5 shows that both Linear Models and Non-linear Models can outperform the random-walk benchmark as indicated by positive signs of AFPIR, with exceptions of those Tree-based models, which are highlighted in pink colours in Table 4.

3. Again, Figure 5 clearly shows that NARX-RNN model achieved the highest forecast performance gains relative to the benchmark, followed by Robust Regression version of ML, then followed by the next group (SVR, GPR-EX, GLM-G, ML) all with reasonable performance gains, leaving two tree-based forecast models, which largely under-perform the benchmark.

4. Table 4 shows that, again, NARX-RNN based NIM forecast models achieve the highest improvement relative to the benchmark; e.g., Wells Fargo Banks (as indicated by "WFC1-WFC3") achieve the highest improvement over the benchmark by 87.8% for "WFC3" in the last column in Table 4.

5. Table 4 includes the Median calculated across the 18 models (see explanations above) associated under each Machine Learning-regression algorithm in the last row; for example, MLR NIM forecast models, on average (based on Median), achieve second best model improvement relative to the benchmark model; MLR models for Wells Fargo Bank, which displays Non-Gaussian distribution and outliers in NIM data, ranked the second highest maybe due to benefits from the performance boost of Robust Regression. As highlighted, the Median for SVR is the third highest, slightly different from the Table **??**, similarly, we highlight ML.

Figure 5: Forecast Performance Gain over Benchmark by Top 3 Models Measured by AFPIR



For reasons discussed in #1 above, we prefer AFPIR as forecast accuracy metric due to its consistent, less noisy outputs relative to (1-UMBRAE). In the rest of the Section, we will use *AFPIR as opposed to 1-UMBRAE* to rank the model performances in Section 3.3.

3.3 *Model-specific Performances for NIM*

In this section, we focus on reporting the performance variations within each of eight Machine Learning-regression algorithms based on our favoured scale-independent AFPIR forecast accuracy metric.

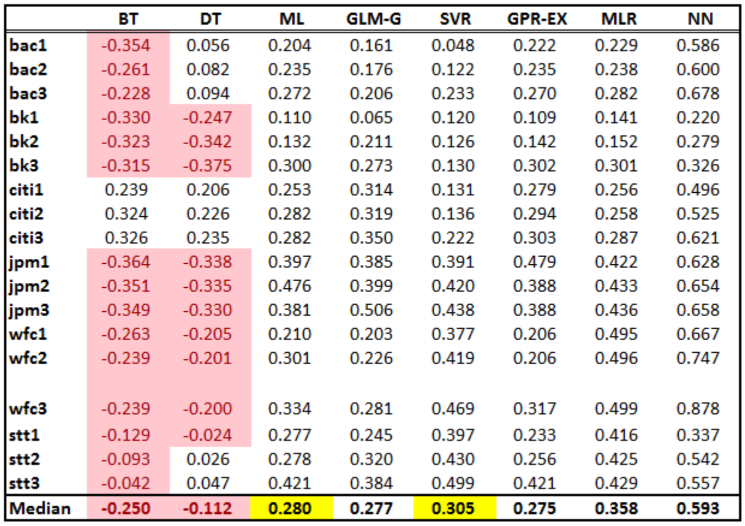
In this part, we first present the Intra-model forecast performances measured by AFPIR related to ML and MLR model types for six G-SIB banks for the 10 Standard Feature Selections (cf. Appendix A). Figure 7 shows the performances for ML and MLR-based Bank-specific NIM forecasting models with the corresponding numerical results given in Table 7 across 10 Standard Feature Selections. For ease of exposition, for the rest of other forecasting models, unless otherwise stated, we only present the six models with higher accuracies than the rest from the standard feature selections.

Regarding ML and MLR, we note that

1. Overall, out of ML and MLR associated models, the vast majority of the cases have higher forecast accuracies than the benchmark, with only 7 out of 120 cases (each bank is treated as one case) under-performing the benchmark, which are highlighted in pink colours in Table 7: 4 cases are related to either FS9 or FS10 (which include bank-specific feature variables such as ΔPD_t and ΔOC_t.) for ”bac”, 1 case to ”citi” banks, which suggests, contrary to expectation, *including bank-specific variables does not always help improve forecast performances in bank-specific NIM model*. In fact, in all models, relative to all macroeconomic variable models, neither FS9 nor FS10 based ML/MLR related models achieve the highest forecast accuracy, indicating that the bank-specific variables may not be as relevant as we expect.

2. For ”wfc”, the remaining two under-performing cases (as highlighted in pink in Table 7) are both related to ML; as expected, their forecast performances are significantly boosted by using MLR instead: for FS1, the AFPIR for ”wfc” jumps to 48% from -4% and for FS7, the AFPIR for ”wfc” increases to 46% from -3%, both due to the use of Robust Regression in regression models.

Table 4: Forecast Performance Gain over Benchmark by Top 3 Models Measured by AFPIR numerically



	BT	DT	ML	GLM-G	SVR	GPR-EX	MLR	NN
bac1	-0.354	0.056	0.204	0.161	0.048	0.222	0.229	0.586
bac2	-0.261	0.082	0.235	0.176	0.122	0.235	0.238	0.600
bac3	-0.228	0.094	0.272	0.206	0.233	0.270	0.282	0.678
bk1	-0.330	-0.247	0.110	0.065	0.120	0.109	0.141	0.220
bk2	-0.323	-0.342	0.132	0.211	0.126	0.142	0.152	0.279
bk3	-0.315	-0.375	0.300	0.273	0.130	0.302	0.301	0.326
citi1	0.239	0.206	0.253	0.314	0.131	0.279	0.256	0.496
citi2	0.324	0.226	0.282	0.319	0.136	0.294	0.258	0.525
citi3	0.326	0.235	0.282	0.350	0.222	0.303	0.287	0.621
jpm1	-0.364	-0.338	0.397	0.385	0.391	0.479	0.422	0.628
jpm2	-0.351	-0.335	0.476	0.399	0.420	0.388	0.433	0.654
jpm3	-0.349	-0.330	0.381	0.506	0.438	0.388	0.436	0.658
wfc1	-0.263	-0.205	0.210	0.203	0.377	0.206	0.495	0.667
wfc2	-0.239	-0.201	0.301	0.226	0.419	0.206	0.496	0.747
wfc3	-0.239	-0.200	0.334	0.281	0.469	0.317	0.499	0.878
stt1	-0.129	-0.024	0.277	0.245	0.397	0.233	0.416	0.337
stt2	-0.093	0.026	0.278	0.320	0.430	0.256	0.425	0.542
stt3	-0.042	0.047	0.421	0.384	0.499	0.421	0.429	0.557
Median	**-0.250**	**-0.112**	**0.280**	**0.277**	**0.305**	**0.275**	**0.358**	**0.593**

3. Using Robust Regression does not always lead to significant performance boost if the variable has been logarithmically transformed, which already has built-in resistance against outliers. For example, comparing the performances between Robust-version models and the Non-Robust ones under FS6 (highlighted in Green colours in Figure 7), the performances are relatively similar for all banks but ”wfc”, which has the most outliers in NIM data. FS6 (cf. Appendix A.1) includes the logarithmic transformation of equity prices, which reduces the variations for extreme values, but the built-in outlier resistance through logarithmic transformation is not sufficient for the extreme outliers of ”wfc” so that Robust Regression is still helpful.

We now move to present the summary for the NIM forecast performance based on scale-independent AFPIR metric for GLM. As discussed in Section 2, GLM represents a large family of regression models; despite its name (linear), it is quite flexible and generalized. In our study, we focus only on Gamma Regressions.

Regarding GLM-related models, we note that

- Figure 8 presents the forecast performances based on Gamma (labelled by ”GLM-g”) Bank-specific NIM forecasting models measured by APFIR for six banks across 6 Feature Selections (FS1-FS6 as described in Appendix A); Table 8 presents the numerical results related to the Figure.

- Figure 8 indicates that the magnitudes for performance variations due to feature selections are significantly different across banks. For instance, ”citi” and ”jpm” are largely small; whereas, the differences are significant for ”bac”, ”bk” and ”wfc”.

Now we move to present the NIM forecast performances based on Regression Tree and Bagged Tree Regression models. Figure 9 presents the summary of the empirical results for Regression Tree model’s performances based on scale-independent AFPIR across all banks; the corresponding numerical results displayed on the Figure can be found in Table 9.

Regarding Regression Trees, we note that

1. As a weak learner, Regression Tree based Bank-specific NIM forecast models under-perform Random Walk for 5 out of 6 banks as shown by the mostly negative numbers in Figure 9. For example, for ”FS7” of ”JPM” (JP Morgan bank), Regression Tree under-performs almost by 80%.

2. For the same banks as indicated by the two groups of paired coloured bands, we can compare the performances between a bank’s NIM model with a subset of feature variables determined based on

so-called ”curvature test” vs the performance for the same bank with ”Others” configuration. The definition for ”Others” is presented in the last column in Table 9. For example, for the orange band of ”bac”, indicating Regression Tree built with a subset of features selected based on ”curvature test” (See 2.7.1), when compared with the same bank's tree based on ”All” (meaning that All feature variables are included in tree growth.), clearly, the latter improves by 6% over Random Walk vs the -43% underperformance relative to Random Walk for the same feature selection, i.e., ”FS8”.

3. However, including all feature variables as splitting variables do not always improve the forecasting performances, which is clear by inspecting the first two bands (from the left) for each pair belonging to the same bank.

4. Our empirical results suggest that Regression Tree is too crude for Bank-specific NIM forecasting in our study.

Regarding Bagged Tree, we note that

1. Figure 10 presents the summary of the empirical results for Regression Tree model's performances based on scale-independent AFPIR across all six G-SIB banks; the underlying numerical results displayed on the Figure can be found in Table 10.

2. We attempt various Ensemble techniques such as Random Forest (or simply denoted by ”RF” in the last column under the heading ”Others” in Table 10), Boosted Tree (”LSboost” as shown under ”Others” in Table 10) and increasing the number of Learning Cycles (labelled with ”BT-300”), we do not see obvious improvement for the relative performances measured by AFPIR metric. Interested readers can find references for ”Others” techniques in the general reference of our paper: Hastie, Tibshirani and Friedman.

3. We believe it still is caused by method of Regression Tree Regression itself being too crude.

We now present the empirical results related to two Gaussian Process Regression (GPR) models, based on two choices of Kernel functions: ”Exponential” kernel (which is labelled by ”E” in Table 11 and Figure 11 and also called ”Ornstein Uhlenbeck” kernel.) and ”Squared-Exponential” kernel (which is labelled by ”SE” in Table 11 and Figure 11.) as described in Section 2.9.

Regarding Gaussian Process Regression, we note that

1. Figure 11 presents the APFIR produced by GPR under the two kernel functions referred to by ”EX” and ”SE” respectively.

2. The results in Figure 11 are grouped into two displayed together for each of the six banks with coloured bands each corresponding with one of the Feature Selections (FS5-FS10) for both kernel functions although for ”EX”, we study for all FS1-FS10.

