University of Twente

Master Thesis

Faculty of Behavioural, Management and Social Sciences

A machine learning approach to one-step overnight interest rate forecasting

In collaboration with
Ernst & Young GmbH Wirtschaftsprüfungsgesellschaft

First Supervisor:
Dr. ir. WJA Van Heeswijk

Student:
Christina Caspari

Second Supervisor:
Dr. B Roorda

External Supervisor:
Dennis Will

December 3, 2021

UNIVERSITY OF TWENTE.

Abstract

Alternative Reference Rates (ARRs) are benchmark interest rates for overnight lending and include the Euro Short-Term Rate (€STR), Secured Overnight Financing Rate (SOFR) and Secured Overnight Index Average (SONIA). Ernst & Young has a variety of clients with financial contracts that are referenced with these rates. Regularly, clients request to receive a cash-in and cash-out flow estimate already in advance of the ending of the interest period. The required overnight rate is then not available yet as it is only published each business day. We calibrated different types of ARIMA and Random Forest models to test which model yields the highest forecasting accuracy of €STR, SOFR and SONIA. We developed one-step prediction models to lay a foundation for research in overnight interest rate forecasting. Furthermore, we included market and mathematical input features to increase the complexity of the models. We found that including input features to ARIMA decreases mean absolute error (mae) for €STR prediction from 19.9 bps to 5.48 bps, for SOFR from 68.8 bps to 34.6 bps and for SONIA from 5.99 bps to 4.91 bps. Including input features into Random Forest decreases the mae for €STR prediction from 240 bps to 16.9 bps and for SOFR from 67 bps to 9.25 bps. The performance of SONIA is less accurate using a Random Forest with input features as compared to without input features with an increase in mae from 6.22 bps to 35.5 bps. Furthermore, we found that the prediction accuracy of a Random Forest is heavily influenced by the choice of parameter settings. We decreased the mae by 12 bps for €STR, 4.1 bps for SOFR and 1.8 bps for SONIA by changing the parameter settings of the Random Forest. Finally, we have compared the performance of ARIMA and Random Forest to a simple prediction model with the assumption that the value of the interest rates of any given day is equal to the value of the interest rate of the previous day. All models that include input features produce error measures lower than the simple model.

Contents

1 Introduction

Internationally, banks are required to keep a certain percentage of their deposit as a reserve, which they are not allowed to use for lending activities. This reserve consists of cash and has to be held in physical form at the bank or on its account at the respective central bank. In addition, a bank has to deal with high fluctuations in liquidity and unexpected cash flows every day. Overnight lending helps banks keep their reserve requirements and payment obligations. For example, a bank that has a surplus over their reserve requirements can lend money to a bank that has a shortfall on their reserve requirements. These borrowing activities happen over night and are referenced with overnight interest rates. Overnight lending means that the borrower repays the respective amount including interest at the beginning of the following business day. A low overnight interest rate means that banks can borrow money from each other at a low cost. This makes it possible for the banks to charge a lower interest rate to their customers which in turn encourages customers to take on loans. Consequently, a growth in economy follows since customers - which include companies and private borrowers - are then in a better position to spend their money. Raising the interest rate has the adverse effect. A high interest rate leads to less interbank borrowing and consequently, fewer customers who take out a loan. Less money circulates through the market when overnight rates are high as compared to when overnight rates are low [1].

In this thesis, we will apply a Random Forest machine learning algorithm to forecast the overnight interest rates Sterling Overnight Index Average (SONIA), the Euro Short-Term Rate (€STR), and the Secured Overnight Financing Rate (SOFR) which are widely used current benchmark risk-free rates for a variety of financial instruments.

1.1 Alternative Reference Rates (ARR) on financial instruments

Overnight interest rates have been widely used as reference interest rates in cash instruments and derivatives. For example, overnight rates are used to reference Overnight Index Swaps (OIS), forward contracts, or floating rate notes.

The payment notice structure, which determines how long in advance an investor knows the interest payment he or she has to take, can be generally divided into *in advance* and *in arrears*. In an *in advance* structure, the payment that has to be made is set before the start of the interest period. The advantage is that investors know what payments they have to make which enables them to better plan their cash flows. However, investors typically tend to avoid the use of outdated interest rates. In an *in arrears* structure, the interest payment is only known at the end of the interest period. The advantage is that the interest rate is up-to-date, whereas the disadvantage is that investors only know what payment they have do make shortly before, or even only on the same day, as the payment has to be made [2].

Alternative Reference Rates (ARRs) are examples of overnight interest rates at which banks lend to each other in the overnight market. ARRs exist for a wide range of currencies internationally. The Sterling Overnight Index Average (SONIA), the Euro Short-Term Rate (€STR), and the Secured Overnight Financing Rate (SOFR) are the ARRs for the currency zones GBP, EUR and USD, respectively. Table 1 shows the ARRs for a variety of different currency zones. Previous to the ARRs, the Interbank Offered Rates (IBORs) were the benchmark interest rates for interbank lending.

Currency	Alternative Reference Rate (ARR)
GBP	Sterling Overnight Index Average (SONIA)
EUR	Euro Short-Term Rate (€STR)
USD	Secured Overnight Financing Rate (SOFR)
AUD	Reserve Bank of Australia Interbank Overnight Cash Rate (AONIA)
CAD	Canadian Overnight Repo Rate Average (CORRA)
CHF	Swiss Average Overnight Rate (SARON)
HKD	Hong Kong Dollar Overnight Index Average (HONIA)
JPY	Tokyo Overnight Average Rate (TONA)

Table 1: ARRs for eight major currencies [3].

The transition from IBOR to ARR is still in progress as the Financial Stability Board (FSB) dictates that contracts shall be referenced with ARRs instead of IBORs by the end of 2021.

1.2 Relevance for EY

This thesis is written in collaboration with Ernst & Young GmbH Wirtschaftsprüfungsgesellschaft. The company has a vast range of internationally active clients who use financial instruments to hedge against various risks or simply use them for financing purposes. To hedge against interest rate risk during the term of these instruments, banks often use interbank interest rates as a variable component of the agreed cash flows. As these cash flows have an impact on the company's profit, the future development of these interest rates is also of interest to the companies and, as their advisor, to EY. Forecasting overnight rates can yield valuable information for investors in terms of cash flow management and interest rate risk hedging.

