UTRECHT UNIVERSITY

Comparing the Performance of XGBoost and Random Forest Models for Predicting Company Cash Position

By

Luc Lubbers

A THESIS FOR THE DEGREE OF
MSc in Applied Data Science

Utrecht, Netherlands

SUPERVISORS
Prof. dr. Y. (Yannis) Velegrakis
V. (Vahid) Shahrivari Joghan, MSc

July 2023

Abstract

This study explores the use of XGBoost and Random Forest machine learning algorithms for predicting cash flow using transaction data from small and medium-sized enterprises (SMEs). The research aimed to identify performance differences between the two algorithms and assess their feasibility for practical use by accountants in their daily operations. While both algorithms showed potential, the Random Forest model marginally outperformed XGBoost, but the performance varied depending on the training data used. Despite XGBoost exhibiting a faster model training process, neither model yielded predictions reliable enough for practical use by accountants, with an average error rate of approximately 101.94% of the target variable's average magnitude. The complex nature of company's transactions and the limitations in the dataset used could have contributed to this low performance. This research contributes new insights to the domain of cash flow prediction, highlighting the need for more accurate and reliable machine learning models for this purpose and suggesting a potential path for further research to explore other models and incorporate additional company-related features. However, a more critical aspect to address would be the enhancement of data quality and the identification of clearer patterns within the dataset, as these factors significantly influence the predictive performance. These findings can guide future investigations and efforts in not only improving cash flow predictions for SME, but also advancing the broader field of time series forecasting.

Table of Contents

1. Introduction

1.1 Introduction

Could machine learning algorithms, such as XGBoost and Random Forest, represent a new frontier in improving the precision of cash flow predictions, thereby offering accountants and finance controllers better insights into a company's financial future?
Cash flows are critical financial indicators that are derived from a company's operating, investing, and financing activities, signifying the inflow and outflow of cash in a business (Needles et al., 1999). The available cash in a company's coffers can significantly fluctuate over time due to the company's payment and collection cycles.

An accurate cash flow estimation empowers accountants and finance controllers to make vital financial decisions that can ultimately dictate the survival of a business. In addition to understanding the current cash flow situation, businesses also require reliable predictions of future cash flows within a specified time frame. These predictive insights assist accountants and financial managers in making informed decisions, such as ensuring continuous operations by facilitating complete and timely payments of necessary operational capital. The ability to predict cash flow is not only integral to business decision-making, but also pivotal for potential lenders. Understanding a company's cash flow status enables them to adhere to sound lending principles, mitigating risks associated with bad loans (Fight, 2005). In sum, cash flow estimation and prediction are indispensable tools for financial stability and strategic planning.

Financial forecasting has a long history. In 1966, Beaver used financial ratios to predict business distress, analyzing successful and failed firms (Beaver, 1966). Since then, researchers have built models to predict a company's financial health, using techniques like (S)ARIMA, neural networks, and other machine learning methods. These methods are explored in Section 1.2, which provides a review of studies predicting companies' cash flow and their financial situation. However, few studies use transactional data for forecasting time-series cash flow data, likely because companies do not want to share sensitive transactional data (Malkus & Nalepa , 2023). While some research uses machine learning for cash flow prediction, no studies specifically use XGBoost and Random Forest to predict company liquidity using transactional data. This study area remains unexplored.

In this study, a comparison of the two models is conducted. The selection of XGBoost and Random Forest algorithms for this study is motivated by their ensemble learning approach, which leverages multiple decision trees to make accurate predictions, and their adaptability to handle complex datasets (Chen & Guestrin, 2016) (Breiman, 2001). However, the methodologies employed by these two algorithms to construct and combine multiple models diverge significantly. The difference is that Random Forest builds each tree independently (Breiman, 2001), while XGBoost builds one tree at a time, learning from the mistakes of the previous tree (Chen & Guestrin, 2016). In short, one is utilizing boosting and the other is employing bagging, making it interesting to compare them on their performance. Additionally, the two models, XGBoost and Random Forest, were specifically selected for this study to test their effectiveness on tasks they are not traditionally used for, such as time series forecasting.

The performance and robustness of the two machine learning models XGBoost and Random Forest are compared in this study by conducting a series of experiments using a time series dataset from almost 500 small and medium-sized businesses (SMEs) from the Netherlands. Companies from diverse industries, such as home care and repair, landscaping, and finance, are featured in the dataset. This diverse and extensive dataset provides a practical basis for the study, thereby enriching the relevance and impact of the results on financial advisory services.

This study significantly contributes to the existing literature, offering new perspectives on algorithmic liquidity predictions utilizing transaction data from SMEs. The findings provide fresh insights that can enhance understanding and practice in the area of cash flow forecasting and liquidity management.

1.2 Related work

Research into financial forecasting is not a new phenomenon. The first research, dating back more than half a century ago, used financial ratios to predict failure. After this paper, many other papers followed to predict the financial situation of a company. However, specific research on cash flow predictions using machine learning and transactional data is limited. In this Section, the history and the current techniques of financial forecasting are discussed.

1.1.1 Where it all began

In 1966, research was conducted on financial ratios as predictors of failure. The study provided evidence that certain financial ratios can predict corporate bankruptcy by using a dichotomous classification test. According to the study, the most significant predictors were the cash flow to total debt ratio and the net income to total assets ratio (Beaver, 1966). In 1968, Altman conducted similar research to Beaver but used a discriminant analysis to rank companies based on a weighted combination of five ratios. His model had an accuracy of 95% in selecting future bankrupts in the year prior to bankruptcy. However, the predictive accuracy declined strongly when predicting more than a year prior to bankruptcy (Altman, 1968).

Over time, researchers tried to improve bankruptcy forecasting models. In 1980, a model to evaluate the probability of bankruptcy was developed by Ohlson based on logistic regression analysis (Ohlson, 1980). In a study that compared Ohlson's model and Altman's model (both original and re-estimated models), the researchers concluded that Ohlson's original model had the strongest overall performance (Begley et al., 1996). Lorek and Willinger (1996) provided evidence on the time-series properties and predictive ability of cash flow. These studies conducted, in some cases more than half a century ago, form the foundation of modern financial prediction models.

1.1.2 Modern predicting models

This research focuses on predicting cash flow using machine learning techniques. Machine learning refers to a collection of techniques that can autonomously identify trends within data sets, learn from these patterns, and use this learning to make predictions or decisions. The discerned patterns are then employed for forecasting future data or making decisions in situations characterized by uncertainty (Murphy, 2014).

One notable study by Weytjens et al., 2021, paralleled this research by comparing different machine learning techniques for cash flow prediction. They used a large dataset comprising over 700,000 invoices per year. They concluded that the cash flows in the dataset had a strong weekly pattern that enabled them to predict the cash flows well. Also, they introduced Interest Opportunity Cost (IOC), a measure proposed to minimize a firm's money lying idle in customer payments' bank accounts, thereby facilitating transfer to interest-earning savings accounts or reducing the firm's working capital. It also seeks to avoid overdraft situations, accounting for the opportunity cost difference between debit and credit interest rates. By using the IOC, MSE, and MAE, they concluded that neural networks, especially LSTM, outperformed traditional forecasting methods like Prophet and ARIMA. However, the results showed that the different models are also very useful to forecast cash flows.

Other studies, like Dairu & Shilong (2021) and Paliari et al., (2021), have utilized XGBoost for forecasting in different (time series) contexts, like sales forecasting and stock market prediction, and reported positive results. Research by Salas-Molina et al., 2016, and Catal et al., 2019, showcased the effectiveness of Random Forest in time series predictions and sales forecasting, respectively.

Building on the existing body of knowledge, this research aims to fill the gap in the literature by providing a focused examination of XGBoost and Random Forest for cash flow prediction using transactional data. Although machine learning techniques have been tested in various forecasting scenarios, their application to cash flow prediction, particularly with time series transactional data is less explored. Therefore, this research will contribute novel insights to the field of algorithmic liquidity predictions.

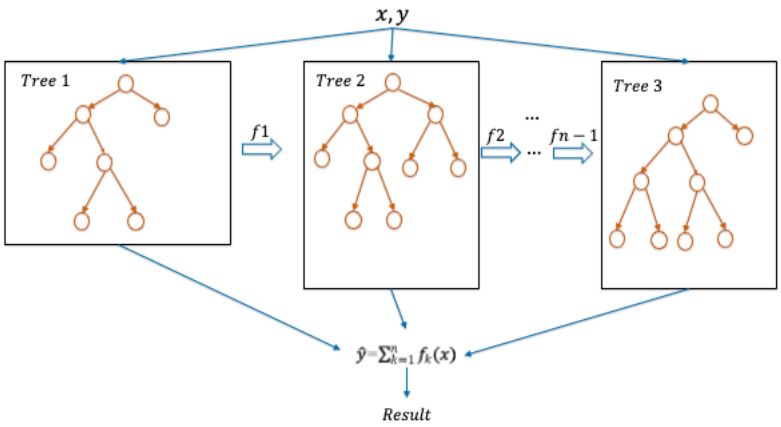
2. Preliminaries

In this Section, an understanding of the operations of XGBoost and Random Forest is provided. Both belong to a family of models referred to as ensemble methods, yet they each employ different techniques. XGBoost utilizes a boosting approach, while Random Forest adopts a bagging strategy. The selection of these two models, Random Forest and XGBoost, for this investigation was based on two main reasons. Firstly, both models are capable of handling complex non-linear relationships. Secondly, the models could measure the significance of distinct features, offering valuable insights into the dataset.