3. From Figure 11, it is not straightforward to see the differences between the performance variations due to the kernel assumptions except that ”WFC” has clearly better performance for ”EX” related to ”FS7” than its counterpart for ”SE”.

4. By observing the numbers in Table 11, we can see that in 27 out of the 36 models displayed in the table, ”EX” kernel functions have led to better performances than ”SE”.

We present a summary of the empirical results related to Support Vector Regression (SVR) with different choices of various linear and non-linear Kernel functions. Section 2.8 presents a summary of SVR and the description about kernel functions used in this study. For non-linear Kernels, we use ”Gaussian”, ”Polynomial of order of 3” and ”RBF” respectively.

Regarding Support Vector Regression, we note that

1. In Figure 12, the APFIR metrics for each of the six banks are displayed adjacent to each other in form of two groups of coloured bands, representing the APFIR metrics for Feature Selections studied for SVR (i.e., FS5-FS10, which readers can refer to Section A for more details).

2. Figure 12 shows that, based on the significantly negative APFIR, FS9, the Feature Selection containing bank-specific variables such as *PD* (Probability of Default) and *OC* (Operational Costs to Asset Ratio), clearly is not favoured by SVR for any of the six banks except for ″stt″ for both linear and non-linear Kernel functions.

3. Comparing each pair related to each of the six banks, based on APFIR metric, SVR using ″Linear″ kernels lead to higher APFIR than ″Non-linear″ kernels. For those with positive APFIR numbers, i.e., ″citi″ (except for FS10), ″jpm″, ″wfc″ and ″stt″, the associated models outperform the benchmark whereas, for the remaining banks', their APFIR are negative or the associated models underperform the benchmark. In either case, ″Linear″ kernel based models perform better than ″Nonlinear″ kernel based models.

4. Table 12 presents the numerical results for Figure 12; in addition, the last row contains the Feature Selections with the best results in terms of APFIR for the associated banks. It is observed that both FS1 and FS6 are the Feature Selections perform better than other Feature Selections across all the 10 Feature Selections in the paper under SVR. Furthermore, it is worthwhile to point out that both FS1 and FS6 are transformation of equity prices based on first difference and logarithm of equity prices respectively, which are chosen to represent the Credit Risk for the economic environment.

For NARX RNN, we study 16 different regression models: for 10 standard Feature Selections, we applied Bayesian Regularization (BR) within Levenberg-Marquardt (LM) backpropgation algorithm, which are indicated by ″NN-BR″; additionally, we tested 6 models, based on FS1-FS6, where we only applied Levenberg-Marquardt without BR, which are indicated by ″NN-LM″ as shown in Figure 13.Correspondingly, Figure 13 only shows FS1-FS6. Regarding their performances measured by AFPIR, we present a summary of empirical results related to NIM forecast models based on NARX RNN across Feature Selections and regularized (by Bayesian) and un-regularized backpropgations.

Regarding NARX RNN model, we note that

1. In Figure 13, each label together with a group of coloured bands on x-axis correspond with a bank and the Feature Selections (FS1-FS6) with the label containing the bank plus the backpropgation algorithm applied to the group. For example, ″bac-NN-BR″ reads like ″Bank of America with NARX-RNN with Bayesian Regularization″ applied to FS1-FS6 with colours distinguishing the Feature Selections.

2. For the same banks, two groups of coloured bands are presented adjacent to each other to contrast the effects of applying Bayesian Regularization or not, i.e., ″NN-BR″ vs ″NN-LM″.

3. Clearly, in all 72 models except for one presented in Figure 13, ″NN-BR″ models perform better than ″NN-LM″ based on our own APFIR metric. For instance, the first group of coloured bands stand for Bank of America's 6 Feature Selections, for which we investigated.

4. Table 13 shows that with Bayesian Regularization applied, across all six Feature Selections, all six banks except for Bank of New York at Mellon (indicated by ″bk″) significantly outperform random-walk benchmark models based on APFIR.

5. In contrast, for 29 out of 36 Feature Selections, the Levenberg-Marquardt Algorithm without applying Bayesian regularization lead to underperformances relative to Random Walk.

3.4 *Robustness Study: PCR and Stepwise regression*

The objective of the section is to present a robustness check we conducted regarding variable selections based on Principal Component Regression (or PCR) and Stepwise Regression. PCR-ML (PCR in the context of ML regression) is used in current NIM study; thus, as a Robustness study, we present Bank-specific NIM forecast performance according to so-called Factor-based regression models. Further, we present results from PCR-MLR (PCR in the context of MLR regression model) with regard to varying the retained number of PCs (denoted by m) for six G-SIB banks. In Figure 14 and Table 14; model types are labelled with ″PCRML″ and ″PCRMLR″ respectively and the retained PC number m are distinguished by different legends; e.g., ″PC-3″ denotes $m = 3$.

Regarding PCR-related models, we note that

1. In Figure 14, the APFIR bands associated with each bank are displayed adjacent to each other to indicate the changes of Forecast Performances Improvement vs the benchmark with regard to m from 3 to 11 ($max(m) = 11$; cf. Appendix A.3).

2. With $m = 3$, indicated by the Blue bands related to all six banks, no model except for "wfc-PCRMLR" across all banks can achieve better performances than the benchmark, which is also indicated by the second row the $AFPIR = 0.19$ as highlighted Green in Table 14.

3. Table 14 indicates that increasing m does not increasing forecasting performances measured by scale-independent AFPIR; instead, in some cases, forecasting performances are observed to peak at a certain m. For example, "bk-PCRMLR", "bk-PCRML", "jpm-PCRMLR", "jpm-PCRML", "wfc-PCRML" all peak their AFPIR at $m = 4$.

4. We observe significant performance differences between PCR-ML and PCR-MLR as we did so between ML and MLR, which, again, suggests that Robust Regression is conducive to NIM forecast performance in our study.

In our study, as a Robustness study, we use Stepwise Regression in combination with ML models to study the NIM forecast performances measured by ARABU (*cf.* Section 2.11) to rank order Feature Selections based on a pre-specified criterion such as BIC, AIC, P-value for F-test (or simply P+F in Table 15), Adjusted R-squared (or simply $adjR^2$ in Table 15), etc. The sequence of models that are created from a list of candidate variables up to various transformation of these variables; e.g., we consider quadratic terms of some feature variables, etc, with the full list of Feature Selections available in Appendix **??**.

Regarding Stepwise Regression, we note that

1. Table 15 shows the rank ordering of performances under 16 combinations labelled as "FS + 1 Model Selection Criterion", with metric called ARABU calculated from the ranks determined through Stepwise Regression; the smaller the ranking, the higher the performance is. Altogether, Stepwise Regression is performed across 16 modelling Set-ups.

2. Table 15 consists of two parts highlighted in Blue and Green respectively: region highlighted in Blue shows the 10 Models (Model 1 to Model 10), each containing one of 10 standard Feature Selections (i.e., FS1 to FS10) + a model selection criterion being BIC; the region in Green has 6 Models (Model 11 to Model 16) containing feature selection labelled by "FS5-FS10" together with their respective selection criterion. The set-up gives us a way to see the performance variations between different feature selections and different selection criteria during Stepwise Selection. For example, one can compare the list of Model 5 to 10 to the list of Model 11 to 16, two are different only by their selection criteria or feature transformations.

3. It is well-known that BIC selection criterion favours more compact and less complex regression models. As shown in Table 15, BIC-based models tend to outperform their counterparts under other selection criteria; e.g., Model 1 is ranked better than Model 4 (cf. Appendix **??** for FS1 and FS4) due to the compactness of Model 1 vs 4. Similarly, given the same feature selections, Model 5 vs 11 comparison shows that BIC based model tends to be ranked better than AIC based model, which is based on definitions of BIC and AIC because BIC penalizes big models more than does AIC. This observation is reflected from other rankings as well: Model-8 vs Model-14, Model-9 vs Model-15, Model-10 vs Model 16, etc.

4. However, ARABU is not universally a good metric for performance comparison. For example, as indicated in Table 15 above, it produces "ties" between Model 1, 7 and 8. Furthermore, although based on APFIR metric shown in Figure 15, "FS8" has a smaller aggregate area size for the coloured bands relative to that for "FS6", ARABU metric still ranks the model improvement across banks for "FS8" above "FS6".

5. Figure 15 presents the APFIR across six banks and all 16 Feature Selections in our study as given in Appendix A; Table 16 presents the numerical results corresponding with Figure 15.

6. Figure 15 shows significant variations across different features and banks despite the same models have been consistently applied to the same set of banks. Notably, ”citi” despite its NIM volatility (0.616) being smaller than ”wfc” (1.466), ”citi’ displays ”0.93” volatility for its APFIR across 16 Feature Selection as shown in the last row of Table 16.

7. FS14 has a 1.14 volatility across banks for its APFIR, indicating that it is an undesired feature selection for constructing NIM forecast model. Similarly, FS16 has the same problems, indicating adding bank-specific variables in Regression models might lead to underperformances by forecasting models.

4 Conclusions & Research Directions

The investigation is a first systematic study regarding the forecast accuracy for Bank-specific NIM forecasting models for individual banks’ and has led to the following conclusions:

1. As a multi-step forecasting study, we have applied forecasting techniques recommended by forecasting literature: first, we adopted a widely used Random-walk benchmark in related literature; also, we used iterated multi-step forecasting as opposed to direct multi-step forecasting, which allows model recalibration to include the data variations and signals between sampling periods; third, we used rolling forecast-origins instead of fixed ones so that we can estimate our horizon-specific forecast accuracies based on a distribution of forecasting errors rather than one data point.

2. Based on the same random-walk benchmark model, in contrast with the findings in literature on Aggregate NIM, we find that both the traditional statistical regression models and the non-linear Machine Learning regression models can forecast Bank-specific NIM with significantly higher accuracies than the random-walk benchmark model based on out-of-sample tests measured by two scale-independent forecast accuracy metrics recommended by literature (*cf.* Table **??** and 4 respectively). Thus, the grounds used by literature to challenge the validity of stress-test may not be sound; instead, forecast models built with more granularity, as we do in this study, may be conducive to achieving better forecast accuracy for NIM models.

3. In terms of regression forecasting techniques, some Machine Learning techniques complement the traditional statistical regression ones by achieving significantly better forecasting performances than the latter; see NARX-RNN as an example.

4. Robust version of traditional regression models can significantly improve forecasting accuracies in presence of outliers and non-Gaussianity in data samples; see the Cross-model performance variations in cases of ML vs MLR, PCR-ML vs PCR-MLR and the overall superior performance exhibited by MLR for banks with outliers in NIM.

5. We demonstrate the importance of using scale-independent forecast accuracy metrics in comparing forecast performances from models based on data of varying characteristics, in our case, NIM data with outliers, as opposed to popular scale-dependent one such as RMSE. In our study, two literature recommended, i.e., (1-UMBRAE) and AFPIR lead to largely similar conclusion but we justify the latter based on its forecasting results that are more consistent, more stable and more interpretative than the former.