Knowledge about future development of interest rates gives investors information about their future cash flows. We will develop a base case model that can make one step predictions for the €STR, SOFR and SONIA. We chose these rates because the €STR is the common overnight rate in the Netherlands and in Germany and the SOFR has one of the highest total transactions volumes. We further selected the SONIA because its predecessor, the London Interbank Offered Rate (LIBOR), was in fact a global - instead of a local - benchmark.

The goal of this research is to develop a reliable one-step prediction model for the overnight rates. Such a model can be used by EY to calculate the month-end accounting entries for various financial instruments. The interest amount of a financial product which is referenced with an ARR and uses an *in arrears* payment structure is usually known only on the same day that the booking has to be made or one day after the booking has to be made. This is because the market data is only available on the same day or even only on the next business day. An accurate one-step prediction model will allow clients to make their bookings in advance of the month-end closing.

1.3 Problem statement

There are different models that can be used to forecast overnight interest rates. For example, the Autoregressive Integrated Moving Average (ARIMA) model is widely used to forecast different kinds of time series data [4], [5]. Research on the prediction of time series further suggests that machine learning algorithms are promising in terms of their forecasting ability [6]. The problem is that the forecasting accuracy of classical time series models such as ARIMA as well as machine learning algorithms such as Random Forest heavily depend on the data set [7]. Therefore, we do not know which model can predict the overnight interest rates €STR, SOFR and SONIA most accurate.

The goal for this thesis is to develop a reliable one-step prediction model for €STR, SOFR and SONIA. Therefore, we will compare the forecasting performance of both ARIMA and Random Forest to make recommendations for the use of these models in practice.

1.4 Research objective and questions

Alternative Reference Rates (ARRs) such as €STR, SOFR and SONIA are overnight interest rates that are used as benchmark rates for interbank lending. A variety of contracts are referenced with these overnight rates per currency zone. It is of interest to the financial world to forecast these rates to get a better understanding of future cash flows. For example, the interest amount to be paid for some cash products that are referenced with overnight rates is only known at the end of the interest rate period. If we can accurately forecast the Alternative Reference Rates, then estimations of future cash flows can be made. A classical time series model that has been used to model and forecast a variety of different data sets including financial data is the Autoregressive Integrated Moving Average (ARIMA) model. The ARIMA model contains an Autoregressive (AR), integrated (I) and a Moving Average (MA) part. Various input features can be implemented in the model by extending it to an ARIMAX model. Besides classical time series models, interest rate values can also be forecasted using machine learning algorithms such as Random Forest Regression. An ARIMAX and a Random Forest Regression algorithm requires the selection and testing of their input features. Overnight rates differ per country and each country has their own systems and regulations. Therefore, the expectation is that features that influence the Alternative Reference Rates differ per currency zone. This leads to the research objective for this thesis:

To develop a Random Forest Regression algorithm and compare its one-step forecasting performance on the €STR, SOFR and SONIA to the forecasting performance of an Autoregressive Integrated Moving Average model.

The research objective will be met by answering multiple research questions. We will divide the research into two parts. First, we will create a base model for ARIMA and Random Forest Regression that predicts the Alternative Reference Rates solely on their historical values, i.e. without any additional input features. Second, we will add input features to the models to slowly make them more complex. We choose this approach to systematically built up a complex model while maintaining interpretability. Hence, the first research question is:

Research Question 1: How effective is a Random Forest Regression algorithm in making a one-step prediction of €STR, SOFR and SONIA based on historical values as compared to an ARIMA(p,d,q) model?

The Random Forest Regression model is a complex algorithm that is subject to a wide range of different parameters that influence the fit of the model. The amount of decision trees that a Random Forest grows or how many input features are used per node to split the data are two examples of parameters that can be tuned and optimized. Let the coefficients $a_0, ..., a_p$ and $b_0, ..., b_q$ and the orders p, d, q be the parameters of an ARIMA(p,d,q) model.

Then, the following sub research questions will be answered:

1. What trends and seasonalities can be observed in historical €STR, SOFR, and SONIA data?
2. Which parameters and orders increases prediction accuracy in ARIMA(p,d,q) for €STR, SOFR, and SONIA data?
3. Which parameter combination increases accuracy for €STR, SOFR and SONIA prediction using Random Forest Regression?

Next, the models can be extended by adding input features. This leads to the second research question.

Research Question 2: How effective is a Random Forest Regression algorithm in making a one-step prediction of €STR, SOFR and SONIA based on input features as compared to an ARIMAX(p,d,q) model?

The ARIMAX model will be a natural extension to the ARIMA model that is calibrated for the first research questions. Therefore, the parameters and orders of the ARIMA model will be taken for the ARIMAX model as well.

1. Which input features increase prediction accuracy of an ARIMAX(p,d,q) model on €STR, SOFR, and SONIA data?
2. Which input features increase prediction accuracy of a Random Forest Regression algorithm on €STR, SOFR, and SONIA data?

Finally, we will compare our results to a simple model that is currently used to determine an interest value of the next day. The simple model simply uses the value of the rate of a given day as the forecasted value of the rate of the next day. Therefore, we formulate our last research question as follows.

Research Question 3: How effective are a Random Forest Regression algorithm and ARIMA(X) model in making one-step predictions for €STR, SOFR and SONIA as compared to the simple model that is currently used?

The results of this analysis shall be used to make recommendations about whether it is suitable to use a Random Forest Regression algorithm for Alternative Reference Rate prediction as compared to ARIMA or ARIMAX.

1.5 Thesis outline

We discuss the theoretical background for the research in detail in Section 2. The section includes some relevant background information on €STR, SONIA and SOFR. We discuss different classical time series models and machine learning algorithms and explain why we chose ARIMA and Random Forest for this research. Section 3 contains the research methodology including the research approach and execution. Section 4 contains the results on the prediction of the ARRs based on historical values. Section 5 contains the results on the prediction of ARRs based on market and mathematical features. We will compare the forecasting results with a prediction model that is based on the simple assumption that the overnight rate of tomorrow is equal to the overnight rate of today. This comparison is presented in Section 6. Finally, we discuss the results and formulate recommendations for practice in Section 7.

2 Theoretical background

In this section, the theoretical background is presented which is needed to understand the forecasting of the three overnight interest rates Euro Short-Term Rate (€STR), Sterling Overnight Index Average (SONIA) and Secured Overnight Financing Rate (SOFR) which serve as benchmark risk-free rates for a wide range of financial instruments. First, the overnight interest rates are described in more detailed and their characteristics are analyzed. Next, classical time series models and machine learning algorithms are presented and their forecasting effectiveness discussed. Furthermore, we will present advantages and disadvantages of different models and conclude by explaining the reasons why we chose ARIMA and Random Forest to make one-step predictions of the €STR, SONIA, and SOFR. The section also contains in depth explanations on the mathematics and the background of ARIMA and Random Forest.