2.1 XGBoost

XGBoost stands out as a high-performing supervised learning algorithm, distinguished as a prime example of gradient boosting machines. Frequently utilized by data scientists, this boosting system provides remarkable results in various machine learning tasks. The essence of XGBoost lies in the concept of gradient boosting, where each new model is trained to correct the errors made by previous ones (Chen & Guestrin, 2016). Figure 1 demonstrates the XGBoost model's basic setup, showcasing its key components and flow of operations, which collectively provide a clear overview of its structure and functionality.

Figure 1 General Architecture XGBoost Model



The model works as follows:
If, for instance, we have for example a dataset DS with **m** features and an **n** number of examples:
D = {(xi, yi)} (|D| = n, xi ∈ R m, yi ∈ R)

Let ŷi be the predicted output of an ensemble tree model generated from the following equations:

$$\text{Å}\cdot_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}$$

To solve the equation above, we must find the optimal collection of functions, denoted by fk (where k is a specific tree in a model consisting of K trees), by minimizing both the loss and regularization objectives.

$$\mathcal{L}(\phi) = \sum_i l(y_i, \text{Å}\cdot_i) + \sum_k \Omega(f_k)$$

The difference between the anticipated output I and the actual output yi is represented by the loss function, which is denoted by the letter l. The loss function measures the difference between the expected output and the actual output, yi. This not only helps in preventing overfitting of the model but also provides a measure of the model's complexity.

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

In the equation above, T stands for the number of leaves on the tree, and w for each leaf's weight. Boosting is a technique used in decision trees to minimize the objective function, and it works by continuously adding new functions while the model is trained. Thus, in the t-th iteration, the following new function (tree) is added:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \left(y_i, Å \cdot_i^{(t-1)} + f_t(x_i) \right) + \Omega(f_t)$$

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda} \right] - \gamma$$

$$g_i = \partial_{Å \cdot^{(t-1)}} l(y_i, Å \cdot^{(t-1)})$$
$$h_i = \partial^2_{Å \cdot^{(t-1)}} l(y_i, Å \cdot^{(t-1)})$$

One advantage of XGBoost over traditional gradient boosting is its inclusion of a regularization parameter, which controls model complexity and helps prevent overfitting. This makes XGBoost more robust to noise and able to handle more complex patterns. The model can also handle missing values and can make use of parallel processing, which makes the processing time faster.

In the context of this study, XGBoost will take the features such as transaction date, classification code, division, sector code, company size code, business type code, state, year, and month, and use a gradient boosting framework to iteratively learn from the errors of prior models to make predictions about the monthly_total. It will treat these as inputs into a series of decision trees that are built in sequence, where each subsequent tree aims to correct the misclassifications made by the previous tree. In the upcoming Section 3, the specific features are clarified.

2.2 Random Forest

While XGBoost builds an ensemble of shallow and weak successive trees with each correcting the errors of the previous ones, random forest builds an ensemble of deep independent trees.

Introduced by Breiman in 2001, Random Forest is an ensemble machine learning algorithm that uses a collection of decision trees. The construction of these decision trees is governed by two key techniques: Bagging and the Random Subspace Method.

1. **Bagging**: Random Forest applies a technique known as bootstrap aggregating, or bagging, which reduces the variance in predictions. This technique involves generating different subsets of the original dataset (with replacement), each of which is used to train a separate decision tree. This creates a "forest" of different trees.

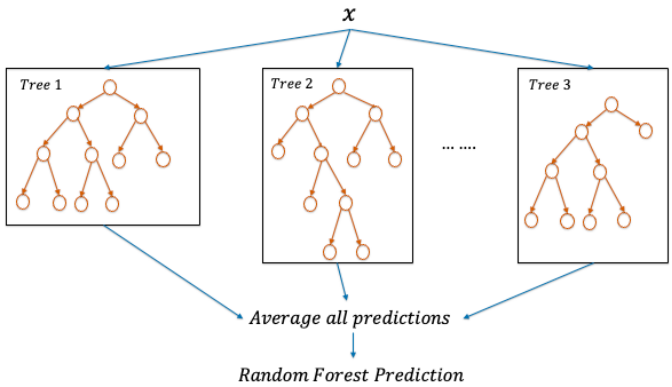
2. **Random Subspace Method**: To further enhance the model's robustness, Random Forest applies the Random Subspace Method during the tree construction phase. Instead of considering all features for each split in the decision tree, a random subset of features is selected. This additional layer of randomness helps create more diverse trees, reducing the correlation between them and thus the model's overall variance.

3. **Out-of-Bag (OOB) Error Estimate:** The bagging procedure in Random Forest introduces an interesting property called the Out-of-Bag (OOB) error estimate. This error is calculated by averaging the prediction errors of each training instance x_i, using only the trees that did not have x_i in their bootstrap sample. The OOB error provides a useful internal error estimate of the model, often eliminating the need for a separate validation dataset.

4. **Prediction**: When making a prediction, each tree in the Random Forest gives its own prediction. For regression tasks, the model's final output is the average prediction across all trees. For classification tasks, the class that receives the majority of votes across all trees is selected as the final output. In this paper, the average prediction across all trees will be calculated since it is a regression task.

In Figure 2, the overall architecture of the Random Forest model is presented. The diagram visualizes the key components and the workflow of the model, providing a comprehensive overview of its structure and function.

Figure 2 General Architecture Random Forest Model



In this study, the Random Forest model uses features like transaction date, classification code to create a multitude of decision trees. Each tree is trained on a random subset of the data, and at each node, a random subset of features is used for splitting. The final prediction of 'monthly_total' is the average across all trees, providing a balanced and robust prediction.

3. The Data

3.1 An overview of the data

In this study, the performance of two machine learning models, XGBoost and Random Forest, is rigorously evaluated using a large transactional dataset from a Dutch accounting firm. The choice of this dataset stems from the prevalent issue in many accounting firms, which despite technological advancements, still heavily rely on manual methods for cash flow forecasting. As these firms scale up, such manual processes can become not only time-consuming but also expensive. By leveraging the large amounts of historical transaction data via machine learning models, there lies potential to automate forecasting systems, significantly reducing time, cost, and enhancing operational efficiency.

This particular dataset is substantial, encompassing nearly 7 million rows, allowing for a robust and meaningful evaluation of the models. Using such a large dataset mimics real-world situations where dealing with big data is common, and thus provides a solid foundation for assessing the performance and robustness of these models. Moreover, both XGBoost and Random Forest excel at modeling complex, non-linear relationships, which are often inherent in large transactional datasets. This makes them appropriate selections for this evaluation.
This dataset, anonymized for privacy, represents the historical transactions of roughly 500 Dutch companies. The span of data reaches back a decade, beginning in 2010 and extending to the current day. The features required for the models have been carefully selected, and after retrieving the data using SQL queries, the following columns remain:

- **Transaction_date:** A standardized date format (YYYY-MM-DD) capturing the timing of each transaction from January 1, 2010, to March 1, 2023.

- **Classification_code:** An RCSFI Code, representing the categorization of transactions based on the General Ledger Schemes (RCSFI) framework.

- **Division:** A unique identifier assigned to each company within the dataset, allowing us to distinguish and analyze individual entities.

- **Sector_code:** An identifier specifying the sector to which each company belongs, providing valuable insights into industry-specific trends and patterns.

- **Company_size_code:** A code indicating the size of each company, allowing us to investigate potential variations based on company size.

- **Business_type_code:** A code representing the type of business conducted by each company, providing additional context for our analysis.

- **State:** The origin state of the company

- **Monthly_total:** The total costs or income recorded per month, serving as a key variable for forecasting and analysis.

The Reference Classification System of Financial Information (RSCFI), also known in Dutch as "*Referentie GrootboekSchema*" (RGS), is a standardized system designed for enhancing the automation and integration of administrative activities. Each general ledger account is associated with an RSCFI code, which is also tied to Standard Business Reporting (SBR). In this study, a subset of these codes is selected for cash flow prediction, including BEiv, Wafs, WBed, WBel, WFbe, WKpr, WOmz, WOvb, WOvt, WPer, and WRed. These codes embody key financial components of a company, including personnel expenses, total revenue, taxes, and depreciation, among others. A description of the RSCFI codes can be found in Appendix I.