6. As a time series forecasting study based on a number of regression models, then performance comparison problem naturally emerges; it turns out our rank-ordering results for forecast performances of non-linear models are: Neural Network (or more specifically NARX RNN), GPR and SVM, which is largely consistent with empirical performance comparison literature represented by Ahmed, Gayar and El-Shishiny.

7. Meanwhile, we find that, applied in the context of Regression for NIM forecasting, Decision Tree and Ensemble/Decision Tree produce unsatisfactory forecasting accuracies.

8. Our Cross-model performance comparisons show that NARX RNN Regression models outperform the benchmark model by up to 87.6%. Intra-model performance comparisons for NARX RNN show that Bayesian Neural Network outperforms other choices of backpropagations, which is consistent with existing literature such as Zhang, Patuwo and Hu (2003).

9. We find that Feature Selections play important roles in forecast performances. We find it important to base feature selections on economic principles and literature findings (*cf.* section 1.2). As well, our robustness study shows that, alternative feature selections can bring up insights; for instance, stepwise regression techniques can automate feature selection to a certain extent and lead to reasonable accuracies while the findings from PCR largely contradicts those from the literature on Aggregate NIM forecasting (*cf.* section 3.4).

10. Contrasting with existing NIM literature, our data samples cover time periods ranging from 2000Q1 to 2016Q4 and each of our training sets contains the Lehman's debacle for benefits explained in section 2.2.

11. As policy suggestions, we recommend that regulators should disclose forecast accuracy information based on out-of-sample tests for the forecast models used by banks and regulators, which should be conducive to maintain confidence from investors and creditor and to the financial stability. Furthermore, banks should be encouraged to conduct research and development for forecast models at more granular levels such as asset classes, which will help regulators and banks identify the vulnerability in business areas within banks at build-up stages before it transforms into a crisis.

As regards directions for future research, the research can be extended in many interesting directions. First, research is needed to examine whether the forecast accuracy problem is applicable to bank-specific asset-loss forecast models. Second, research is lacking in terms of modelling bank's balance sheet dynamics. Third, research is needed to study forecast models at more granular levels, e.g., asset class level. Last, the study is based on six G-SIB US banks; research based on different geographies and different types of banks can be conducted to see if the conclusions remain valid.

References

[1] Acharya V., R. Engle and D. Pierret. 2014. ”Testing macroprudential stress tests: The risk of regulatory risk weights”. *Journal of Monetary Economics*, Vol. 65 (July) p36-53

[2] Ahmed, N., A. Atiya, N. Gayar and H. El-Shishiny. (2010). ”An Empirical Comparison of Machine Learning Models for Time Series Forecasting”. *Econometric Reviews*. (September) 29:5,594-621.

[3] Armstrong, J. S., and F. Collopy. (1992). ”Error Measures for Generalizing About Forecasting Methods: Empirical Comparisons”. *International Journal of Forecasting*, Volume 8, Issue 1 (June), Pages 69-80

[4] Basel Committee on Banking Supervision (BCBS). (June 2011). ”Basel III: A global regulatory framework for more resilient banks and banking systems”, BIS.

[5] Bergmeir C., R. J. Hyndman and B. Koo (2018). ”A note on the validity of cross-validation for evaluating autoregressive time series prediction”, *Computational Statistics and Data Analysis* 120, p70–83

[6] Berndt, A., R. Douglas, D. Duffie, M. Ferguson and D. Schranz. (2005). ”Measure Default Risk Premia from Default Swap Rates and EDFs”, *BIS Working Papers* 173.

[7] Bolotnyy, V., R. Edge and L. Guerrieri. (May 2015). ”Revenue Forecasts, Capital Adequacy and the Uncertainty of Stress Test Results”, *Harvard Univ. and Federal Reserve Board working paper*.

[8] Breiman L., J. Friedman, R. Olshen and C. Stone. (1984). *Classification and Regression Trees*, New York:Wadsworth.

[9] Brummelhuis, R. and Z. Luo. (2019a). ”CDS Proxy Construction via Machine Learning Techniques: Methodology and Results”. *Journal of Financial Data Science Spring Vol. 2019*.

[10] Brummelhuis, R. and Z. Luo. (2019b). ”CDS Proxy Construction via Machine Learning Techniques: Parametrization, Correlation and Benchmarking”. *Journal of Financial Data Science Spring Vol. 2019*.

[11] Brummelhuis, R. and Z. Luo. (2018c). ”Arbitrage Opportunities in CDS Term Structure: Theory and Implications for OTC Derivatives”. *SSRN Electronic Journal*.

[12] Brummelhuis, R., A. Cordoba, M. Quintanilla and L. Seco. (2002). ”Principal Component Value at Risk”. *Mathematical Finance*, Vol. 12, pp. 23-43.

[13] Busch, R. and C. Memmel. (2014). ”Quantify the Components of banks’ Net Interest Margin”, *Financial Markets and Portfolio Management* (November) Volume 30, Issue 4, pp 371–396.

[14] Cortes C. and V. Vapnik. (September 1995). ”Support-vector networks”, *Journal of Machine Learning*, Vol. 20, Issue 3, pp (273-297)

[15] Chen C, J. Twycross and JM Garibaldi. (2017). ”A new accuracy measure based on bounded relative error for time series forecasting”. *PLOS ONE* 12(3): e0174202. https://doi.org/10.1371/journal.pone.0174202

[16] Covas, F., B. Rump and E. Zakrajsek. (2014). ”Stress-testing U.S. Bank Holiding Companies: A Dynamic Panel Quantile Regression Approach”. *International Journal of Forecasting* 30(3), 691-713.

[17] Delgado M. and D. Amorim. (2014). ”Do we need Hundreds of Classifiers to Solve Real World Classification Problems?”, *Journal of Machine Learning Research* 15, 3133-318.

[18] European Central Bank. (2010). ”Beyond ROE: How to Measure Bank Performance”, Appendix to the report on EU banking structures.

[19] Fox, J. and S. Weisburg. (2013). ”Robust Regression”. School of Statistics, Univ. of Minnesota

[20] Glasserman, P. and G. Tangirala. (January 2016). ”Are the Fed’s Stress Test Results Predictable?” *The Journal of Alternative Investments*. Vol. 18, Issue 4, (Spring): pp.82-97.

[21] A. Graves. (June 2014). ”Generating Sequences With Recurrent Neural Networks”, Department of Computer Science, University of Toronto

[22] Greene, W. *Econometric Analysis*. (1997). 3rd ed. New Jersey:Prentice Hall.

[23] Grover, S. and M. McCracken. (2014). ”Factor-Based Prediction of Aggregate Bank Stress”. *Federal Reserve Bank of St. Louis Review* (Second Quarter) 96(2), pp. 173-93.

[24] Guerrieri, L. and M. Welch. (July 2012). ”Can Macro Variables Used in Stress Testing Forecast the Performance of Banks?”. *FEDS Working Paper* No. 2012-49.

[25] Hastie, T., R. Tisbshirani and J. Friedman. (2009). *The Elements of Statistical Learning*. 2nd ed. Springer Science+Business Media LLC.

[26] Hirtle B., A. Kovner, J. Vickery and M. Bhanot. (2016). ”Assessing financial stability: The Capital and Loss Assessment under Stress Scenarios (CLASS) model”. *Journal of Banking & Finance* 69 S35–S55

[27] Ho, T.S.Y. and A. Saunders. (1981). ”The Determinants of Bank Interest Rate Margins: Theory and Empirical Evidence”. *Journal of Financial and Quantitative Analysis* 16(4), 581-600.

[28] Hyndman Rob J. and Anne B. Koehler. (1992). ”Another look at measures of forecast accuracy”, *International Journal of Forecasting* Volume 8, Issue 1 (June) Pages 69-80

[29] Kilian L. and M. Taylor. (2003). ”why is it so difficult to beat the random walk forecast of exchange rates?”, *Journal of Int’l Economics* 60 (2003) 85–107.

[30] Kim, S. and C. Scott. (September 2012). ”Robust Kernel Density Estimation”, *The Journal of Machine Learning Research*, Vol. 13 Issue 1, pp2529-2565 .

[31] Kupiec, P. (2018). ”On the accuracy of alternative approaches for calibrating bank stress test models”. *Journal of Financial Stability*. JFS-639.

[32] Lin T., B. Horne, Peter Ti and C. Giles. (1996). ”Learning long-term dependencies in NARX recurrent neural networks”, IEEE Transactions on Neural Networks , vol. 7, no. 6, p.1329.

[33] Loh, W. (2002). ”Regression Trees with Unbiased Variable Selection and Interaction Detection”. *Statistica Sinica* 12. 361-386.

[34] Marcellino M, J. Stock and M. Watson. (2006). ”A Comparison of Direct and Iterated Multistep AR Methods for Forecasting Macroeconomic Time Series”. *Journal of Econometrics*. 135 :499-526.

[35] MacKay, DJC. (1992). ”Bayesian interpolation.” *Neural computation*. Vol. 4, No. 3, pp. 415-447.

[36] Matheron, G. (1963). ”Principles of geostatistics”. *Economic Geology*, 58, pp 1246-1266.

[37] McNeil, A. J., R. Frey and P. Embrechts. (2015). *Quantitative risk management: Concepts, techniques and tools*, Princeton Univ. Press, 2nd edition

[38] Platt J. (1998). ”Fast training of support vector machines using sequential minimal optimization”, In Scholkopf, B., C.J.C. Burges and A.J. Smola (eds). *Advances in Kernel Methods - Support Vector Learning*, pp. 185-208, MIT Press

[39] Rasmussen C. and C. Williams. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

[40] Scholkopf B., P. Bartlett., A. Smola and R. Williamson. (1998). ”Shrinking the tube: a new support vector regression algorithm”, In Kearns M. S., Solla M. S., and Cohn D. A., editors, *Advances in Neural Information Processing Systems*, 11. MIT Press.

[41] Schuermann, T. (2014). ”Stress Testing Banks”. *International Journal of Forecasting*. 30. 717-728.

[42] Tashman, Leonard J. (2000). ”Out-of-sample tests of forecasting accuracy: an analysis and review”, *International Journal of Forecasting* 16, 437–450.

[43] Tukey, J. (1977). *Exploratory Data Analysis*. Pearson. 1 ed. 1 (January).

[44] Zhang, G., B. Patuwo and M. Hu. (1998). ”Forecasting with artificial neural networks: The state of the art”, *International Journal of Forecasting* 14 35-62

A Feature Selections

In section A.1, we present the ten sets of feature variables or so-called Standard Feature Selections, to which we refer as ”FS1-FS10” in Table 1. The standard feature selection is based on economic principles or feature variables used in existing NIM literature. In section A.2, we present a list of additional feature selections that have been experimented with in this thesis but led to poorer forecasting accuracies and are presented for completeness. In section A.3, we present a list of untransformed feature variables representing each term of yield curves, from which PCA is conducted to extract principal components, which are then used as independent variables for Multiple Linear Regression to perform so-called Principal component regression. In section **??**, we present the list of candidate feature variables before we perform automated feature selection or Stepwise Selection as reported in Table 1.