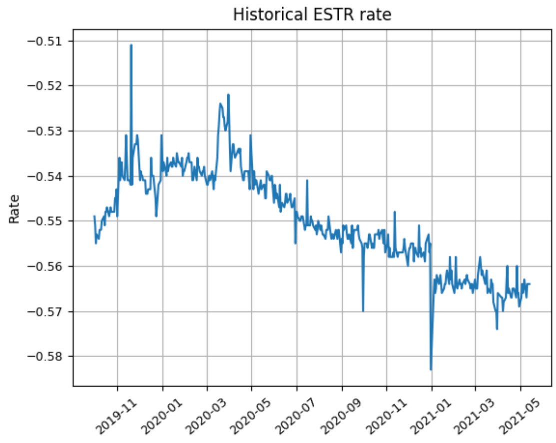
2.1 Benchmark overnight interest rates

The development of overnight rates have an impact on investors profit through financial instruments that are used for financing or hedging purposes. Therefore, the goal of this research is to efficiently forecast Alternative Reference Rates (ARRs). ARRs replace the London Interbank Offered Rates (LIBORs) as the benchmark rates for interbank lending. The Euro Short-Term Rate (€STR), Secured Overnight Financing Rate (SOFR), and the Sterling Overnight Index Average (SONIA) are the new benchmark rates for the currency zones EUR, USD, and GBP, respectively. Figure 1 shows the plots for historical €STR, SOFR, and SONIA values, respectively. Data for SONIA is available since 23 April 2018, €STR data since 01 October 2019, and SOFR data since 02 April 2018. Table 2 summarizes the basic statistic for €STR, SONIA, and SOFR. Data points are collected until 17 May 2021.

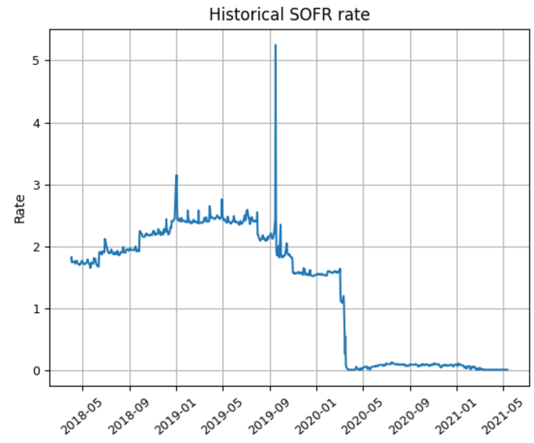
Statistic	€STR	SONIA	SOFR
Data Points	414	773	780
Minimum (in %)	-0.58	0.04	0.01
Maximum (in %)	-0.51	0.71	5.25
Mean (in %)	-0.55	0.44	1.30
Variance (in %)	0.0001	0.0932	1.0137
Standard Deviation (in %)	0.011	0.3053	1.0068

Table 2: Summary statistics for historical ARR data.

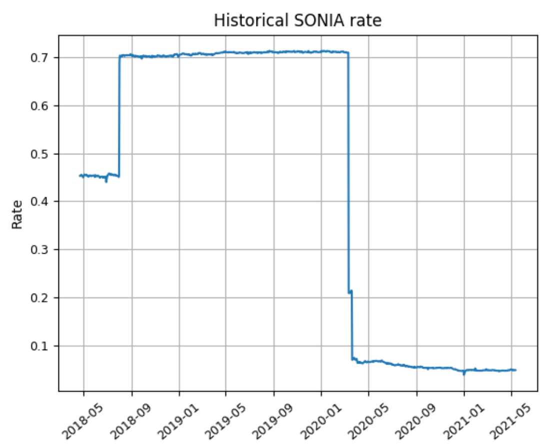
The European Central Bank (ECB) started publishing the €STR as shown in Figure 1a in October 2019 and continuously published it on every business day. This Euro-zone interest rate is used by a wide range of entities and is the benchmark for unsecured overnight interbank borrowing. An unsecured loan refers to a loan that is based on the creditworthiness of the borrower instead of on disposing any form of collateral. The rate is based on unsecured transactions that reflect the activities of the previous business day. The €STR is calculated by taking the previous values and removing the highest and lowest 25% of values. The remaining 50% of values are weighted according to the total volume of transactions. The mean of these weighted observations reflects the current interest rate value [8]. The values typically remain within the range $[-0.583\%, -0.511\%]$.



(a) Historical values for €STR. Data points are collected from a period of 02 October 2019 until 13 May 2021.



(b) Historical SOFR values. Data points are included from a period of 23 April 2018 until 13 May 2021.



(c) Historical SONIA values. Data points are included from a period of 02 April 2018 until 13 May 2021.

Figure 1: Historical ARR values.

The SOFR, shown in Figure 1b, is published daily by the New York Federal Reserve Bank and reach a transaction volume of around one trillion USD daily [9]. This rate is a secured interest rate, i.e. it is not based on the creditworthiness of the counterpart. The benchmark rate is calculated similarly to the €STR as a median of 50% of sorted past rates weighted by volume of transactions [10]. Next to the SOFR, the Federal Reserve Bank of New York additionally publishes the Tri-Party General Collateral Rate (TGCR) and the Broad General Collateral Rate (BGCR). The SOFR, TGCR, and the BGCR are all Treasury Repo Reference Rates. The TGCR is a secured overnight rate that is based on transactions between counterparties that are familiar with each other. The BGCR adds transactions on General Collateral Finance (GCF) to the TGCR. GCF refers to repurchase agreements in which the collateral is defined only on the end of the transaction period.

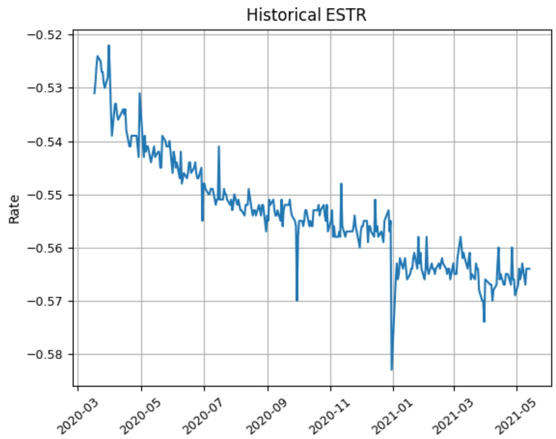
The SOFR rate is based on the same data as the BGCR and on additional transactions proposed by the Fixed Income Clearing Corporation's Delivery-versus-Payment (DVP) repo service. In these additional transactions, collateral to secure the lending activities are replaced by specific securities [11].