3.2 Data preparation

The dataset for this study was obtained from an accounting firm's database. Utilizing SQL to query data, an initial feature selection was conducted to remove features not pertinent to this study. The original dataset contained 6,863,336 rows and 15 columns. However, during the data preprocessing stage, several key decisions were made to refine the data for the predictive models.

- **Columns Elimination:** Irrelevant columns were removed from the dataset to reduce the dimensionality and improve the computational efficiency of the models. Especially columns including descriptions of the data were removed.

- **Handling Missing Values:** Rows with 'division' as NA were eliminated. For other variables with NA values, new categories were created to maintain the integrity of the dataset without excluding valuable data. While XGBoost and Random Forest are capable of handling missing values, new categories were created for these NA values to maintain the integrity and completeness of the dataset.

- **RCSFI Codes:** RCSFI codes were shortened to four characters, reducing the complexity of the dataset. Non-relevant RCSFI codes were also removed for the same reason. These include codes that do not contribute to the determination of the cash flow.

- **Data Aggregation:** Data was grouped by month, RCSFI code, and company. The decision to group data by monthly simplifies the dataset, providing a fixed set of data points for each month rather than variable daily values. This approach not only makes the data more manageable, but also aligns well with the study's goal of forecasting the cash flow situation on a month-by-month basis.

- **Feature Scaling:** The 'monthly_total' column was standardized using the scale function. This process, which involves subtracting the mean and dividing by the standard deviation, makes it easier to compare the models on their performance.

- **Time Frame:** Rows with a 'transaction_date' earlier than 2016-01-01 were removed to focus the study on recent data, which is likely more relevant to future predictions. The number of divisions prior to 2016 was also relatively low.

- **Outlier Removal:** Outliers can influence a model's performance negatively. Therefore, data points below the 2.5th percentile minus 1.5 times the interquartile range (IQR) or above the 97.5th percentile plus 1.5 times the IQR were identified as outliers and removed. In this dataset, the outliers were extremely large transactions.

- **Feature Engineering:** Additional 'month' and 'year' columns were derived from the 'transaction_date' column. This process enhances the data's dimensional richness and allows the models to capture potential seasonal or annual trends in the transaction data.

Table 1 illustrates examples of three full transactions including the features that are used.

Transaction_date	Classification_ code	Division	Sector_code	Company_ size_code	Business_ type code	State	Monthly_total	Year	Month
2017-01-04	WOmz	999321	K	A	90	UT	0.55352	2017	January
2020-05-10	WKpr	567423	F	H	41	NH	-0.65352	2020	May
2021-05-10	BEiv	192756	M	D	41	UT	-1.45439	2021	May

Table 1 Full transactions example

4. Experiments

In this Section, the two models are evaluated by conducting several experiments. The purpose of the experiments is to assess the robustness and effectiveness of the models in different circumstances. The performance of the models is expressed in terms of RMSE and MAE. Seven different main experiments are conducted, each of which also has sub-experiments. After these experiments, the model is tested on practical operation to see if it can assist accountants. Firstly, the performance indicators are clarified. Secondly, the seven experiments are conducted and discussed. Lastly, the model is tested for its applicability to accountants.

4.1 Performance Indicators

The presented models are evaluated based on their errors. The comparison facilitates the identification of the best-performing model, which will subsequently serve as the foundation for developing the most accurate cash flow forecaster in this research. Specifically, two performance indicators, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), are compared. These metrics were chosen as they are widely used in similar predictive modeling studies for their interpretability and effectiveness at capturing model performance. This approach aligns with similar research, such as the study by Weytjens et al., (2021), which used Mean Squared Error (MSE) and MAE. However, this study used RMSE over MSE due to its enhanced interpretability, as it is expressed in the same units as the original data, thus providing a clearer understanding of the model's error.

Root Mean Squared Error (RMSE): A widely used indicator for assessing how well a model predicts the future. It determines the differences between actual and predicted values, also referred to as residuals. This metric evaluates the predictive errors of various models on a particular dataset, instead of making comparisons across multiple datasets.
The RMSE can be calculated by using the following formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

The Mean Absolute Error (MAE): By taking the average of absolute differences between the predicted and actual values, provides a clear and robust measure of the model's prediction error, less sensitive to the influence of potential outliers. It delivers an easily comprehensible metric, expressing the average error in the same units as the original data.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y^i - {Y^i}^2\right|$$

By leveraging both Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) in conjunction, this study ensures a comprehensive assessment of the model's performance. It takes into account both the average error magnitude, as measured by MAE, and the distribution of error magnitudes, as captured by RMSE. This dual-metric approach allows for a more nuanced evaluation and comparison of the predictive models under investigation in this study.

In Section 4.2.3, the RMSE as a percentage of the mean absolute is calculated. This percentage provides an understanding of the relative size of the RMSE in relation to the average value of the data. The formula that is utilized is:

$$RMSE\ Percentage = ((\frac{RMSE}{MA}) * 100)$$

4.2 The Experiments

4.2.1 The Experimental Setup

The experiments to assess the robustness and effectiveness of the two models in different circumstances are conducted utilizing the transactional dataset which is clarified in Section 3. At first, default models are created utilizing default hyperparameters provided by the used package. In Appendix III, the default parameters that are used for building the models are listed and in Appendix II, the results of the experiments are given. The experiments that follow are meant to evaluate and compare the robustness of the default models. In Section 4.2.2, the results of the experiments are discussed. The models to conduct the experiments were built using the *xgboost* and *randomForest* packages. The training set for the default model contains data from 2016-01-01 till 2022-01-01.

The XGBoost model, trained with default parameters, produced an RMSE of 0.5906 and an MAE of 0.3503 on the test set. This resulted in an RMSE accounting for 101.83% of the absolute mean of the test set. The training time for the XGBoost model was 23.24 seconds.

On the other hand, the Random Forest model, also trained with default parameters and 100 trees, resulted in an RMSE of 0.578 and an MAE of 0.338 on the test set. The RMSE represented 99.66% of the absolute mean of the test set, indicating a slightly better fit to the test data compared to the XGBoost model. However, the training time was considerably longer, taking 12.53 minutes.

Once the performance of the default models was determined, the study further employed these models in a series of seven distinct experiments. The default models' performance, specifically in terms of their Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), was used as a benchmark for comparison and analysis of the results obtained from these subsequent experiments.

The comparative visualization of the RMSE and MAE for XGBoost and Random Forest is presented in Figure 3. Despite the longer training duration required, the Random Forest model demonstrated marginally superior performance in terms of the RMSE, attributed to its density more closely mirroring the actual density.

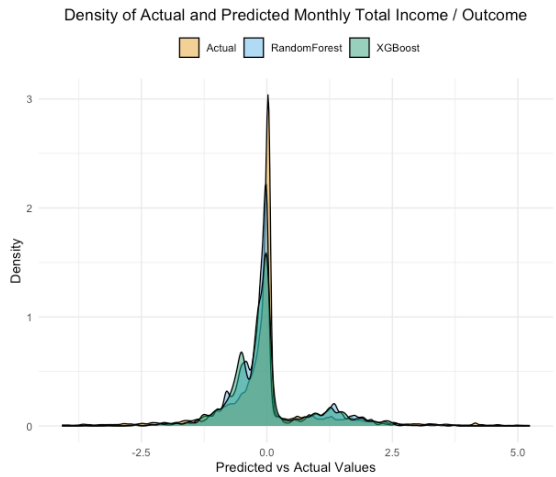


Figure 3 Density plot Actual vs. Predicted values

Feature Performance

In Appendix IV, the feature importance matrices for both models can be found. For the Random Forest model, the most determinant factor is the RCSFI Code of the transaction, followed by the division and sector. On the contrary, the XGBoost model places greater emphasis on the division as the most influential feature, succeeded by the RCSFI code 'WKpr'.

4.2.2 The Experiments

Experiment 1: *Examining Performance Improvement through Hyperparameter Optimization*

To optimize the hyperparameters, the Bayesian optimization technique is applied to both models. Bayesian optimization is a powerful technique for finding the extrema of functions that are computationally demanding (Brochu et al., 2010). This optimization technique carries the advantage of operating at a relatively swift pace. The optimal parameters to use according to the Bayesian optimization approach for both models can be found in Appendix III. The objective was to optimize the models in order to enhance their performance.

When comparing the optimized models with the default models, it is evident that the Bayesian optimization had a positive impact on both models' performances. There was a notable improvement in both the RMSE and MAE for the XGBoost and Random Forest models. Table 2 includes the results of the experiment. These results indicate that applying hyperparameter tuning techniques like Bayesian optimization can improve the performance of the models. This results in better financial forecasting and more informed decision-making for accountants, given the reduction in prediction error rates.