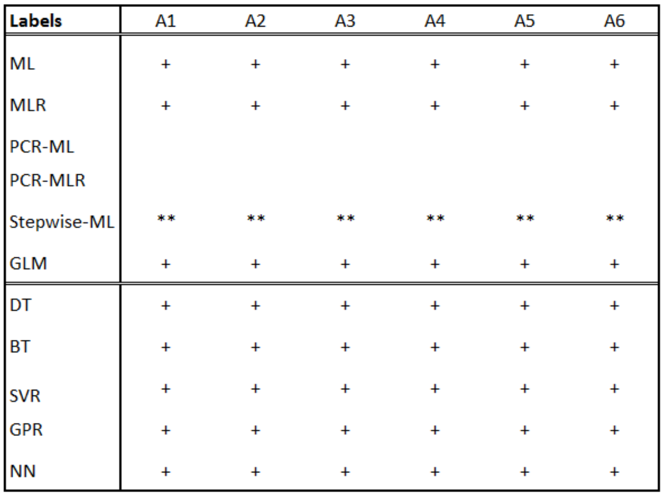
A.1 *Standard Feature Selections*

Corresponding with the 10 standard feature variable selections (or FS1-FS10) in Table 1, we provide descriptions as follows:

FS1: $\mathcal{X}_t = (y_{t-1}, \kappa_t, \Delta P_t)$
FS2: $\mathcal{X}_t = (y_{t-1}, T, \kappa_t, \Delta P_t)$
FS3: $\mathcal{X}_t = (y_{t-1}, \kappa_t, \Delta s_t)$
FS4: $\mathcal{X}_t = (y_{t-1}, \kappa_t, \Delta \sigma_t)$
FS5: $\mathcal{X}_t = (y_{t-1}, T, \kappa_t, r_{P,t})$
FS6: $\mathcal{X}_t = (y_{t-1}, T, \kappa_t, , \log(P_t))$
FS7: $\mathcal{X}_t = (y_{t-1}, T, \kappa_t, \log(s_t))$
FS8: $\mathcal{X}_t = (y_{t-1}, T, \kappa_t, r(s, t))$
FS9: $\mathcal{X}_t = (y_{t-1}, \kappa_t, \Delta PD_t, \Delta OC_t)$
FS10: $\mathcal{X}_t = (y_{t-1}, T, \kappa_t, \Delta PD_t, \Delta OC_t)$

Here we note that: y_{t-1} denotes the NIM observed for the quarter prior to quarter t; κ_t stands for the prior quarter's Slope of yield curve calculated as the difference between 10-year yield and 3-month yield; We calculate quarterly change, quarter-specific logarithm and quarter over quarter Growth Rate for all three credit-risk indicators, P_t, s_t and σ_t, represented by S&P 500 Equity Index, Bank of America Investment Grade Bond Spread Index, VIX (Implied Volatility) Index respectively. As a result, take P_t for example, we denote them by ΔP_t, $log(P_t)$ and $r_{P,t}$ for the three derived feature variables; We include bank-specific variable, i.e., the change of a bank's own ΔPD_t together with one or more general credit-risk indicators; We include another bank-specific variable, i.e., the change of a bank's own *Operating Costs to Asset Ratio* denoted by ΔOC_t to indicate Bank-specific operating costs; T stands for the dummy variables indicating the quarter from which the data are collected, i.e., $1, 2, 3, 4$; For the last two equations above, we add two more macroeconomic variables, i.e., *GDP* and *CPI* for investigation.

Table 5: A List of 54 Additional Models



Labels	A1	A2	A3	A4	A5	A6
ML	+	+	+	+	+	+
MLR	+	+	+	+	+	+
PCR-ML						
PCR-MLR						
Stepwise-ML	**	**	**	**	**	**
GLM	+	+	+	+	+	+
DT	+	+	+	+	+	+
BT	+	+	+	+	+	+
SVR	+	+	+	+	+	+
GPR	+	+	+	+	+	+
NN	+	+	+	+	+	+

A.2 *Additional models*

In addition to the standard feature selections above, we have investigated the following 54 models trained in the Machine Learning-regression algorithms discussed in Section 2. Typically, these feature selections are transformed (see below) from standard feature selections and have led to poorer forecasting accuracies. They are presented in this section for completeness.

1. As for ML models, each additional feature selections (i.e., A1-A6 in Table 5) is based on FS5-FS10 from the Standard Feature Selections above, constructed as the squared term of feature variables. For instance, for the x_1 used in a standard feature selection, x_1^2 appears in an additional feature selection.

2. As regards MLR, the additional six models marked as ”A1-A6” in Table 5 follows the same variable transformation for ML above except that the Robust version of ML is applied.

3. As for Stepwise Regression, 10 standard feature selections are conducted using ”bic” as the selection criterion; additionally, we investigated forecasting accuracies using different selection criteria, i.e., ”aic”, ”sse”, ”adjusted-square”, represented by A1-A6 in Table 5.

4. Regarding GLM regression, we experiment with different assumption for response variable, for which we assume Normal; again corresponding with FS5-FS10 indicated above, we have six additional models.

5. Regarding Regression Tree, the 10 Regression Tree models discussed in Section 2.7.1 assumes no interactions between feature variables; in contrast, additional models assume there is interactions between feature variables, which leads to slower speed of the execution. Again, corresponding with FS5-FS10 indicated above, we have six additional models.

6. Regarding Ensemble/Bagged Tree, based on FS5-FS10, we use different # of Learning Cycles to see the impacts. As a result, we have six additional models marked by A1-A5 in Table 5.

7. Regarding Gaussian Process Regression, based on FS5-FS10, we experiment with ”squared exponential” Kernel function with feature-specific length scale, which leads to six additional models marked by A1-A5 in Table 5.

8. Regarding Support Vector Regression, based on FS5-FS10, we experiment with different choices of Kernel functions namely, Gaussian, Polynomial of order 3, RBF, which leads to six additional models marked by A1-A5 in Table 5.

9. Regarding NARX, instead of using Bayesian regularized backpropagation during training, based on FS5-FS10, we experiment with using Levenberg-Marquardt backpropagation, which leads to six additional models marked by A1-A5 in Table 5.

A.3 *PCR & Stepwise Regression*

As a robustness study, we use PCR based on the principal components from Macroeconomic Factors, i.e.,$\mathcal{X}_t^d = \left(y_{t-1}, r_{3m,t}, r_{1y,t}, r_{2y,t}, r_{3y,t}, r_{4y,t}, r_{5y,t}, r_{7y,t}, r_{10y,t}, r_{15y,t}, r_{20y,t}\right)$ as: $\mathcal{X}_t^p = (p_1, p_2, \ldots, p_m)$.

In the context of ML (Multiple Linear Regression) and MLR (Multiple Linear Robust Regression) respectively, we studied so-called PCR-ML and PCR-MLR. The descriptions for Feature Selections apply to both PCR-ML and PCR-MLR models, where y_{t-1} stands for the NIM observed at the end of each Rolling Windows the time t as explained in Section 2.11, prior to Forecast Horizons $h := \{1, \ldots, H\}$; The rest of variables in the first equation above stand for treasury yield variables over different terms, i.e., 1-year, 2-year, etc., all observable for the Rolling Windows and used for training PCR-ML and PCR-MLR models; The second equation above shows the Feature Selection $\mathcal{X}_t^p$ for PCR when the number of principal components included in regression model is m. In this paper, we explore the performances for PCR models in response to the number of Principals to keep, i.e., $m = \{3, 4, 5, \ldots, 11\}$; in our study, we experiment $m := \{3, \ldots, 11\}$ for the number of principal components to keep for Regression study. Clearly, the principle component analysis study can be extended to other Regression models.

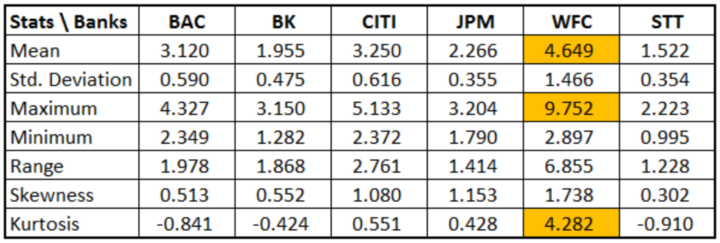
As a second robustness study, we studied the following 10 Feature Selections as the candidate variable list for Stepwise Regression.

FS1: $\mathcal{X}_t = (y_{t-1}, \kappa_t, \Delta P_t, log(P_t), r_{P,t}, P_t)$
FS2: $\mathcal{X}_t = (y_{t-1}, \kappa_t, \Delta s_t, log(s_t), r_{s,t}, s_t)$
FS3: $\mathcal{X}_t = (y_{t-1}, \kappa_t, \Delta \sigma_t, log(\sigma_t), r_{\sigma,t}, \sigma_t)$
FS4: $\mathcal{X}_t = (y_{t-1}, T, \kappa_t, \Delta P_t, log(P_t), r_{P,t}, P_t, \Delta s_t, log(s_t), r_{s,t}, s_t)$
FS5: $\mathcal{X}_t = (y_{t-1}, T, \kappa_t, \Delta P_t, log(P_t), r_{P,t}, P_t, \Delta \sigma_t, log(\sigma_t), r_{\sigma,t}, \sigma_t)$
FS6: $\mathcal{X}_t = (y_{t-1}, \kappa_t, \Delta PD_t, \Delta OC_t, \Delta P_t, \Delta s_t, \Delta \sigma_t)$
FS7: $\mathcal{X}_t = (y_{t-1}, T, \kappa_t, \Delta PD_t, \Delta OC_t, \Delta P_t, \Delta s_t, \Delta \sigma_t)$
FS8: $\mathcal{X}_t = (y_{t-1}, T, \kappa_t, \Delta PD_t, \Delta OC_t, \Delta P_t, P_t, \Delta s_t, st, \Delta \sigma_t, \sigma_t)$
FS9: $\mathcal{X}_t = (y_{t-1}, T, \kappa_t, \Delta PD_t, \Delta OC_t, \Delta P_t, GDP_t, r_{GDP,t}, \Delta s_t, \Delta \sigma_t)$
FS10: $\mathcal{X}_t = \left(y_{t-1}, T, \kappa_t, \Delta PD_t, \Delta OC_t, \Delta P_t, GDP_t, r_{gdp,t}, CPI_t, r_{cpi,t}, \Delta s_t, \Delta \sigma_t\right)$

Here we note that:

- Again, y_{t-1} denotes the NIM observed for the quarter prior to prediction quarters.
- κ_t stands for the prior quarter's Slope of yield curve calculated as the difference between 10-year yield and 3-month yield.
- We calculate quarterly change, quarter-specific logarithm and quarter over quarter Growth Rate for all three credit-risk indicators, P_t, s_t and σ_t, represented by S&P 500 Equity Index, Bank of America Investment Grade Bond Spread Index, VIX (Implied Volatility) Index respectively. As a result, take P_t for example, we denote them by ΔP_t, $log(P_t)$ and $r_{P,t}$ for the three derived feature variables.
- We include bank-specific variable, i.e., the change of a bank's own ΔPD_t together with one or more general credit-risk indicators.
- We include another bank-specific variable, i.e., the change of a bank's own *Operating Costs to Asset Ratio* denoted by ΔOC_t to indicate Bank-specific operating costs.
- T stands for the dummy variables indicating the quarter from which the data are collected, i.e., 1, 2, 3, 4.
- For the last two equations above, we add two more macroeconomic variables, i.e., *GDP* and *CPI* for investigation.