The SONIA, shown in Figure 1c, is published daily by the Bank of England. The SONIA is an unsecured interest rate. Similarly to €STR and SOFR, this benchmark interest rate of the GBP currency zone is based on actual transactions. SONIA was first introduced in March 1997 but was revised in 2018 [12]. We only include the data points for the revised SONIA in this analysis. This benchmark is calculated as volume-weighted mean of data that is reduced by 25% of highest and 25% of lowest values [13].

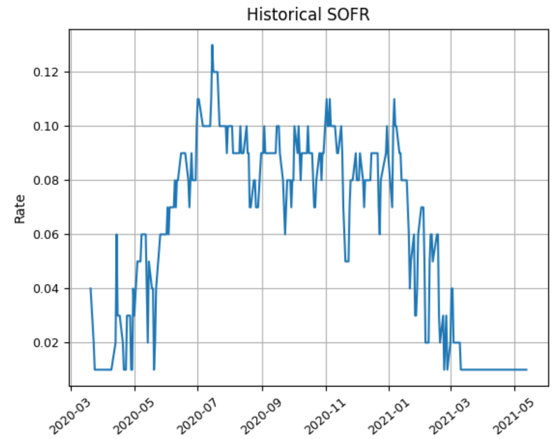
The behavior of the SONIA rate as shown in Figure 1c is distinct from the behavior of the €STR (Figure 1a) and of SOFR (Figure 1b). In Figure 2 we can compare the rates to each other on the same time window in contrast to Figure 1 where we can see the rates on different time windows. The figure shows the interest rates from March 2020 until May 2021.

The €STR and the SONIA follow a similar underlying process as both rates decrease over time and have a distinct minimum around January 2021. The SOFR clearly follows a different underlying process as compared to €STR and SONIA since this rate first increases before it decreases later on. All series include some kind of direction and random fluctuations. More details about characteristics of the behaviour of the interest rates such as trends, seasonal components and random fluctuations is described in the Section 2.2.

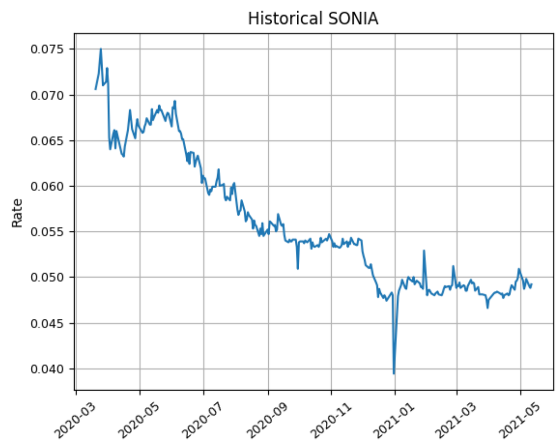
Another distinctive aspect in the historical SOFR and SONIA rates that can be observed in Figure 1 are the sudden jumps in value, either in form of a rapid increase or decrease. The SOFR (Figure 1b) shows a peak in September 2019. The bank reserves declined by around 120 billion USD within two business days while there were more Treasury securities that had to be financed during that time than usual. This mismatch led to the sudden peak in SOFR on September 16 [14]. Furthermore, the SOFR shows a downward jump in March 2021. This is because the New York Federal Reserve Bank lowered the target rate to $0\% - 0.25\%$ on March 15 as a response to the Covid-19 crisis [15]. Also the SONIA shows two specific jumps.



(a) Historical values for €STR.



(b) Historical SOFR values.



(c) Historical SONIA values.

Figure 2: Historical ARR values. Data points for a period from March 2020 until May 2021 are included.

The first jump occurred on August 2018 when the Bank of England raised SONIA from 0.5% to 0.75% [16]. The second jump can be observed in March 2020 when the Bank of England decided to lower the base rate to 0.25% in response to Covid-19 [17].

The €STR, SONIA and SOFR are all sequences of interest rate values that are collected over time. Therefore, the interest rates can be considered time series. Different methods to model and forecast time series are discussed in the next section.

2.2 Classical time series models

Time series such as Alternative Reference Rates (ARRs) can be modelled and predicted using a variety of methods. The goal of this research is to establish a forecasting method that can effectively predict future values of ARRs. We will discuss different time series analysis models in this section. Furthermore, we will present advantages and disadvantages and choose which forecasting methods to use to predict the €STR, SONIA, and SOFR.

In a general approach to time series analysis, a series is described by a trend, a seasonal component and an incidental component. A trend reflects a general up or down movement which can also switch direction over the course of time [18]. As an example, the SOFR values from March 2020 until May 2021 in Figure 2b show an upward trend up until approximately August 2020 followed by a downward trend up until approximately mid March 2021. Figures 2a and 2c on the other hand show a continuing downward trend on the time period from March 2020 until May 2021 without changing its direction. If observations repeat themselves over the course of time, then the time series shows periodic behavior and includes a seasonal component [18]. The figures on historical €STR, SOFR and SONIA do not show visually observable seasonal components. The last characteristic of a time series, which can be observed in all three ARRs, is the incidental component. Figure 1 shows that all €STR, SOFR, and SONIA have random fluctuations along their paths.

A natural feature of time series that follows from the trend, seasonal component and incidental component is memory. Memory is a distinctive feature of time series and key for successful prediction. It describes the dependence of one value of the series on a past value of the series and can be captured by the covariance and the autocorrelation function [18]. Let X_t be a time series and X_s be a lagged version of that same time series. Then, the covariance function $\gamma(t, s)$ of a time series X_t given by

$$\gamma(t, s) = cov(X_t, X_s) = \mathbb{E}[(X_t - \mathbb{E}[X_t])(X_s - \mathbb{E}[X_s])] \tag{2.1}$$

where t and s are two time indices. The autocorrelation function $\rho(t, s)$ is the normalized covariance function given by

$$\rho(t, s) = \frac{\gamma(t, s)}{\sqrt{\gamma(t, t)\gamma(s, s)}} \tag{2.2}$$

where $\rho \in [-1, 1]$ and s and t are two time indices. If $t = s$, then $\rho(t, s) = \rho(s, s) = \rho(t, t) = 1$ [18].