Table 2 Results Experiment 1

Experiment 1	**Optimized**	**XGBoost (MAE, RMSE)**	**Random Forest (MAE, RMSE)**
Default model	No	MAE: 0.3503 RMSE: 0.5906	MAE: 0.338 RMSE: 0.578
Experiment 1.1	Yes	MAE: **0.333** RMSE: **0.5636**	MAE: **0.306** RMSE: **0.5396**

Experiment 2: *Time Series Flexibility Assessment*

In experiment 2, the performance of the two models over different time frames is examined. After every experiment, the training period is extended, and the test data is shifted by one year. These experiments provide insights into how the models adapt and perform with the alteration in training and testing datasets over time. Also, this helps to in understanding how well the model would perform on unseen future data while respecting the chronological order of the data. Assumed was that the performance of the models declines when the amount of training data available is limited.

As can be seen in Figure 4, the RMSE and MAE improve slightly when adding more years to the training set. However, after adding two years, the performance stabilizes, indicating that further expansion of the training set does not significantly enhance the predictive performance of the models. Both models demonstrate relatively stable performance as the training data expands. However, the Random Forest model appears to be more stable than XGBoost, with less fluctuation in its RMSE and MAE values. Also, the relatively stable performance as more data is added might suggest that overfitting is not a major issue in these models. If overfitting were present, a decrease in performance would be expected as more data is added. In short, using more historical data can improve the models' performance, but it does not guarantee better results, especially for XGBoost.

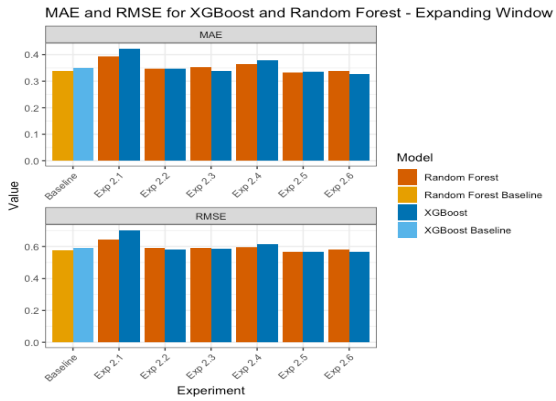


Figure 4 Bar chart of RMSE and MAE Experiment 2

Experiment 3: *Sliding Window Robustness Test*
In this experiment, the rolling window analysis is introduced and conducted. The goal of this experiment is to address potential temporal dynamics and evaluate the robustness of the predictive models under changing conditions over time. Different then experiment 2, the training set shifts ahead by one year for each iteration, maintaining a constant size while focusing on more recent data. It was initially expected that the performance would improve as the training and testing sets became more recent, considering that with each passing year, more divisions are added to the dataset, thereby increasing the number of data points.

From the data presented in Figure 5, it's apparent that the performance of both XGBoost and Random Forest models, as assessed by Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) metrics, is at its lowest in Experiment 3.1. In this experiment, the models are trained with data from 2016. The reason for this could be that there were fewer datapoints in 2016 since there were fewer divisions. As subsequent experiments were conducted, the performance of the models stabilized and showed improved results. In general, Random Forest outperforms XGBoost in terms of both RMSE and MAE metrics for most of the years tested. This suggests that the Random Forest model may be better suited to handling this specific dataset's temporal dynamics, even if the differences are small.

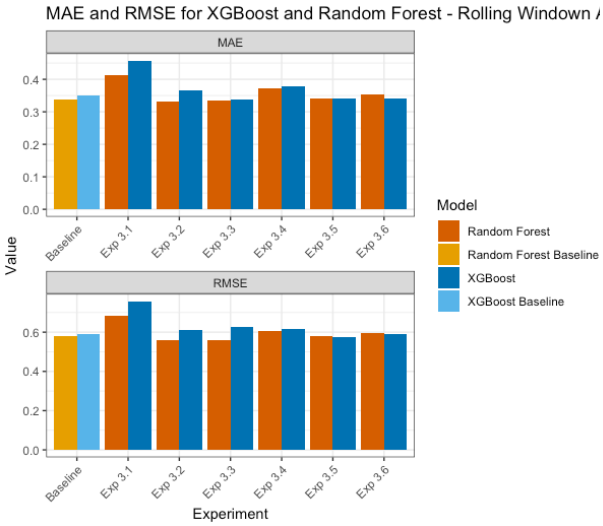


Figure 5 Bar chart of RMSE and MAE Experiment 3

Experiment 4: *Scalability Evaluation with Decreased Data Points*
To assess the adaptability and stability of the models under different conditions, this experiment systematically altered the number of divisions involved. Initial testing was conducted with half of the total divisions, delivering an initial evaluation of the model's performance. Following this, the experiment further reduced the divisions to 25% of the total, providing a stringent test of the models' ability to maintain accuracy and reliability in the face of substantial data reduction. This is insightful when considering the use of these models in scenarios where data is limited.

Table 3 includes the results of the experiments. The performance of Random Forest improves slightly after reducing the number of divisions in the dataset. The performance of XGBoost decreases by a reduction of 50% and increases when reducing the number of divisions by 75%. One reason for better performance when reducing the number of divisions for Random Forest might be due to the reduction in overfitting or noise, as having fewer divisions can sometimes help to improve the model's ability to generalize. The performance of the XGBoost model shows a more variable response to the change in divisions. When the number of divisions in the dataset is reduced by 50%, the performance decreases. This could potentially be due to underfitting, as the model might not have enough data to accurately learn the underlying patterns. However, when the number of divisions is reduced further by 75%, the performance increases. It suggests that at this level of data reduction, the model might be getting an optimal balance between bias and variance, leading to improved performance.

Table 3 Results Experiment 4

Experiment 4	Reduction in Divisions	XGBoost (MAE, RMSE)	Random Forest (MAE, RMSE)
Default model	0%	MAE: 0.3503 RMSE: 0.5906	MAE: 0.338 RMSE: 0.578
Experiment 4.1	50%	MAE: 0.348, RMSE: 0.61332	MAE: 0.3267, RMSE: 0.5576
Experiment 4.2	75%	MAE: **0.3126** RMSE: **0.533**	MAE: **0.3194** RMSE: **0.55370**

Experiment 5*: Performance Assessment After Integration of Auxiliary Datasets*
In this experiment, the performance of the models is evaluated before and after the inclusion of two external datasets. The datasets that are joined to the existing dataset include the consumer confidence of Dutch households and the Dutch Consumer Price Index (CPI). These external datasets might provide additional context or features that could potentially improve the performance of the models since the original dataset could have data limitations. The external dataset may fill in the missing information. The two external datasets are retrieved from the statistics Netherlands database Statline. The selected datasets were chosen strategically for their unique attributes. Both datasets depict a monthly representation of the economic situation in the Netherlands.
The additional datasets had not much influence on the performance of both models. However, adding the CPI dataset resulted in a slightly better performance for XGBoost. This could be because the CPI data may contain useful information that the model can leverage to make more accurate predictions. See Table 4 for the results of the experiment.

Table 4 Results Experiment 5

Experiment 5	Dataset	XGBoost (MAE, RMSE)	Random Forest (MAE, RMSE)
Default model	-	MAE: 0.3503 RMSE: 0.5906	MAE: **0.338** RMSE: **0.578**
Experiment 5.1	Consumer Confidence	MAE: 0.3583, RMSE: 0.5956	MAE: 0.3469 RMSE: 0.5888
Experiment 5.2	CPI	MAE: **0.3468** RMSE: **0.5790**	MAE: 0.35182, RMSE: 0.5935

Experiment 6: *Outlier Sensitivity Analysis*
To boost model performance and avoid overfitting, outliers from the initial dataset have been removed. Particularly, 2.5% of the data's bottom and upper extremes have been eliminated. To evaluate the impact of outliers on the performance of the two models, the outliers are reintroduced to the dataset in this experiment. Reintroducing the outliers back in the dataset resulted in a decrease in the RMSE and MAE for both models, thus improvement of the performance. The nature of the outliers may be the cause of this. The outliers might have aided the model in capturing more data variance, which would have improved predictions and hence lower error values. For Random Forest the increase in performance is smaller than for XGBoost. This suggests that XGBoost benefits more from the inclusion of outliers. The experiment demonstrated that reintroducing outliers back into the dataset had a positive effect on the performance of both models, resulting in slightly improved performance.

Table 5 Results Experiment 6

Experiment 6	**Outliers Removed**	**XGBoost (MAE, RMSE)**	**Random Forest (MAE, RMSE)**
Default model	Yes	MAE: 0.3503 RMSE: 0.5906	MAE: 0.338 RMSE: 0.578
Experiment 6.1	No	MAE: **0.1085** RMSE: **0.5096**	MAE: **0.1116** RMSE: **0.5616**

Experiment 7: *Assessing Random Forest and XGBoost Model Robustness Against Added Noise*
To expose the models to variations and uncertainties in the data, noise is added to the training and test dataset. Noise levels with standard deviations of 0.01, 0.1, 1, and 10 are added to the original data. It was expected that the performance of the models would decrease when adding more noise. This experiment was conducted to evaluate how sensitive XGBoost, and Random Forest models are to slight changes in the data. If the model's performance significantly decreases by adding a small amount of noise to the data, the models might not be robust to real-world data.