Table 6: Summary Statistics for Six G-SiB Banks in Percentages



Stats \ Banks	BAC	BK	CITI	JPM	WFC	STT
Mean	3.120	1.955	3.250	2.266	4.649	1.522
Std. Deviation	0.590	0.475	0.616	0.355	1.466	0.354
Maximum	4.327	3.150	5.133	3.204	9.752	2.223
Minimum	2.349	1.282	2.372	1.790	2.897	0.995
Range	1.978	1.868	2.761	1.414	6.855	1.228
Skewness	0.513	0.552	1.080	1.153	1.738	0.302
Kurtosis	-0.841	-0.424	0.551	0.428	4.282	-0.910

B Summary of Data

B.1 *Data Sources*

We source all our data from public available data based on a combination of Bloomberg^{TM}, Thomson Reuters^{TM}and US Federal Reserve. The time series data sourced range from the beginning of 2000 to the end of 2016 on quarterly basis as explained below:

- *Response Variable or NIM Data*: we collect the NIM time series data for six out of the eight so-called G-SiBs) by eliminating Goldman Sachs and Morgan Stanley because the two banks only become bank holding companies in recent years; the authors in Covas et al (2012, [16]) cited the same reason for data elimination.
- We choose to focus our study on G-SiBs US banks because the vast majority of current NIM literature focus on US banks; thus, it is convenient to compare forecasting performances among different literature. Also, US banks tend to have historical data for NIM more readily available.
- *Macroeconomic Variables*: we have compiled the complete list of Macroeconomic variables available from Bank of England's website dedicated to Stress Testing[15]; as well, we enrich our data from Bloomberg Terminal's for the rest of Macroeconomic variables.
- *Data Sampling Windows*: our data sampling window starts from 2000Q1 and ends at 2016Q4; altogether, we have 68 quarterly time series data for the response variable, i.e., NIM as well as the explanatory variables, i.e., Macroeconomic variables.
- In all our models, the response variable is $y_t = NIM_t$; the explanatory variables are based on: the autoregressive term, y_{t-1}, $Slope := Yield_{10Y} - Yield_{3M}$ (10-year Treasury Yield minus 3-month Treasury Yield) and $P_t - P_{t_1}$ (S&P 500 Index quarterly variations).

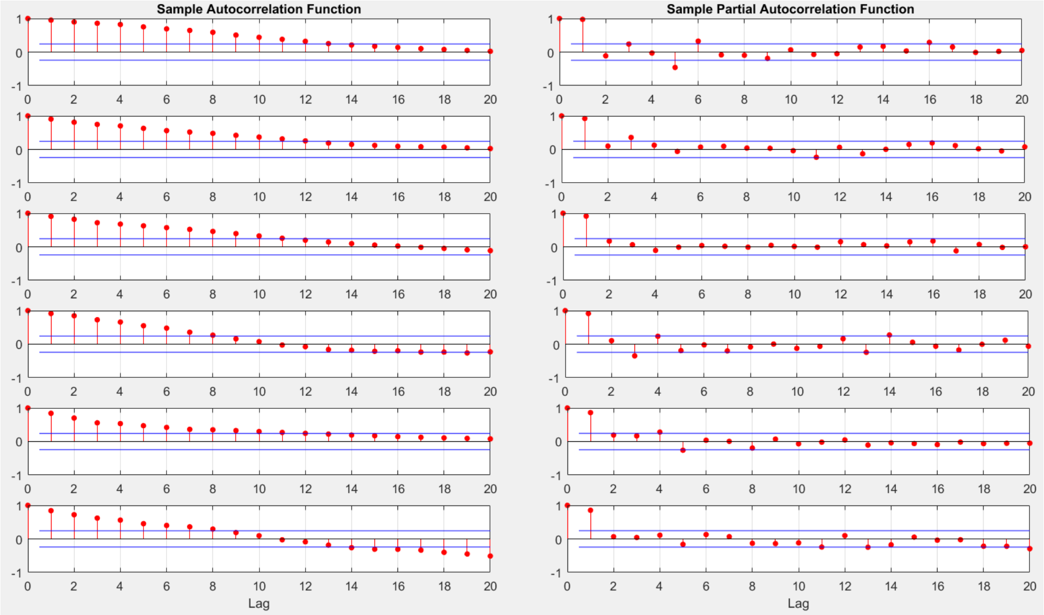
There are eight recognized so-called *G*lobal *S*ystematically *I*mportant (US) *B*anks (or simply G-SIBs); eliminating Goldman Sachs and Morgan Stanley, which have become Bank Holding companies in recent years, we study the six G-SIBs US banks.

B.2 *NIM Summary Statistics*

Table 6 presents the Summary Statistics for the six US banks, which we represent by their stock tickers as the short names: ”BAC” for Bank of American; ”BK” for Bank of New York Mellon; ”CITI” for Citigroup; ”JPM” for JP Morgan; ”WFC” for Wells Fargo Bank; ”STT” for State Street Bank. We note that the NIM of WFC displays a high mean (4.649), high volatility (9.752), significantly leptokurtic (Kurtosis=4.282) and positively skewed (Skewness=1.738). Figure 6 presents the sample Auto Correlations (left, ACF) and Partial Auto Correlations (right, PACF) calculated from each of the six banks in the order presented in Table 6. Clearly, one can see the PACF starts to fall within the 95% confidence intervals represented by the blue lines.

[15] Bank of England. 2017. `https://bit.ly/2nrpxVM`

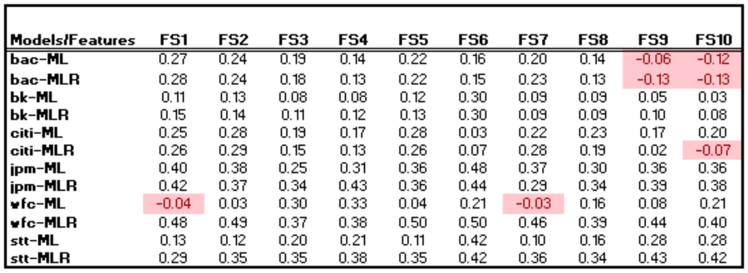
Figure 6: Autocorrelation and Partial Autocorrelation for Six Banks



Sample Autocorrelation Function

Sample Partial Autocorrelation Function

Table 7: Performance Improvement by AFPIR for ML and MLR related models: Numerical Results



Models/Features	FS1	FS2	FS3	FS4	FS5	FS6	FS7	FS8	FS9	FS10
bac-ML	0.27	0.24	0.19	0.14	0.22	0.16	0.20	0.14	-0.06	-0.12
bac-MLR	0.28	0.24	0.18	0.13	0.22	0.15	0.23	0.13	-0.13	-0.13
bk-ML	0.11	0.13	0.08	0.08	0.12	0.30	0.09	0.09	0.05	0.03
bk-MLR	0.15	0.14	0.11	0.12	0.13	0.30	0.09	0.09	0.10	0.08
citi-ML	0.25	0.28	0.19	0.17	0.28	0.03	0.22	0.23	0.17	0.20
citi-MLR	0.26	0.29	0.15	0.13	0.26	0.07	0.28	0.19	0.02	-0.07
jpm-ML	0.40	0.38	0.25	0.31	0.36	0.48	0.37	0.30	0.36	0.36
jpm-MLR	0.42	0.37	0.34	0.43	0.36	0.44	0.29	0.34	0.39	0.38
wfc-ML	-0.04	0.03	0.30	0.33	0.04	0.21	-0.03	0.16	0.08	0.21
wfc-MLR	0.48	0.49	0.37	0.38	0.50	0.50	0.46	0.39	0.44	0.40
stt-ML	0.13	0.12	0.20	0.21	0.11	0.42	0.10	0.16	0.28	0.28
stt-MLR	0.29	0.35	0.35	0.38	0.35	0.42	0.36	0.34	0.43	0.42