As an example, if the autocorrelation between a time series consisting of daily observations and a one day lagged version of itself is close to one, then there is a strong positive relationship between each value of the series and the preceding value. The downside of using the autocorrelation function as a measure of relationship is that it does not take into account a possible influence by other lags. For example, if the autocorrelation of a series $X(t)$ with its lagged version $X(t+4)$ is high, then another high correlation with lag $X(t + 5)$ might be purely due to the strong relationship at lag 4. The autocorrelation function does not capture this phenomenon. However, the partial autocorrelation function does take previous lags into account [18].

The autocorrelation and partial autocorrelation functions can be used to form a class of models that describe and forecast time series data such as interest rates and are called Autoregressive Moving Average (ARMA) models [18]. ARMA(p,q) models consist of an Autoregressive (AR) process of order p and a Moving Average (MA) process of order q. An Autoregressive process A_t of order p satisfies

$$A_t = a_1 A_{t-1} + a_2 A_{t-2} + ... a_p A_{t-p} + \epsilon_t \tag{2.3}$$

where $a_0, ..., a_p \in \mathbb{R}$, $a_p \neq 0$ and ϵ_t is white noise. White noise reflects a purely random error part of the process. This is similar to the incidental component described above. A Moving Average process M_t of order q can be described as

$$M_t = b_0 \epsilon_t + b_1 \epsilon_{t-1} + ... + b_q \epsilon_{t-q} \tag{2.4}$$

where $b_0, ..., b_q \in \mathbb{R}$ and $\epsilon_i, i \in \mathbb{Z}$ is white noise. Combining the AR and MA process yields an equation of an ARMA process X_t of order (p, q) given by

$$X_t = b_0 \epsilon_t + b_1 \epsilon_{t-1} + ... + b_q \epsilon_{t-q} + a_1 X_{t-1} + a_2 X_{t-2} + ... a_p X_{t-p} \tag{2.5}$$

where $t > 0, t > p, t > q$ [18]. One key assumption of the ARMA(p,q) model is that the time series needs to be stationary, i.e. its probability distribution remains the same after a time shift. A stationary series has the property that statistical information such as mean and standard deviation do not change over time [18]. Considering Figure 1, this property is not satisfied for €STR, SOFR or SONIA. In fact, this property is rarely satisfied in financial applications of time series. The ARMA(p,q) model can be extended to an Autoregressive Integrated Moving Average (ARIMA) model by adding a term - called the differencing term - to Equation 2.5 which makes the series stationary. The differencing method removes any trends and seasonal components of the data and is called called seasonal differencing [19]. Let x_t be the observations of the process X_t. Then, a differencing term is given by

$$x'_t = x_t - x_{t-d} \tag{2.6}$$

where d is the order of differencing [19]. This results in an ARIMA(p,d,q) model proposed by Box and Jenkins (1970) [20]. Clearly, setting $d = 0$ in an ARIMA(p,d,q) model results in the ARMA(p,q) model. ARIMA has been widely used in forecasting time series data from a variety of different fields [21]. ARIMA models are also extensively used in finance to model data such as stock prices (e.g. [4]) and exchange rates (e.g. [5]). The main advantage of an ARIMA(p,d,q) model is that it can handle small data sets. The model does not require a large number of data points to represent an accurate fit for the time series [22]. The main disadvantage of ARIMA models is that they are not efficient in capturing volatility clustering effects.

Volatility clustering refers to the fact that large changes tend to cluster together [23]. This means that periods of high volatility and periods of low volatility appear in the time series data. For example, if volatility increases drastically due to some significant market event, then volatility clustering means that this shock will continue for some time in the future. The family of Autoregressive Conditional Heteroskedasticity (ARCH) models proposed by Engle (1982) [24] account for volatility clustering or volatility patterns. Bollerslev (1986) proposed a Generalized ARCH (GARCH) model [25]. ARCH models represent volatility as linear functions of past variances, whereas GARCH models also account for lagged volatility effects.

Radha and Thenmozhi compared different versions of ARMA and GARCH models regarding their performance to forecast short-term interest rates. They found that an ARIMA-GARCH combination performs best in forecasting overnight rates [26]. However, the scope of this research does not include to forecast the volatility of the overnight rates. Therefore, we disregard (G)ARCH models for further analysis.

As established above, ARIMA is a suitable model to fit and forecast financial data including interest rates. Prediction with an ARIMA model is straightforward, as it uses the historical data points as an input and gives the prediction of one or multiple data points as an output. An extension of the ARIMA called ARIMAX takes multiple input variables (e.g. other time series) as an input. These input variables are called explanatory or exogenous variables. The ARIMAX might be preferred over ARIMA as financial data has been known to depend on multiple macroeconomic factors [27]. Taking these factors into account for prediction might increase the prediction accuracy of the model.

Classical time series analysis indicates that the ARIMA model is well suited in forecasting financial data including overnight interest rates. However, ARIMA has limitations as it is constrained in handling sudden, irregular fluctuations within the data [28]. Figure 1 shows that all three overnight rates show sudden fluctuations regularly. Therefore, different types of models might be better suited for forecasting ARRs. In the next section, we will present Machine Learning as another class of forecasting methods.

2.3 Machine leaning algorithms

Artificial intelligence is a method to imitate human intelligence in form of programs and algorithms. Central characteristics of human intelligence are learning, reasoning, and perception abilities which are simulated using artificial intelligence. Machine learning is a discipline that falls under the broader term artificial intelligence and includes teaching programs to autonomously build models and algorithms. In addition, deep learning is a subspace of machine learning which is based on artificial neural networks that consist of multiple, also called deep, layers [29]. Figure 3 shows a visualization of the relationship between artificial intelligence, machine learning, and deep learning.

The three main approaches to machine learning are supervised, unsupervised, and reinforcement learning. Data that is fed to a supervised machine learning algorithm contains input and output labels. Unsupervised machine learning algorithms are characterized by a lack of output data fed to the algorithm. Lastly, reinforcement learning involves algorithms to adapt to their environment to maximize some predetermined notion of reward. Furthermore, machine learning algorithms can be divided into Classification and Regression algorithms. Classification algorithms allocate a final decision class, whereas Regression algorithms allocate a final decision value. For example, a Classification algorithm can predict the direction of interest rates by classifying the future value into the classes ”the rate increases”, ”the rate remains the same”, and ”the rate decreases”. A Regression algorithm on the other hand predicts an actual quantitative value for the interest rate [30].