The RMSE and MAE did not change much after adding 0.01 and 0.1 standard deviations of noise to the data compared to the default model. However, when adding 1 or 10 standard deviations of noise to the data, the performance of the models decreased strongly. The introduction of noise into the dataset complicates the model's task of discerning the inherent patterns or trends, as it must contend with these random variations that do not reflect the true nature of the data. However, the models are robust to low levels of noise. Figure 6 demonstrates a noticeable decline in both models' performance. It highlights a limitation of the two models: their sensitivity to high levels of noise. While they can handle a small amount of randomness, large disruptions in the data, that do not correspond to real-world patterns, can significantly impair their prediction capabilities.

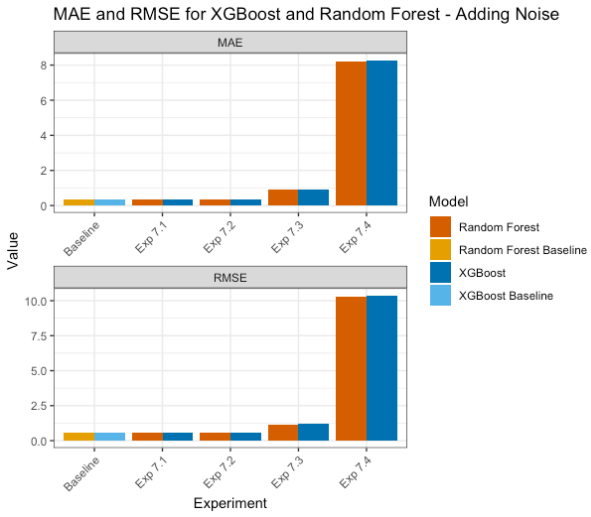


Figure 6 Bar chart of RMSE and MAE Experiment 7

4.3 Practical application of the two algorithms

In Section 4.2.1, a technical analysis of the XGBoost and Random Forest machine learning models has been performed. This section shifts focus to the practical applications of these models, particularly within the field of accounting. The central objective is to examine how well these models perform in real-world settings, with a spotlight on their capacity to predict a company's cash flows. Moreover, this analysis investigates the potential of these models to aid accountants and financial controllers in making more insightful, strategic decisions. An in-depth investigation of prediction errors across RCSFI codes and sectors is carried out in this Section, contributing to a comprehensive evaluation of the practical value of these machine learning models.

4.3.1 Default Model

Upon comparison, the Random Forest model outperformed XGBoost on unscaled data across all sectors and RGS codes, with a lower RMSE (3557.33 vs. 3774.98) and MAE (1891.20 vs. 2089.81). These findings suggest the superior predictive performance of the Random Forest model in the given context. In Figure 7, the relationship between actual and predicted values is presented. Comparing the two plots shows only relatively minor differences, suggesting that the models predict quite similar values. When translated into monetary terms, the Random Forest model's predictions deviate from the actual values by approximately €3557.33, whereas the XGBoost model deviates by about €3774.98, resulting in a difference of around €217.65.

The mean of absolute values of the test set is €3703.06. This means that the RMSE as a percentage of the mean absolute target value is 101.94%. In other words, the model's predictions are off by an amount that is more than the mean of the absolute target values. A lower percentage of the mean Absolut target value means a better-performing model.

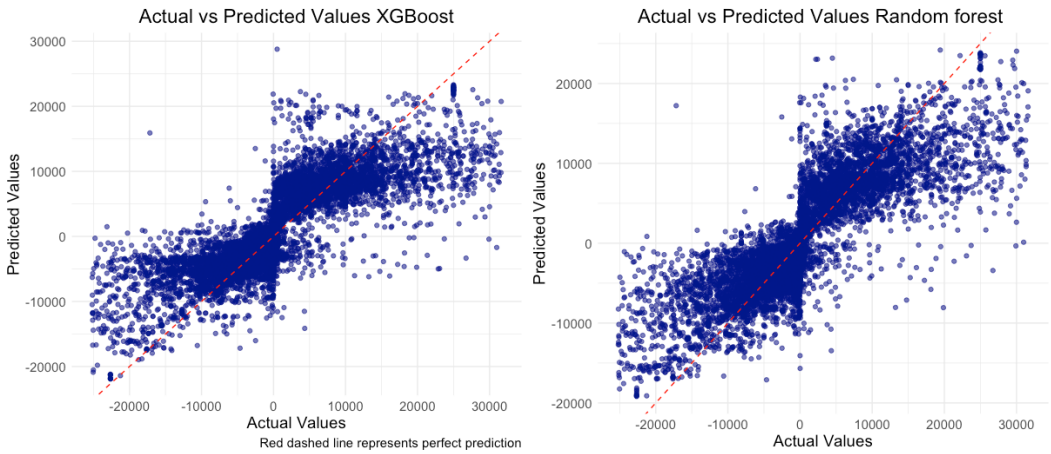


Figure 7 Actual vs Predicted values XGBoost and Random Forest

4.3.2 Performance by Sector

It is hypothesized that certain sectors are inherently more challenging to predict due to factors such as less regularity in their patterns. Figure 8 illustrates the absolute prediction error, denominated in euros, across various sectors, revealing substantial variability. Sectors B, G, N, and R - representing *mineral resource extraction, wholesale and retail trade, professional scientific and technical activities,* and *arts, entertainment, and recreation,* respectively - show the highest prediction errors. The highest prediction error is observed in sector B, with a value of €5624 when using the XGBoost model. This means that the average difference between the predicted values and the actual values is €5624. However, a high prediction error does not necessarily indicate a poor or ineffective model.

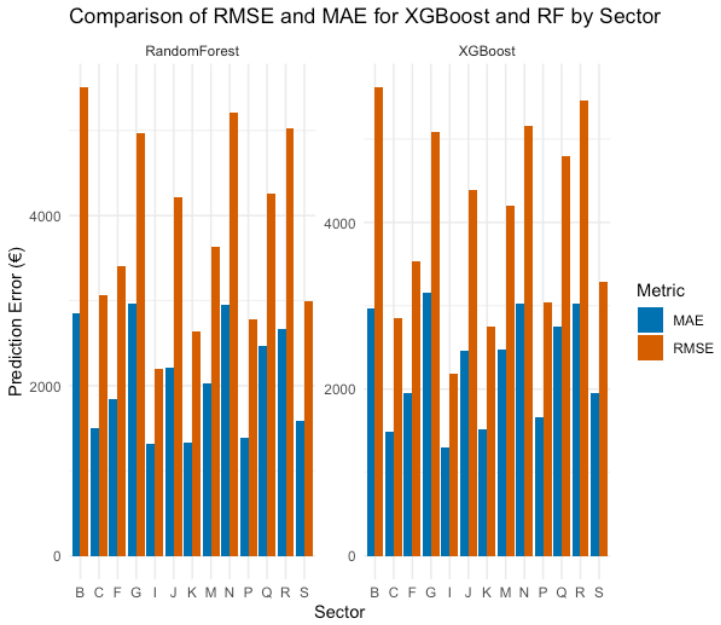


Figure 8 Absolute RMSE and MAE per sector

In order to assess the model's efficacy and its potential utility for accountants, the prediction error was calculated as a percentage of the absolute mean. This percentage-based error metric provides a more normalized basis for evaluating the model's performance. Figure 9 visualizes the RMSE as a percentage of the absolute mean, computed using the Random Forest model. Figure 8 indicated sectors B, G, N, and R as those with the highest prediction errors. While sectors R and B continue to demonstrate relatively high prediction errors, the model's performance for sectors G and N is notably better. The model is the most efficient when predicting values for sectors M and K. However, since the average RMSE as a percentage of the absolute mean is around 100%, the model's predictions deviate by an amount equivalent to the mean of the actual observed values.

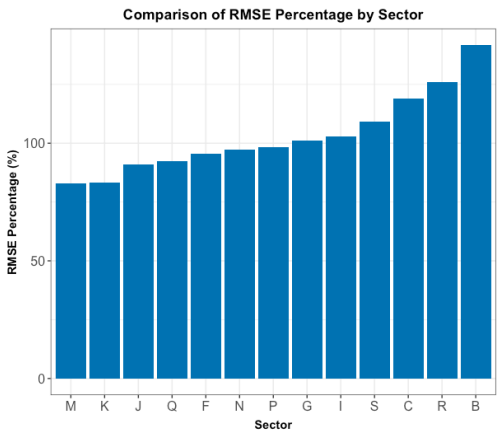


Figure 9 RMSE as percentage of Absolute mean by sector (RF)

4.3.3 Performance by RCSFI Code

The operational structure of a company encompasses both fixed and variable expenses. Fixed costs, such as depreciation, interest expense, insurance, and often salaries, are generally predictable since they remain constant regardless of production volume. Conversely, variable costs and revenues, including operating expenses and taxes, fluctuate with business activity levels, making them more challenging to predict accurately.