C Model-specific Empirical Results by Figures and Tables

Figure 7: Performance Improvement by AFPIR for ML and MLR related models

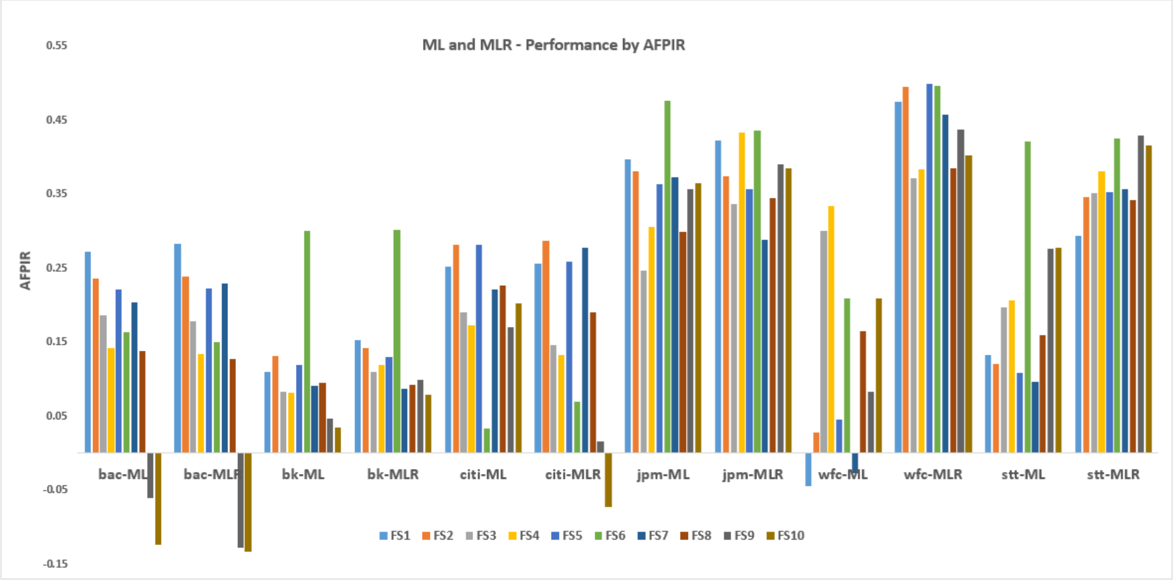


Figure 8: Generalized Linear Model - Gamma Performance by APFIR

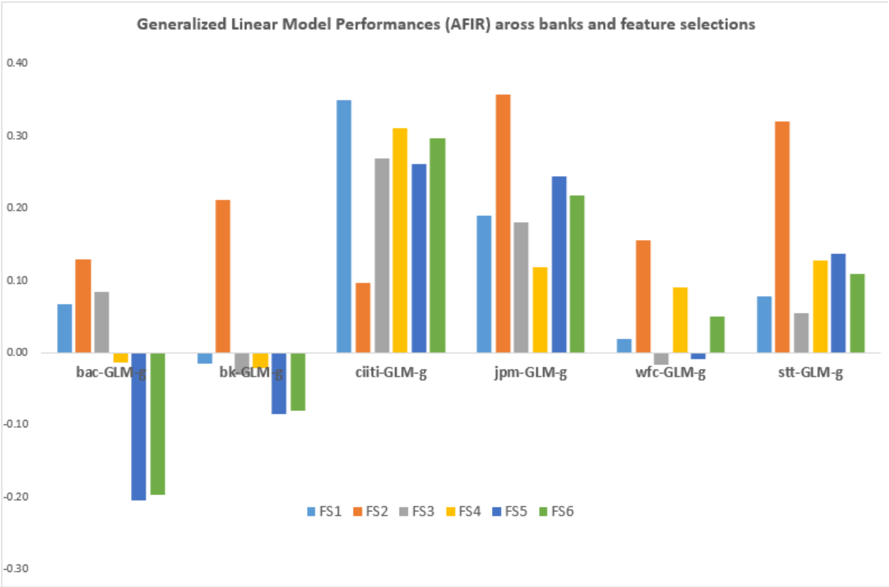
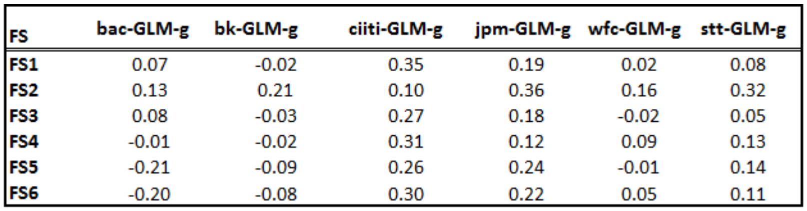


Table 8: Generalized Linear Model - Gamma Performance by APFIR: Numbers



FS	bac-GLM-g	bk-GLM-g	ciiti-GLM-g	jpm-GLM-g	wfc-GLM-g	stt-GLM-g
FS1	0.07	-0.02	0.35	0.19	0.02	0.08
FS2	0.13	0.21	0.10	0.36	0.16	0.32
FS3	0.08	-0.03	0.27	0.18	-0.02	0.05
FS4	-0.01	-0.02	0.31	0.12	0.09	0.13
FS5	-0.21	-0.09	0.26	0.24	-0.01	0.14
FS6	-0.20	-0.08	0.30	0.22	0.05	0.11

Figure 9: Regression Tree Performance Comparison by APFIR

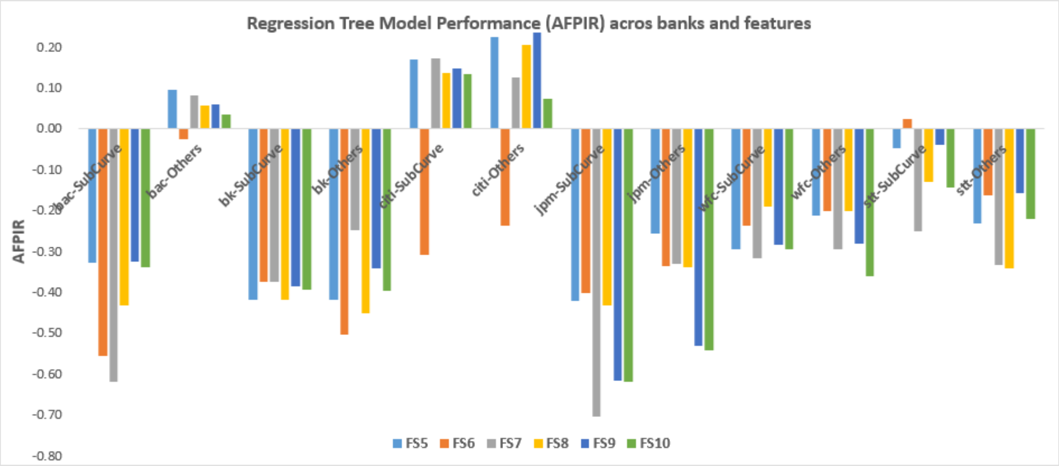
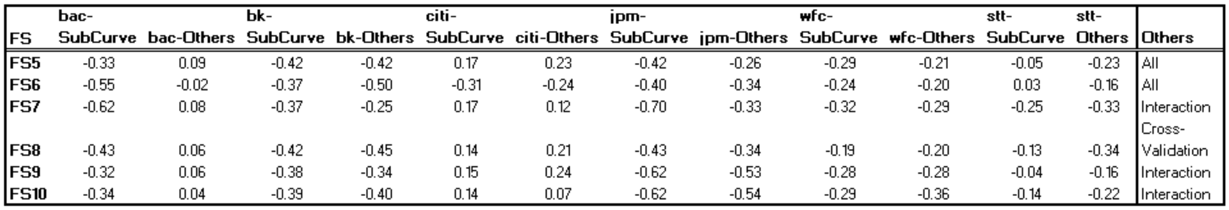


Table 9: Regression Tree Performance Comparison by APFIR: Numbers



FS	bac-SubCurve	bac-Others	bk-SubCurve	bk-Others	citi-SubCurve	citi-Others	jpm-SubCurve	jpm-Others	wfc-SubCurve	wfc-Others	stt-SubCurve	stt-Others	Others
FS5	-0.33	0.09	-0.42	-0.42	0.17	0.23	-0.42	-0.26	-0.29	-0.21	-0.05	-0.23	All
FS6	-0.55	-0.02	-0.37	-0.50	-0.31	-0.24	-0.40	-0.34	-0.24	-0.20	0.03	-0.16	All
FS7	-0.62	0.08	-0.37	-0.25	0.17	0.12	-0.70	-0.33	-0.32	-0.29	-0.25	-0.33	Interaction
FS8	-0.43	0.06	-0.42	-0.45	0.14	0.21	-0.43	-0.34	-0.19	-0.20	-0.13	-0.34	Cross-Validation
FS9	-0.32	0.06	-0.38	-0.34	0.15	0.24	-0.62	-0.53	-0.28	-0.28	-0.04	-0.16	Interaction
FS10	-0.34	0.04	-0.39	-0.40	0.14	0.07	-0.62	-0.54	-0.29	-0.36	-0.14	-0.22	Interaction

Figure 10: Ensemble with Regression Trees' Performance Comparison by APFIR

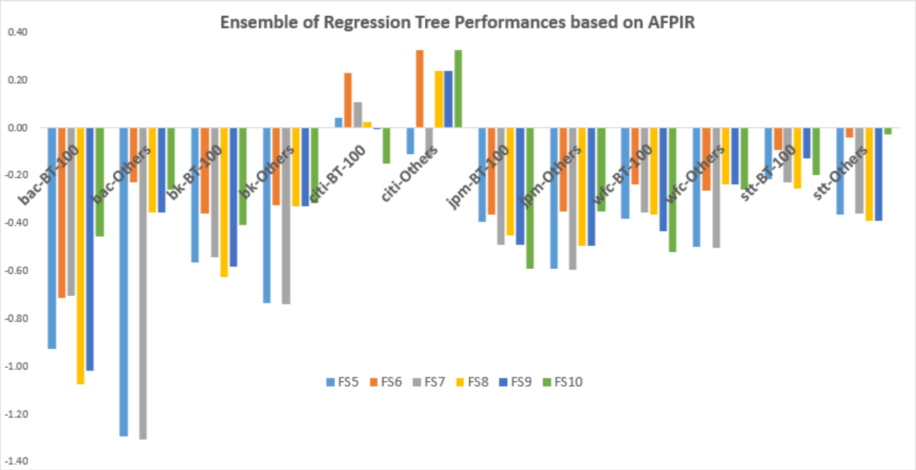
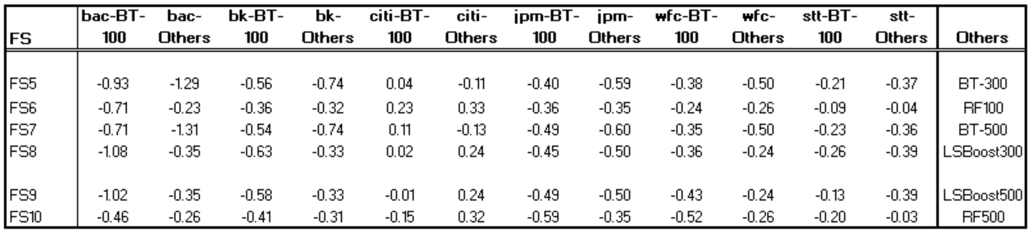


Table 10: Ensemble with Regression Trees' Performance Comparison by APFIR: Numbers



FS	bac-BT-100	bac-Others	bk-BT-100	bk-Others	citi-BT-100	citi-Others	jpm-BT-100	jpm-Others	wfc-BT-100	wfc-Others	stt-BT-100	stt-Others	Others
FS5	-0.93	-1.29	-0.56	-0.74	0.04	-0.11	-0.40	-0.59	-0.38	-0.50	-0.21	-0.37	BT-300
FS6	-0.71	-0.23	-0.36	-0.32	0.23	0.33	-0.36	-0.35	-0.24	-0.26	-0.09	-0.04	RF100
FS7	-0.71	-1.31	-0.54	-0.74	0.11	-0.13	-0.49	-0.60	-0.35	-0.50	-0.23	-0.36	BT-500
FS8	-1.08	-0.35	-0.63	-0.33	0.02	0.24	-0.45	-0.50	-0.36	-0.24	-0.26	-0.39	LSBoost300
FS9	-1.02	-0.35	-0.58	-0.33	-0.01	0.24	-0.49	-0.50	-0.43	-0.24	-0.13	-0.39	LSBoost500
FS10	-0.46	-0.26	-0.41	-0.31	-0.15	0.32	-0.59	-0.35	-0.52	-0.26	-0.20	-0.03	RF500

Figure 11: Gaussian Process Regression Performance Comparison by APFIR w.r.t. Kernel Functions