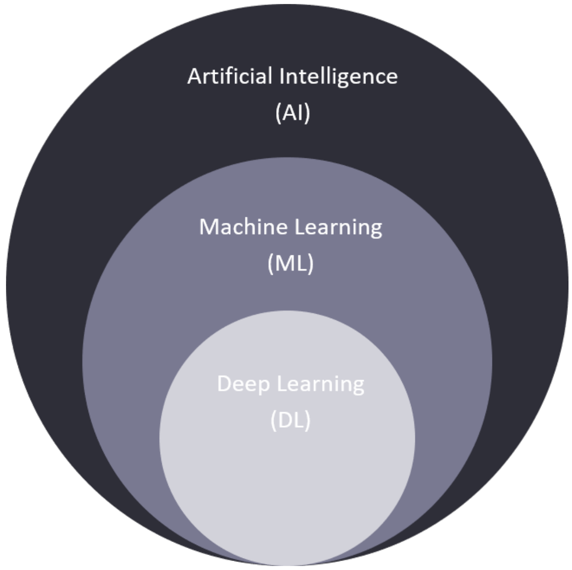


Figure 3: Hierarchy of artificial intelligence, machine learning, and deep learning.

There exists a wide range of machine learning algorithms. Commonly well known machine learning algorithms can be roughly described as regression algorithms and include Ordinary Least Squares Regression and Logistic Regression [31]. Ordinary Least Squares Regression fits a dependent variable with a linear combination of independent variables and minimizes the sum of squared errors which is the difference between the residual and the actual value squared [32]. Logistic Regression works similarly but is applied when the dependent variable is binary [33]. Another type of machine learning algorithm makes comparisons between data points to find best fits. Examples are k-Nearest Neighbor, Learning Vector Quantization or Support Vector machines [31]. Decision Tree algorithms are another common form of machine learning which - combining multiple Decision Tree models - result in Random Forest. Decision Trees and Random Forests are explained in more detail later in this chapter. Finally, Artificial Neural Networks or Deep Learning mimics the structure of biological neural networks such as in the brain [31]. Caruana and Niculeszu-Mizil (2006) compared all of these algorithms on their performance on different kinds of data sets. On average, Random Forest outperformed the other algorithms. The methods that performed poorest on average include Naive Bayes, Logistic Regression and single Decision Trees. However, the authors suggests that performance of models heavily depends on the specific data set [7].

Other research focuses on forecasting performance in machine learning algorithms and classical models. Kaidhem et al. (2016) used a Random Forest Classifier to forecast the direction of stock prices and compared their results to findings from researches on other models. They found that a Random Forest Classifier outperforms other methods such as Support Vector Machines, Naive Bayes and Neural Networks [34]. However, Neural Networks have been found to successfully forecast interest rate data [35]. More specifically, research has shown that a recurrent neural network can outperform ARIMAX for SOFR prediction [36]. Rundo et al. (2019) conducted an extensive survey on the use of classical forecasting algorithms and machine learning algorithms on financial data. Most researchers used data sets on stock markets or indices. Studies on classical methods used ARIMA, GARCH or ARIMA-GARCH hybrid models to forecast financial data.

Studies with a machine learning approach used Support Vector Machines, K-Nearest Neighbor, Random Forest or Neural Networks. In general, the shortfall of classical methods is that they are less adequate in handling big numbers of data points and to determine hidden patterns as compared to machine learning algorithms. Regarding machine learning algorithms, Support Vector Machines are limited in predicting financial data. One main conclusion of the survey is that prediction performance heavily depends on the kind of data that is examined. It is not possible to define the single best machine learning algorithm [6].

Two promising methods to predict financial time series are Artificial Neural Networks and Random Forests. Artificial Neural Networks can handle complex and large data sets and make forecasts without prior assumptions on the data. Also more advanced variations of the Artificial Neural Network have proved useful in financial data forecasting [6]. Furthermore, Artificial Neural Networks have the ability to perform multiple predictions at the same time. The biggest disadvantage of a Neural Network is that we know input and output of the network but it is difficult to trace and understand what happens within the model itself. Furthermore, Artificial Neural Networks have potentially a high number of parameters that need to be tuned [37]. Random Forests make up for some of these disadvantages. Random Forests have been shown to perform well on a variety of different data sets including data sets with missing values [6]. Furthermore, Random Forests are relatively simple to use, include a small number of parameters that have to be tuned and can deal with large as well as small sample sizes [38]. The main disadvantage of a Random Forest is that as the number of trees increases, the computation time can become ineffectively slow [39].

The common ground of different research papers is that the performance of machine learning algorithms heavily depends on the data set. Based on the survey outcomes mentioned above, Neural Networks and Random Forests seem somewhat superior to other forecasting methods when it comes to predicting financial data. Considering the limitations of the Neural Network and the Random Forest, we decide to use the Random Forest algorithm to predict €STR, SOFR, and SONIA. Computation time can be controlled by limiting the number of trees that the Random Forest grows. Furthermore, data sets for the interest rates have a manageable size. In addition, the prediction of overnight rates is of primary concern to members of the financial world. Therefore, it is of interest that they can comprehend the method despite being unfamiliar with machine learning. The mechanics of a Random Forest are explained in more detail in the following section.

2.4 Random Forest

The Random Forest is an example of a supervised machine learning algorithm and was first proposed by Breiman in 2001 [40]. Random Forest algorithms can be calibrated to be classification or regression algorithms. A classification algorithm sorts the input data into predefined classes. For example, a classification algorithm on interest rates could be used to predict whether the interest rate tomorrow falls in the class 'interest rate increases' or 'interest rate decreases'. The Random Forest Regression algorithm on interest rates can be used to predict the actual value of the rate tomorrow. A Random Forest Regression model is an ensemble of decision trees, i.e. it consists of a predefined number of decision trees. Figure 4 visualizes a decision tree and shows the names of different types of nodes.