It is observed in this analysis that the smallest prediction errors arise when forecasting fixed costs such as depreciation (*WAfs*), financial income and expenses (*WFbe*), and share in the result of associated companies (*WRed*), see Figure 10. Conversely, predicting variable components like revenue (*WOmz*) results in the largest prediction errors, closely followed by taxes (*WBel*). On average, the predicted value for WOmz deviates from the actual value by approximately €6000, while more accurate predictions would be beneficial. These findings could suggest the inherent predictability of fixed costs and the volatility of variable costs in forecasting models.

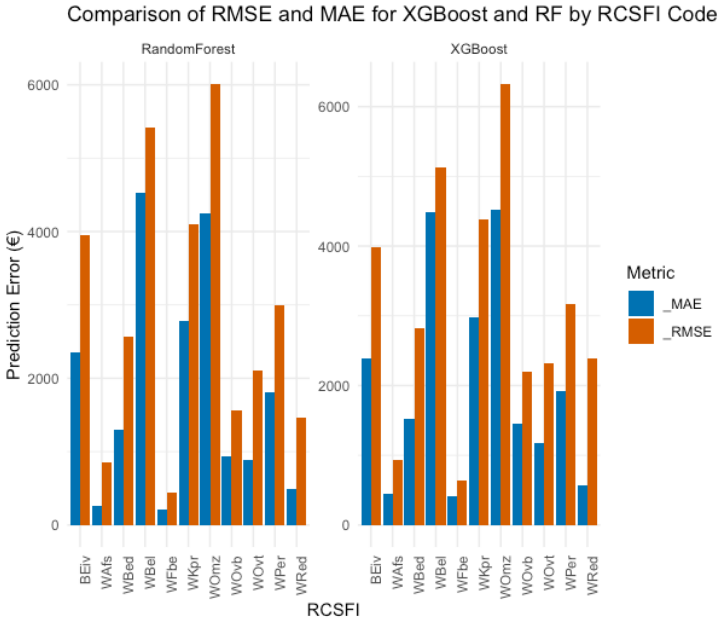


Figure 10 Absolute RMSE and MAE per RCSFI Code)

However, the absolute prediction errors do not provide a comprehensive understanding of the model's performance. To get a better sense of the model's predictive performance, the RMSE should be considered as a percentage of the absolute mean. As indicated in Figure 11, the model shows the greatest precision in predicting salaries (WPer) and revenue (WOmz). Conversely, the Random Forest model struggles significantly with predicting the share in the result of associated companies (WRed), as indicated by an exorbitant 2872.91% RMSE, making accurate predictions nearly impossible.

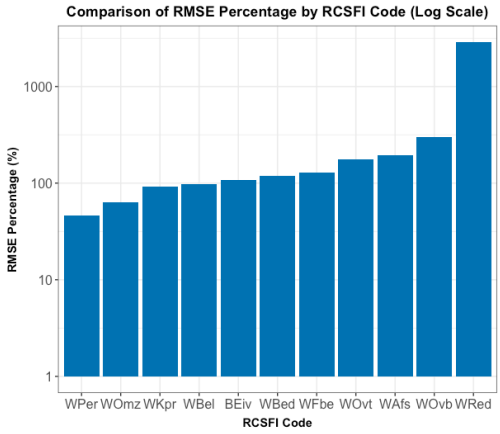


Figure 11 RMSE as percentage of Absolute mean by RCSFI Code (RF)

5. Conclusion & Discussion

5.1 Conclusion & Discussion

This research aimed to identify the differences in performance between XGBoost and Random Forest when predicting cash flow using time series transactional data from SMEs. The study also aimed to assess whether these predictive models could realistically be employed by accountants in their daily operations. In conclusion, this study determined that while the performances of the models are comparable, the default Random Forest model marginally outperformed XGBoost. However, in various experiments conducted, there were instances where XGBoost outperformed Random Forest. Therefore, the performance of one model over another depends on the training data used. One key advantage of XGBoost is the accelerated model training process. Random Forest is more than 12 minutes slower when training the model than XGBoost.

When it comes to the usability of the model, and in answering the question of whether accountants can use machine learning models like XGBoost and Random Forest to predict a company's liquidity with this specific dataset, the oversimplified answer is no. Although Random Forest and XGBoost show promising performance in certain circumstances, the RMSE of the best default model as a percentage of the mean absolute target value was 101.94% using this cash flow time series data. This means that, on average, the predictions of the default model have an error that is approximately 101.94% of the average magnitude of the target variable. This renders the predictions unreliable for usage by accountants since they benefit from more accurate predictions.

Accountants benefit from accurate predictions since they base their budget models for companies on these predictions. Since accountants manually predicted the cash flow position of companies, they wanted to optimize this process, which led to the request to create a prediction model based on transactional data. However, the two models cannot be employed because of their lack of precision. Interestingly, in other cases, such as in the study of Weytjens et al., (2021), machine learning predictions based on transactional data were useful. This study used a smaller dataset and roughly identical attributes to those used in the current analysis. This can be due to the nature of the data.

Two fundamental issues may be to blame for the prediction models' low performance. The first is related to the debits and credits of a company. Figure 8 illustrates the income and outcome per RCSFI code of the most represented division in the dataset. In the data, it is difficult, if not impossible, to identify a distinct pattern for the different RCSFI codes. This could be the result of the debits and credits. For instance, if 'WOmz', representing a company's net sales, is received and later needs to be reversed, a negative 'WOmz' transaction results. While expected is that net sales are always positive, this is not the case in the transactional dataset. If a 'WOmz' transaction has been received and for some reason needs to be reversed, a negative 'WOmz' transaction will result.

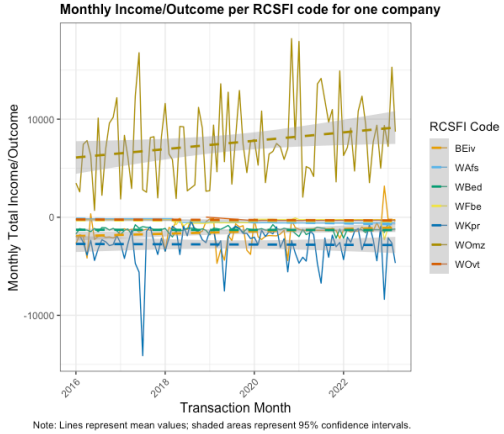


Figure 12 Income and Outcome of one company

Secondly, the dataset's limitations may contribute to the issue at hand. The main factors influencing the predictive models are the division, the RCSFI code, and the date. The use of other existing features in the predictions is minimal. Considering the absence of a significant seasonal pattern in the data, the introduction of other features might be necessary to enhance the predictive performance. Although additional features are not typically employed in the relevant literature, their datasets generally exhibit a greater degree of seasonality. This difference could potentially stem from the varied methods used in bookkeeping. In the fifth experiment, despite incorporating two general economic datasets into the existing one, there was no observed improvement in performance.

5.2 Further Research

Future research could explore the creation of individualized cash flow prediction models for each company, using methods like (S)ARIMA or Prophet. Instead of a singular model that fits all companies, this approach allows each company to base their predictions on a model unique to its situation. However, this method requires identifiable trends in the data to ensure its success. Hence, further investigation is needed to confirm the feasibility and effectiveness of this more personalized approach to cash flow prediction. To delve further into this issue, one possible approach is to explore hybrid models that blend machine learning models with traditional statistical methods. For example, this could involve coupling an ARIMA model, recognized for its effectiveness in time-series analysis, with machine learning models like Random Forest or XGBoost. Support for this proposition comes from the compelling research conducted by Phan & Nguyen (2020). They successfully integrated ARIMA with machine learning models, finding this hybrid approach superior and more reliable than both other hybrid models and traditional single-component methods. This affirms the potential value and effectiveness of hybrid modeling cash flow prediction, reinforcing its viability as an avenue for future exploration."

Although machine learning offers innovative approaches to cash flow prediction, it is crucial to understand and address the specific challenges posed by time series data to improve predictive performance.

5 References

Altman, E. I. (1968). Financial Ratios, Discriminant Analysis And The Prediction Of Corporate Bankruptcy. *The Journal of Finance*, 589-609, DOI: 10.2307/2978933.

Beaver, H. W. (1966). Financial Ratios As Predictors of Failure. *Journal of Accounting Research*, 71-111, DOI: 10.2307/2490171.

Begley, J., Ming, J., & Watts, S. (1996). Bankruptcy Classification Errors in the 1980s: An Empirical Analysis of Altman's and Ohlson's Models. *Review of Accounting Studies*, 267-284, DOI: 10.1007/BF00570833.

Breiman, L. (2001). Random Forests. *Machine Learning*, pages5–32, DOI: 10.1023/A:1010933404324.

Brigham, E. F., Ehrhardt, M. C., & Fox, R. (2016). Financial Management: Theory and Practice. Cengage Learning.

Brochu, E., Cora, V. M., & De Freitas, N. (2010). *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning.* CoRR: arXiv.