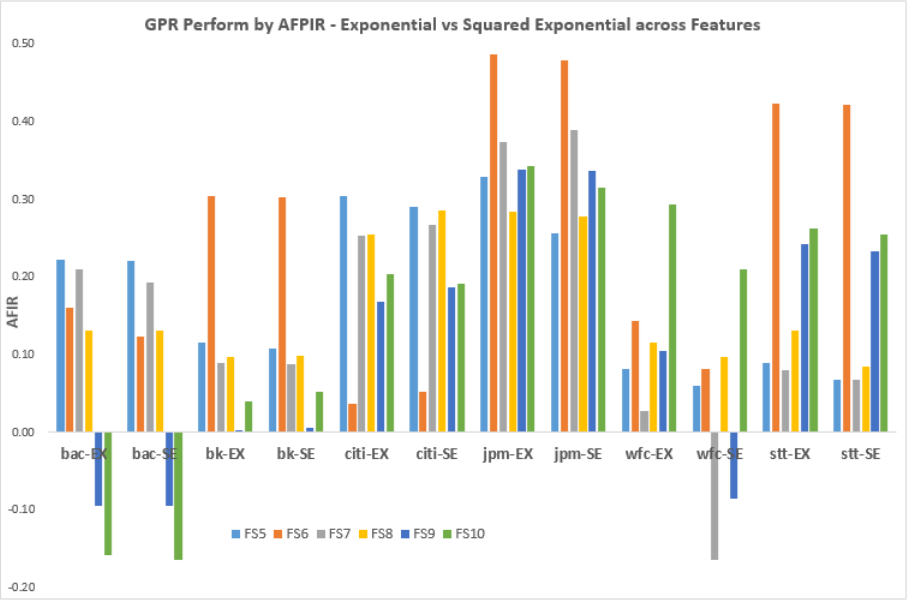
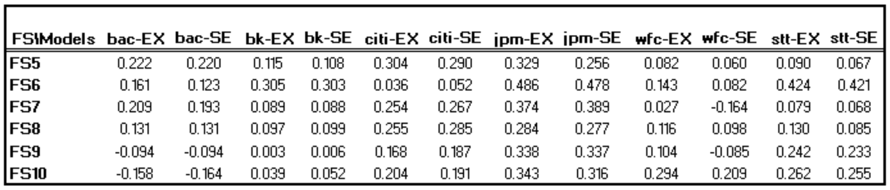


Table 11: Gaussian Process Regression Performance Comparison by APFIR w.r.t. Kernel Functions: Numbers



FS\Models	bac-EX	bac-SE	bk-EX	bk-SE	citi-EX	citi-SE	jpm-EX	jpm-SE	wfc-EX	wfc-SE	stt-EX	stt-SE
FS5	0.222	0.220	0.115	0.108	0.304	0.290	0.329	0.256	0.082	0.060	0.090	0.067
FS6	0.161	0.123	0.305	0.303	0.036	0.052	0.486	0.478	0.143	0.082	0.424	0.421
FS7	0.209	0.193	0.089	0.088	0.254	0.267	0.374	0.389	0.027	-0.164	0.079	0.068
FS8	0.131	0.131	0.097	0.099	0.255	0.285	0.284	0.277	0.116	0.098	0.130	0.085
FS9	-0.094	-0.094	0.003	0.006	0.168	0.187	0.338	0.337	0.104	-0.085	0.242	0.233
FS10	-0.158	-0.164	0.039	0.052	0.204	0.191	0.343	0.316	0.294	0.209	0.262	0.255

Figure 12: Support Vector Regression Performance Comparison by APFIR w.r.t. Kernel Choices

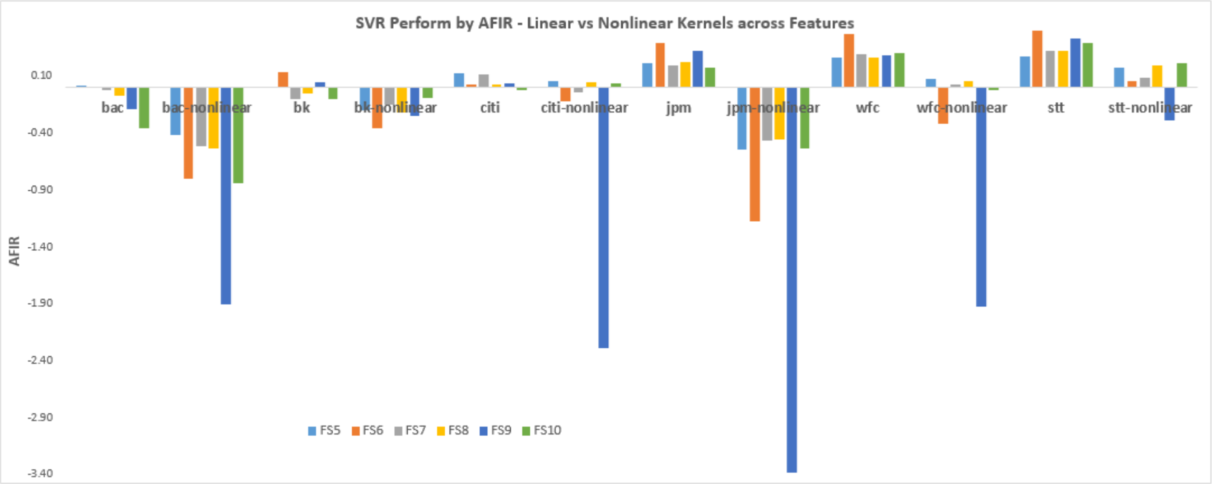
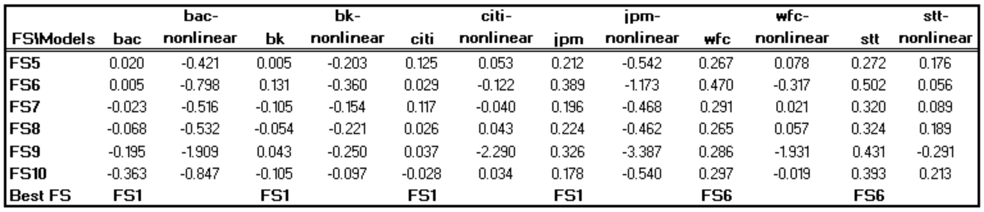


Table 12: Support Vector Regression Performance Comparison by APFIR w.r.t. Kernels: Numbers



FS\Models	bac	bac-nonlinear	bk	bk-nonlinear	citi	citi-nonlinear	jpm	jpm-nonlinear	wfc	wfc-nonlinear	stt	stt-nonlinear
FS5	0.020	-0.421	0.005	-0.203	0.125	0.053	0.212	-0.542	0.267	0.078	0.272	0.176
FS6	0.005	-0.798	0.131	-0.360	0.029	-0.122	0.389	-1.173	0.470	-0.317	0.502	0.056
FS7	-0.023	-0.516	-0.105	-0.154	0.117	-0.040	0.196	-0.468	0.291	0.021	0.320	0.089
FS8	-0.068	-0.532	-0.054	-0.221	0.026	0.043	0.224	-0.462	0.265	0.057	0.324	0.189
FS9	-0.195	-1.909	0.043	-0.250	0.037	-2.290	0.326	-3.387	0.286	-1.931	0.431	-0.291
FS10	-0.363	-0.847	-0.105	-0.097	-0.028	0.034	0.178	-0.540	0.297	-0.019	0.393	0.213
Best FS	FS1		FS1		FS1		FS1		FS6		FS6	

Figure 13: Neural Network Performance Comparison by APFIR for All Banks

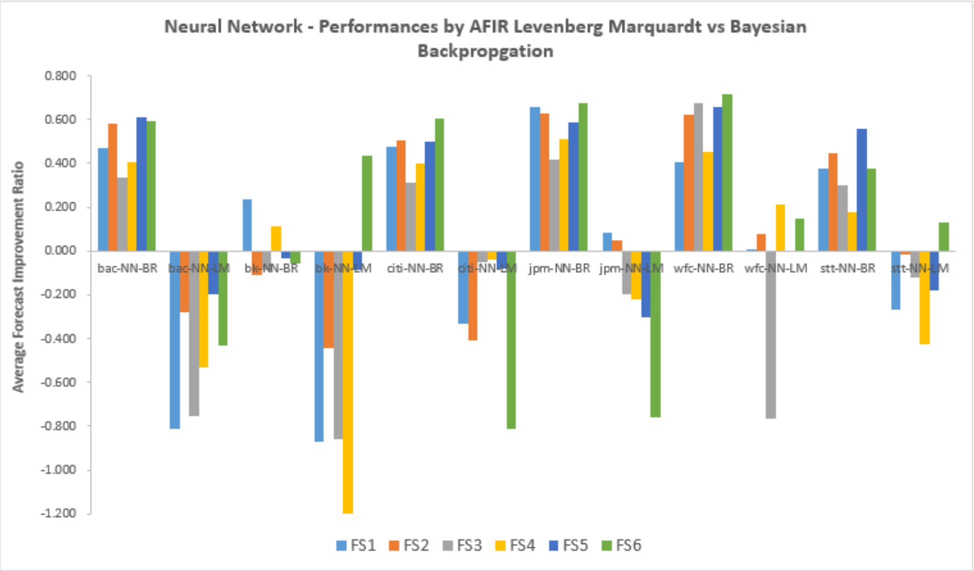
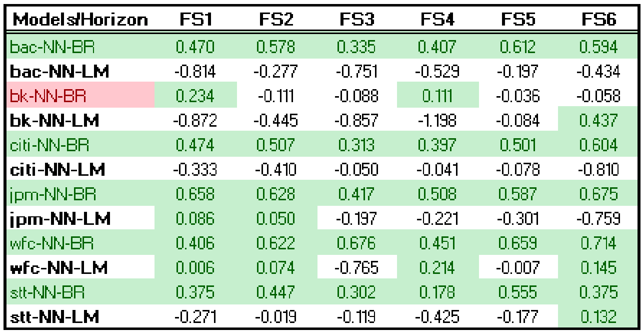
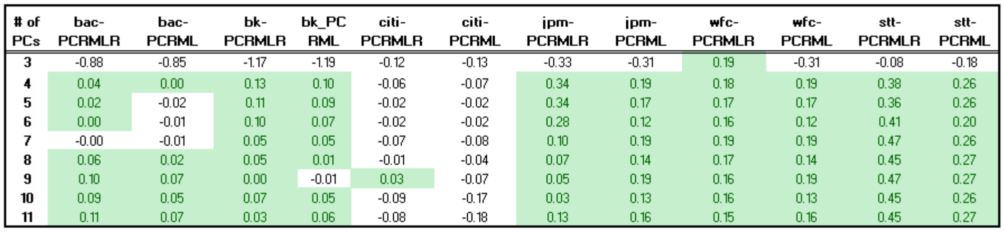


Table 13: Neural Network Performance Comparison by APFIR: Numbers



Models\Horizon	FS1	FS2	FS3	FS4	FS5	FS6
bac-NN-BR	0.470	0.578	0.335	0.407	0.612	0.594
bac-NN-LM	-0.814	-0.277	-0.751	-0.529	-0.197	-0.434
bk-NN-BR	0.234	-0.111	-0.088	0.111	-0.036	-0.058
bk-NN-LM	-0.872	-0.445	-0.857	-1.198	-0.084	0.437
citi-NN-BR	0.474	0.507	0.313	0.397	0.501	0.604
citi-NN-LM	-0.333	-0.410	-0.050	-0.041	-0.078	-0.810
jpm-NN-BR	0.658	0.628	0.417	0.508	0.587	0.675
jpm-NN-LM	0.086	0.050	-0.197	-0.221	-0.301	-0.759
wfc-NN-BR	0.406	0.622	0.676	0.451	0.659	0.714
wfc-NN-LM	0.006	0.074	-0.765	0.214	-0.007	0.145
stt-NN-BR	0.375	0.447	0.302	0.178	0.555	0.375
stt-NN-LM	-0.271	-0.019	-0.119	-0.425	-0.177	0.132