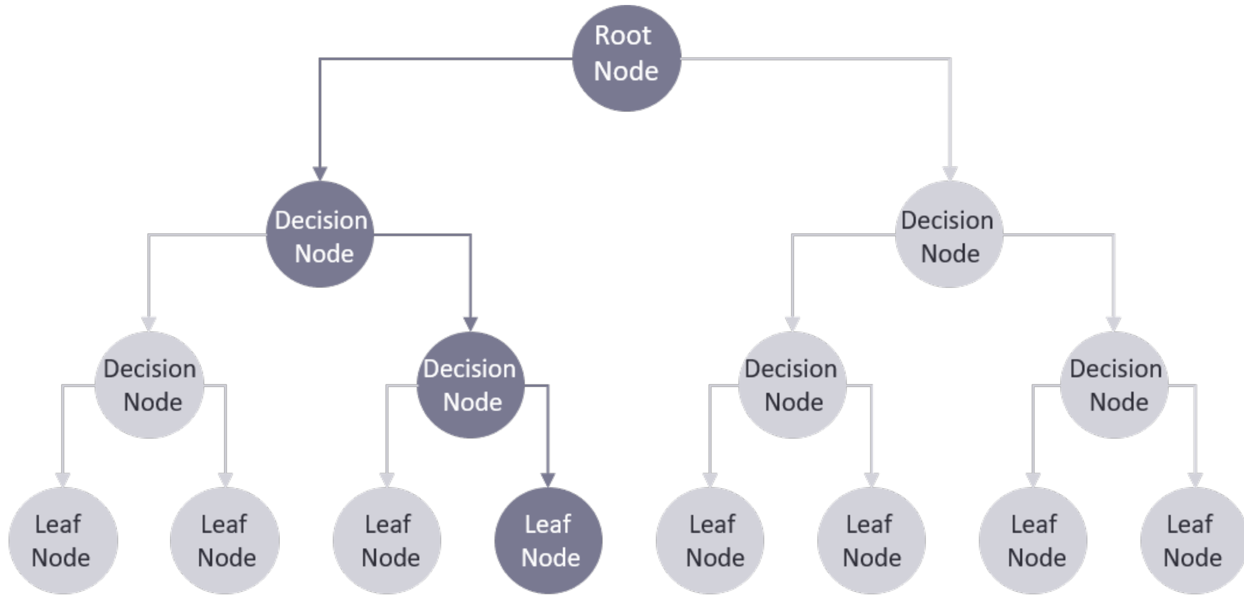


Figure 4: Visualization of a decision tree including the different names of nodes.

The starting point of a decision tree where the first decision is made is called the root node. The subsequent nodes are called decision nodes which split the data set into smaller chunks of data. Decisions are based on features that are given to the model as input factors. The final line of nodes are called leaf or terminal nodes and indicate that a final decision is reached. Each decision tree reaches one final prediction. A simple example of an application is shown in Figure 5. The goal is to predict the colour of an object. The input features are whether the object has corners and how many corners the object has. The root node begins with the whole data set. The first decision that has to be made is whether the object has corners or no corners. Based on the total data set, objects with no corners are blue and objects with corners are either green or yellow. The next decision is whether the object has three or four corners. Objects with three corners are green and objects with four corners are yellow. The previous example shown is very simple and limited. An advanced example is to predict the value of an overnight interest rate for tomorrow. An input feature could be the number of transactions that were referenced with that rate today and the total volume of these transactions.

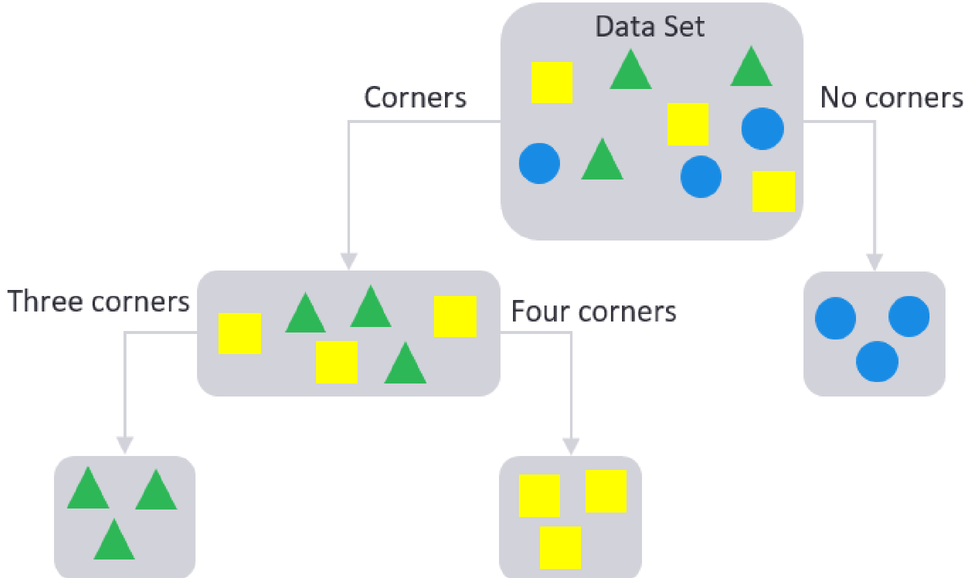


Figure 5: Example of a decision tree that predicts the colour of an object based on its geometric features.

The root node is already more complex than in the colour of the objects example as the overnight rate can take on a wide range of numbers. For example, the €STR is historically within the range $[-0.583\%, -0.511\%]$ (cf. Table 2). A possible decision could be whether the total volume of transactions is above or below a certain number to split the data sets into smaller pieces. A further decision node could split the data based on whether the number of transactions is above or below a certain value. Clearly, we still do not know the value of the overnight rate tomorrow based on these two decisions. More complex decision trees are necessary. Furthermore, decisions could be ambiguous in the sense that one decision tree reaches a certain value and another decision tree reaches another value. That is why a lot of complex decision trees are formed in the Random Forest. Let n be the number of decision trees in a Random Forest. Each of these trees makes a prediction of the overnight rate value of tomorrow which results in n predictions. The Random Forest calculates the average number of predictions to form a final prediction [30]. For example, if $n = 3$ there are three decision trees. Assume that the first decision tree predicts that the value of the €STR is going to be -0.5 tomorrow, the second tree predicts a value of -0.54 and the last decision tree predicts -0.48. Then, the Random Forest calculates the final prediction as $\frac{-0.54-0.54-0.48}{3} = -0.52$. Hence, the Random Forest predicts that the value of the €STR tomorrow is -0.52. Figure 6 visualizes a Random Forest Regression algorithm.