Catal, C., Ece, K., Arslan, B., & Akbulut, A. (2019). Benchmarking of Regression and Time Series Analysis Techniques for Sales Forecasting. *Balkan Journal of Electrical and Computer Engineering* , 20-26, DOI: 10.17694/bajece.494920.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794, DOI: 10.1145/2939672.2939785.

Dairu, X., & Shilong, Z. (2021). Machine Learning Model for Sales Forecasting by Using XGBoost. *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)* (p. DOI: 10.1109/ICCECE51280.2021.9342304). Guangzhou: IEEE.

Fight, A. (2005). Cash Flow Forecasting. Butterworth-Heinemann Ltd.

Lorek, K. S., & Willinger, G. L. (1996). A Multivariate Time-Series Prediction Model For Cash-Flow Data. *The Accounting Review*, 81-101.

Malkus, B., & Nalepa , G. J. (2023). Forecasting Daily Cash Flows in a Company - Shortcoming in the Research Field and Solution Exploration. *Lecture Notes in Computer Science, vol 13810* (pp. DOI: 10.1007/978-3-031-25599-1_27). Cham.: Springer.

Murphy, K. P. (2014). Machine Learning: A Probabilistic Perspective. The MIT Press; Illustrated edition .

Needles, B. E., Powers, M., & Anderson, H. R. (1999). Principles of accounting. 7th Edition. Boston, MA.: Houghton Mifflin.

Ohlson, J. A. (1980). Financial Ratios and the Probabilistic Prediction of Bankruptcy. *Journal of Accounting Research*, 109-131, DOI: 10.2307/2490395.

Paliari, I., Karanikola, A., & Kotsiantis, S. (2021). A comparison of the optimized LSTM, XGBOOST and ARIMA in Time Series forecasting. *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)* (p. DOI: 10.1109/IISA52424.2021.9555520). Chania Crete, Greece: IEEE.

Phan, T.-T.-H., & Nguyen, X. H. (2020, Ag). Combining statistical machine learning models with ARIMA for water level forecasting: The case of the Red river Author links open overlay panel. *Advances in Water Resources, Volume 142*, DOI: 10.1016/j.advwatres.2020.103656.

Salas-Molina, F., Martin, F. J., Rodríguez-Aguilar, J. A., Serrà, J., & Arcos, J. L. (2016). Empowering cash managers to achieve cost savings by improving predictive accuracy. *International Journal of Forecasting*, Pages 403-415, DOI: doi.org/10.1016/j.ijforecast.2016.11.002.

Weytjens, H., Lohmann, E., & Kleinsteuber, M. (2021). Cash flow prediction: MLP and LSTM compared to ARIMA and Prophet. *Electronic Commerce Research*, 371–391, DOI: 10.1007/s10660-019-09362-7.

Appendix I – Sector and RCSFI Codes

Below, the different sector codes included in the dataset are explained. Each code corresponds to a specific industry. Sector T is added for all rows where sector was NA. Sectors E, L and T were not included in Section 4.3 due to the insufficient number of data points.

Sector Codes:

B	Winning of mineral resources
E	Distribution of water; waste and wastewater management and sanitation
F	Construction industry
G	Wholesale and retail trade; repair of motor vehicles and motorcycles
I	Provision of accommodation and meals.
J	Information and communication
K	Financial activities and insurances
L	Real estate operations and trading
M	Professional, scientific, and technical activities
N	Administrative and support services
P	Education
Q	Human health and social work activities
R	Art, entertainment, and recreation
S	Other services
T	Unknown

There are 11 RCSFI codes used which can be used to calculate the cash flow / working capital. The different RCSFI codes are:

RCSFI Codes:

BEiv	Group equity, Equity capital, Capital
WAfs	Depreciation
WBed	Other operating expenses
WBel	Taxes
WFbe	Financial income and expenses
WKpr	Cost of sales
WOmz	Revenue
WOvb	Other operating income
WOvt	Revenue from claims which are fixed assets and from securities.
WPer	Charges related to employee benefits.
WRed	Share in the result of associated companies.

Appendix II – Experiments

Detailed explanations and results of the seven different experiments are provided below. For each experiment, both the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) have been calculated. In total, seven different main experiments are conducted.

Examining Performance Improvement through Hyperparameter Optimization
Experiment 1: Hyperparameters are fine-tuned using the Bayesian optimization approach. The results are as follows:

- XGBoost: RMSE = 0.5636, MAE = 0.333
- Random Forest: RMSE = 0.5396, MAE = 0.306

Time Series Flexibility Assessment
Experiment 2.1 (Training: Jan 2016-Dec 2016, Testing: Jan 2017-Dec 2017)

- XGBoost: RMSE = 0.70273, MAE = 0.42274
- Random Forest: RMSE = 0.6170, MAE = 0.3776

Experiment 2.2 (Training: Jan 2016-Dec 2017, Testing: Jan 2018-Dec 2018)

- XGBoost: RMSE = 0.64418, MAE = 0.3937
- Random Forest: RMSE = 0.59606, MAE = 0.3640

Experiment 2.3 (Training: Jan 2016-Dec 2018, Testing: Jan 2019-Dec 2019)

- XGBoost: RMSE = 0.58348, MAE = 0.34704
- Random Forest: RMSE = 0.5669, MAE = 0.33564

Experiment 2.4 (Training: Jan 2016-Dec 2019, Testing: Jan 2020-Dec 2020)

- XGBoost: RMSE = 0.5904, MAE = 0.3469
- Random Forest: RMSE = 0.56908, MAE = 0.3313

Experiment 2.5 (Training: Jan 2016-Dec 2020, Testing: Jan 2021-Dec 2021)

- XGBoost: RMSE = 0.5844, MAE = 0.3372
- Random Forest: RMSE = 0.5685, MAE = 0.32613

Experiment 2.6 (Training: Jan 2016-Dec 2021, Testing: Jan 2022-Dec 2022)

- XGBoost: RMSE = 0.5910, MAE = 0.3515
- Random Forest: RMSE = 0.579620, MAE = 0.3398

Sliding Window Robustness Test
Experiment 3.1 (Training: Jan 2016-Dec 2016, Testing: Jan 2017-Dec 2017)

- XGBoost: RMSE = 0.7575, MAE = 0.4572
- Random Forest: RMSE = 0.61696, MAE = 0.3776

Experiment 3.2 (Training: Jan 2017-Dec 2017, Testing: Jan 2018-Dec 2018)

- XGBoost: RMSE = 0.6859, MAE = 0.413
- Random Forest: RMSE = 0.605137, MAE = 0.3711

Experiment 3.3 (Training: Jan 2018-Dec 2018, Testing: Jan 2019-Dec 2019)

- XGBoost: RMSE = 0.6126, MAE = 0.3665
- Random Forest: RMSE = 0.57272, MAE = 0.3401

Experiment 3.4 (Training: Jan 2019-Dec 2019, Testing: Jan 2020-Dec 2020)

- XGBoost: RMSE = 0.5570, MAE = 0.3314
- Random Forest: RMSE = 0.5785, MAE = 0.33949

Experiment 3.5 (Training: Jan 2020-Dec 2020, Testing: Jan 2021-Dec 2021)
- XGBoost: RMSE = 0.6259, MAE = 0.3373
- Random Forest: RMSE = 0.59149, MAE = 0.341740

Experiment 3.6 (Training: Jan 2021-Dec 2021, Testing: Jan 2022-Dec 2022)
- XGBoost: RMSE = 0.5581, MAE = 0.335
- Random Forest: RMSE = 0.5974, MAE = 0.35486

Scalability Evaluation
Experiment 4.1: *Decrease the number of divisions with 50%*
- XGBoost: MAE: 0.348, RMSE: 0.61332
- Random Forest: MAE: 0.3267, RMSE: 0.5576

Experiment 4.2: *Decrease the number of divisions with 75%*
- XGBoost: MAE: 0.3126, RMSE: RMSE: 0.533
- Random Forest: MAE: 0.3194, RMSE: 0.55370

Auxiliary Dataset Integration
Experiment 5.1: *Adding the consumer confidence dataset.*
- XGBoost: MAE: 0.3583, RMSE: 0.5956
- Random Forest: MAE: 0.3469, RMSE: 0.5888

Experiment 5.2: *Adding the CPI dataset.*
- XGBoost: MAE: 0.3468, RMSE: RMSE: 0.5790
- Random Forest: MAE: 0.35182, RMSE: 0.5935

Outlier Sensitivity Analysis
Experiment 6.1: *Adding the existing outliers to the data.*
- XGBoost: MAE: 0.1085, RMSE: 0.50958
- Random Forest: MAE 0.1116, RMSE: 0.5616

Assessing Random Forest and XGBoost Model Robustness Against Added Noise
Experiment 7.1: *Adding Gaussian noise to test and training data (standard deviation = 0.01)*
- XGBoost: MAE: 0.3479, RMSE: 0.5929
- Random Forest: MAE: 0.3402, 0.57941