Table 14: Performance Comparison by APFIR for Principal Component Regressions with/without Robust Regression w.r.t. # of PCs: Numbers



# of PCs	bac-PCRMLR	bac-PCRML	bk-PCRMLR	bk_PCRML	citi-PCRMLR	citi-PCRML	jpm-PCRMLR	jpm-PCRML	wfc-PCRMLR	wfc-PCRML	stt-PCRMLR	stt-PCRML
3	-0.88	-0.85	-1.17	-1.19	-0.12	-0.13	-0.33	-0.31	0.19	-0.31	-0.08	-0.18
4	0.04	0.00	0.13	0.10	-0.06	-0.07	0.34	0.19	0.18	0.19	0.38	0.26
5	0.02	-0.02	0.11	0.09	-0.02	-0.02	0.34	0.17	0.17	0.17	0.36	0.26
6	0.00	-0.01	0.10	0.07	-0.02	-0.02	0.28	0.12	0.16	0.12	0.41	0.20
7	-0.00	-0.01	0.05	0.05	-0.07	-0.08	0.10	0.19	0.19	0.19	0.47	0.26
8	0.06	0.02	0.05	0.01	-0.01	-0.04	0.07	0.14	0.17	0.14	0.45	0.27
9	0.10	0.07	0.00	-0.01	0.03	-0.07	0.05	0.19	0.16	0.19	0.47	0.27
10	0.09	0.05	0.07	0.05	-0.09	-0.17	0.03	0.13	0.16	0.13	0.45	0.26
11	0.11	0.07	0.03	0.06	-0.08	-0.18	0.13	0.16	0.15	0.16	0.45	0.27

Figure 14: Performance Comparison by APFIR for Principal Component Regressions with/wo Robust Regression w.r.t. # of PCs

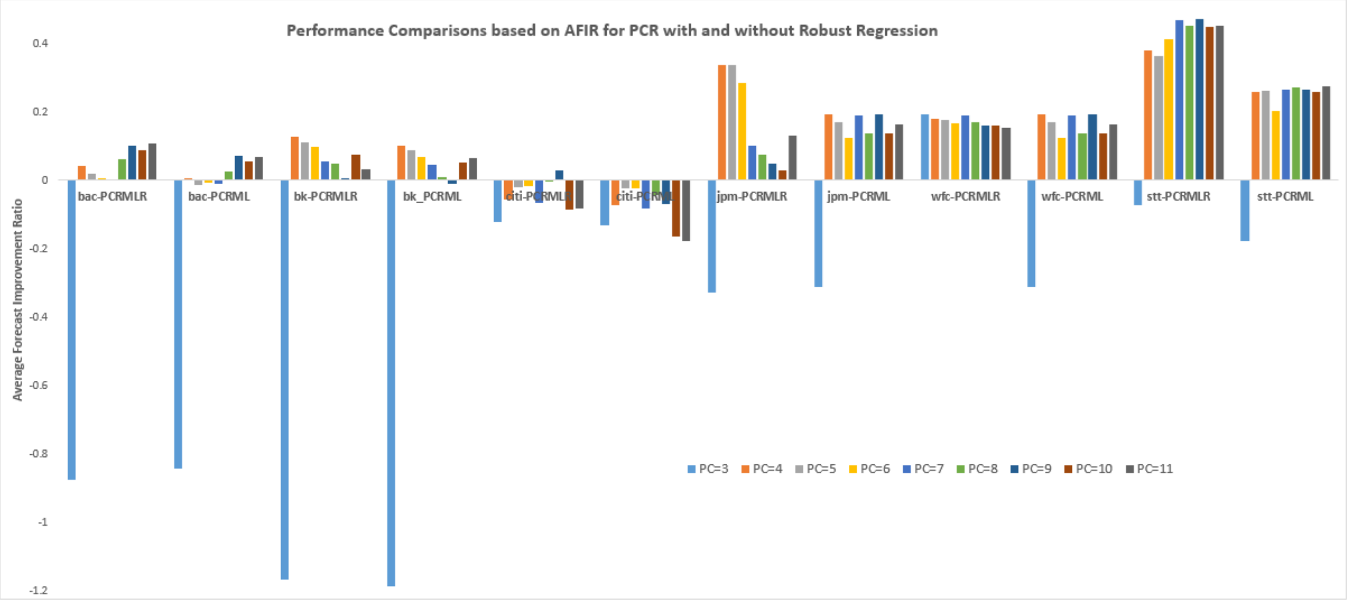
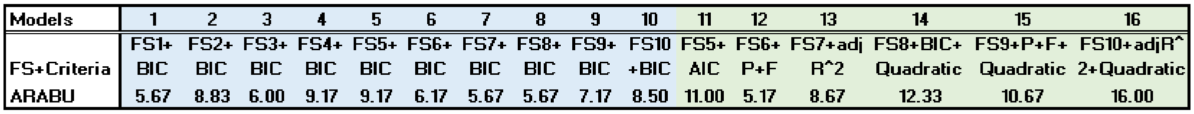


Table 15: Rank ordering of Feature Selections for Stepwise Regression based on ARABU



Models	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
FS+Criteria	FS1+ BIC	FS2+ BIC	FS3+ BIC	FS4+ BIC	FS5+ BIC	FS6+ BIC	FS7+ BIC	FS8+ BIC	FS9+ BIC	FS10 +BIC	FS5+ AIC	FS6+ P+F	FS7+adj R^2	FS8+BIC+ Quadratic	FS9+P+F+ Quadratic	FS10+adjR^ 2+Quadratic
ARABU	5.67	8.83	6.00	9.17	9.17	6.17	5.67	5.67	7.17	8.50	11.00	5.17	8.67	12.33	10.67	16.00

Figure 15: Stepwise Regression Performance Comparison by APFIR w.r.t. Feature Selections

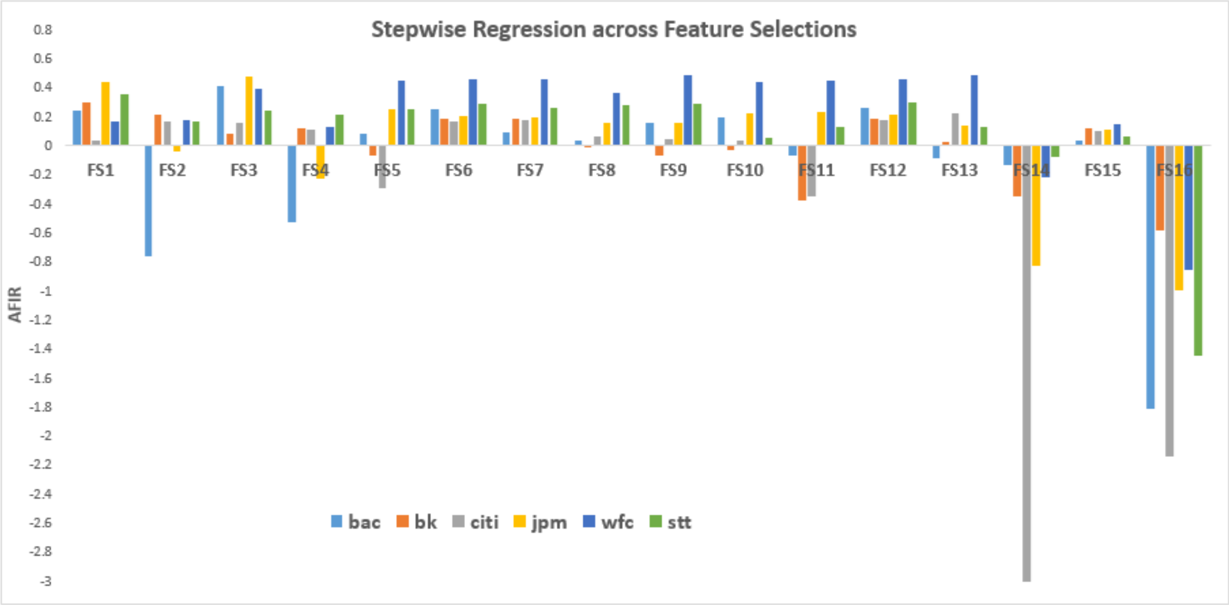
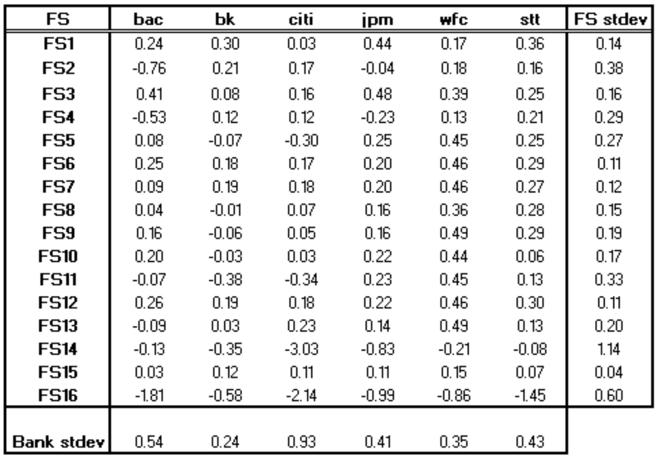


Table 16: Stepwise Regression Performance Comparison Numerical Results by APFIR w.r.t. Feature Selections



FS	bac	bk	citi	jpm	wfc	stt	FS stdev
FS1	0.24	0.30	0.03	0.44	0.17	0.36	0.14
FS2	-0.76	0.21	0.17	-0.04	0.18	0.16	0.38
FS3	0.41	0.08	0.16	0.48	0.39	0.25	0.16
FS4	-0.53	0.12	0.12	-0.23	0.13	0.21	0.29
FS5	0.08	-0.07	-0.30	0.25	0.45	0.25	0.27
FS6	0.25	0.18	0.17	0.20	0.46	0.29	0.11
FS7	0.09	0.19	0.18	0.20	0.46	0.27	0.12
FS8	0.04	-0.01	0.07	0.16	0.36	0.28	0.15
FS9	0.16	-0.06	0.05	0.16	0.49	0.29	0.19
FS10	0.20	-0.03	0.03	0.22	0.44	0.06	0.17
FS11	-0.07	-0.38	-0.34	0.23	0.45	0.13	0.33
FS12	0.26	0.19	0.18	0.22	0.46	0.30	0.11
FS13	-0.09	0.03	0.23	0.14	0.49	0.13	0.20
FS14	-0.13	-0.35	-3.03	-0.83	-0.21	-0.08	1.14
FS15	0.03	0.12	0.11	0.11	0.15	0.07	0.04
FS16	-1.81	-0.58	-2.14	-0.99	-0.86	-1.45	0.60
Bank stdev	0.54	0.24	0.93	0.41	0.35	0.43	