Experiment 7.2: *Adding Gaussian noise to test and training data (standard deviation = 0.1)*
- XGBoost: MAE: 0.370, RMSE: 0.6014
- Random Forest: MAE: 0.3577, RMSE: 0.5864

Experiment 7.3: *Adding Gaussian noise to test and training data (standard deviation = 1)*
- XGBoost: MAE 0.9318, RMSE: 1.184
- Random Forest: MAE: 0.907, RMSE: 1.1555

Experiment 7.4: *Adding Gaussian noise to test and training data (standard deviation = 10)*
- XGBoost: MAE: 8.272, RMSE: 10.3760
- Random Forest: 8.1995, 10.2810

Appendix III – Parameters

Default Parameters

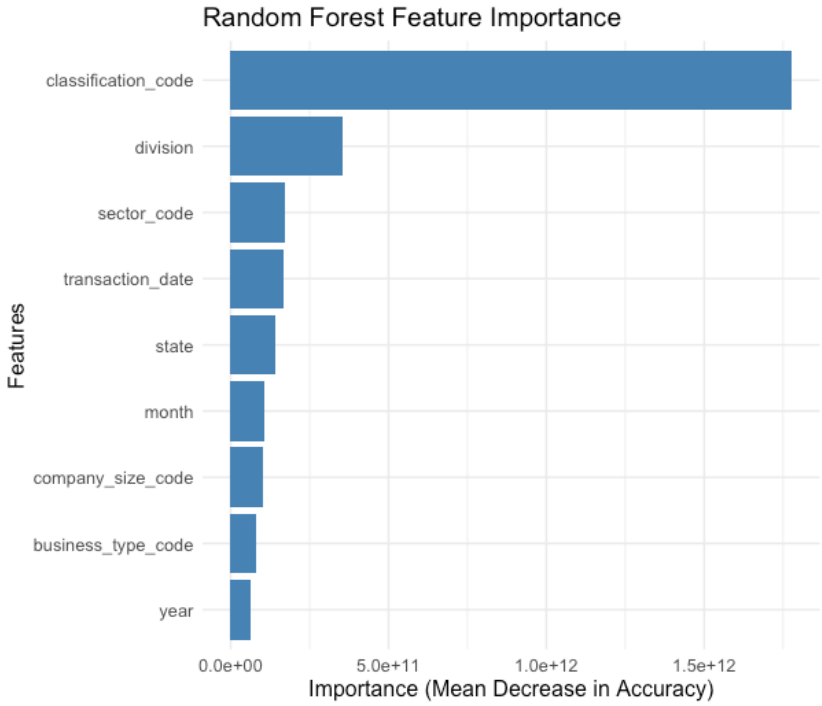
The default parameters are used for the seven experiments. On the left side, the parameters after optimization are given. The differences are small between the default version and the optimized version.

Default Parameters	After Bayesian Optimization
Random Forest: • ntree=100 • mtry=sqrt(number of features) • replace=TRUE • classwt=NULL • cutoff=1/(number of classes) • strata=NULL • sampsize=NULL • nodesize=5 • maxnodes=NULL • importance=FALSE • localImp=FALSE • nPerm=1	**Random Forest:** • ntree=100 • mtry=3.996979 • replace=TRUE • classwt=NULL • cutoff=1/(number of classes) • strata=NULL • sampsize=NULL • nodesize=5 • maxnodes=NULL • importance=FALSE • localImp=FALSE • nPerm=1
XGBoost: • missing=NA • nrounds=100 • objective="reg:squarederror" • booster="gbtree" • gamma=0 • max_depth=6 • min_child_weight=1 • subsample=1 • colsample_bytree=1 • colsample_bylevel=1 • eta=0.3 • tree_method="auto" • verbosity=1 • nthread • feval • maximize • early_stopping_rounds • eval_metric • seed • base_score=0.5 • monotone_constraints • interaction_constraints • n_jobs=1 • scale_pos_weight=1 • validate_parameters • predictor="auto"	**XGBoost:** • missing=NA • nrounds=23 • objective="reg:squarederror" • booster="gbtree" • gamma=0 • max_depth=10 • min_child_weight=0.9948652 • subsample=0.9948652 • colsample_bytree=1 • colsample_bylevel=1 • eta=0.2239748 • tree_method="auto" • verbosity=1 • nthread • feval • maximize • early_stopping_rounds • eval_metric • seed • base_score=0.5 • monotone_constraints • interaction_constraints • n_jobs=1 • scale_pos_weight=1 • validate_parameters • predictor="auto"

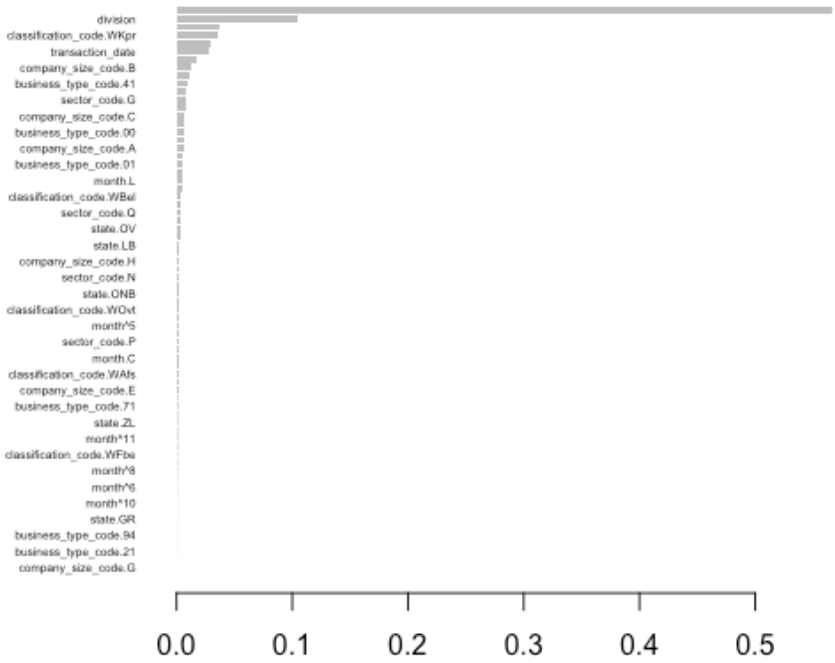
Appendix IV – Feature Importance

The feature importance chart of both models is given. The XGBoost chart provides a more detailed analysis, while the Random Forest chart offers a broader perspective. For Random Forest, the most important feature is classification code or RCSFI code. For XGBoost, this is Division.

Feature Importance – Random Forest



Feature Importance – XGBoost



Appendix V – Performance per Sector and RCSFI Code

The two tables below provide the RMSE and MAE per sector and per RCSFI code. Also, the absolute mean and the percentage of Random Forest's RMSE relative to the absolute mean is given, providing a measure of the error that is relative to the scale of the data. This is used in Section 4.3.

Per Sector code

Sector	***XGBoost (RMSE, MAE)***	***Random Forest (RMSE, MAE)***	**Absolute Mean**	***% (RF)***
B	5624, 2966	5511, 2846	3889.788	141.68
C	2852, 1489	3067, 1500	2574.315	119.13
E	1578, 1050	936, 650	2146.579	43.60
F	3535, 1957	3405, 1841	3560.564	95.64
G	5079, 3156	4967, 2968	4909.238	101.18
I	2189, 1296	2201, 1321	2137.994	102.95
J	4393, 2458	4218, 2217	4635.416	91.00
K	2745, 1524	2634, 1328	3168.016	83.17
L	1164, 918	528, 410	4059.563	13.00
M	4200, 2469	3636, 2034	4387.077	82.88
N	5164, 3026	5209, 2953	5351.826	97.33
P	3041, 1658	2781, 1389	2831.005	98.23
Q	4801, 2744	4262, 2472	4615.339	92.34
R	5461, 3030	5025, 2669	3988.825	125.98
S	3290, 1950	2990, 1588	2740.866	109.09
T	3153, 1702	3191, 1657	3162.996	100.88

The RMSE and MAE per RCSFI Code. Also, the absolute mean of the RCSFI code is given to calculate the RMSE as a percentage of the absolute mean.

RCSFI	**XGBoost *(RMSE, MAE)***	**Random Forest *(RMSE, MAE)***	**Absolute Mean**	**% (RF)**
BEiv	3984, 2383	3943, 2354	3665.32	107.57%
WAfs	929, 441	859, 263	446.17	192.57%
WBed	2829, 1521	2571, 1296	2176.32	118.13%
WBel	5130, 4484	5414, 4531	5597.42	96.73%
WFbe	644, 406	442, 207	344.48	128.34%
WKpr	4388, 2984	4091, 2783	4407.79	92.81%
WOmz	6329, 4528	6015, 4242	9531.65	63.10%
WOvb	2204, 1460	1559, 931	520.00	300.17%
WOvt	2311, 1178	2109, 887	1194.02	176.65%
WPer	3164, 1921	2998, 1801	6449.05	46.48%
WRed	2394, 575	1470, 488	51.16	2872.91%