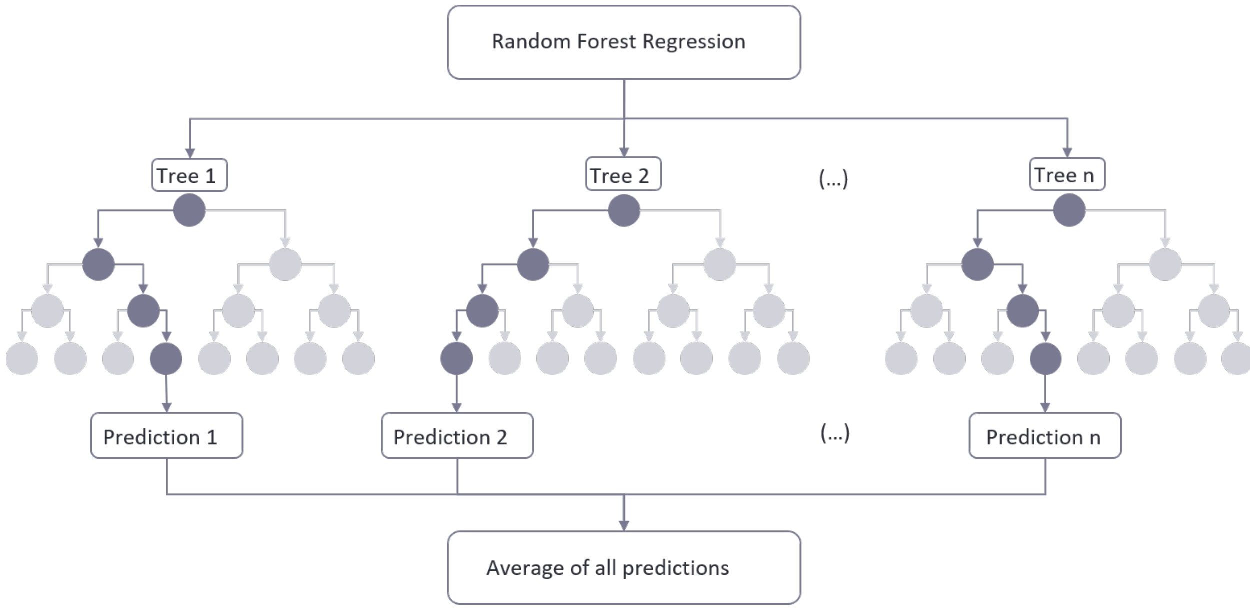


Figure 6: Visualization of a Random Forest Regression algorithm.

The data set is split into smaller chunks of data at each decision node. In Figure 5, the total data set consisting of triangles, circles and squares is split into one data set consisting of squares and one data set consisting of triangles and squares. At the next decision, the piece of data that consisted of triangles and squares is split into one smaller data set consisting of triangles and one smaller data set consisting of squares. The goal of the algorithm is to arrive at a data set that has as least impurity as possible. The example with the objects reaches that goal as in the terminal nodes, each data set consists purely of triangles, squares or circles.

Figure 7 visualizes the splitting of the data set at each decision node.

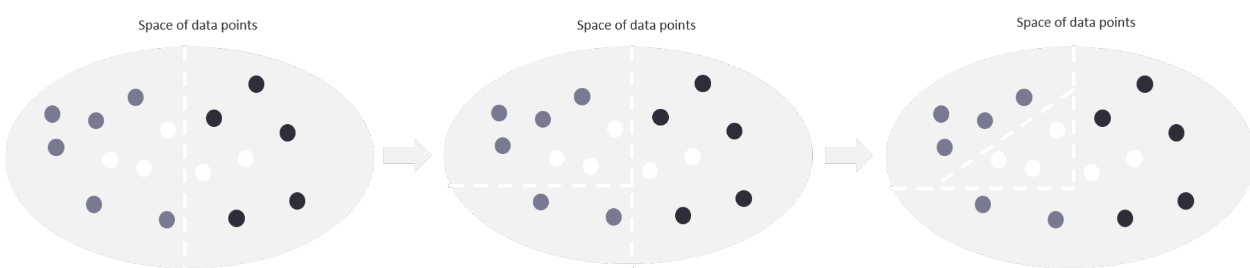


Figure 7: Basic visualization of the splitting of the data set at each decision node. The goal is to reach a final decision of "white", "grey", or "black". Each dotted line represents a split at a decision node which makes the data set smaller.

In this example, pure data refers to having the same colour of dots in the data set after a decision. After the final decision, impurity is minimized since only two sets of data remain. One set contains only grey data points and the other set contains only white data points. When white and grey dots are mixed, then the data set is impure.

If we would add even more colours to the data set, then the decision making process with the prospect of minimizing impurity gets more and more complex [30]. The quality of each split can be calculated according to the impurity measure. The impurity measure quantifies the information gain. At each decision node, the information gain shall be maximized. Each dot in Figure 7 would have a different colour in a Random Forest Regression algorithm as each dot represents a single value of the data set. Therefore, the impurity of a split in an Random Forest Regression algorithm is calculated according to the mean squared error. The mean squared error of node a is given by

$$I(a) = mse(a) = \frac{1}{N_a} \sum_{i \in D_a} (y(i) - \widehat{y}_a)^2 \tag{2.7}$$

where N_a is the number of data points at node a, D_a is the corresponding subset at node a for the split, y_a is the actual and $\widehat{y_a}$ the predicted value at node a [30].

The objective at each node is to maximize $I(a)$ which is also called the objective function. In individual decision trees, the depth of the tree (i.e. the number of decision nodes) highly influences the quality of the prediction. Furthermore, the algorithm of an individual decision tree can overfit the data. If the data is fed to one single decision tree for learning, then it might perform above average on the data that it is trained on but poorly on new data. A Random Forest avoids overfitting by growing a large number of decision trees and basing its final decision on the outcomes of all individual trees. The Random Forest Regression is less susceptible to overfitting due to the randomness component. To execute an Random Forest Regression algorithm, the data set first has to be split into training and test data. The training data is fed to the algorithm as a basis for learning. The testing data is used after the algorithm has been built to see how accurate the predictions of the algorithm are. Common training to test split ratios are $60 : 40$, $70 : 30$, or $80 : 20$, depending on the size of the data set. If the testing data set is too big, then valuable information is withhold from the algorithm. On the other hand, if the testing data is too small, then the algorithm might overfit to the training data. After an appropriate training to testing split ratio has been chosen, the decision trees are grown as follows: The algorithm draws a random sample with replacement from the training data set. At each node, a random number of features is chosen. From the pool of features, the one which minimizes the mse objective function is used to split the node. This procedure is repeated until a number n of decision trees is grown. Splitting is based on the mse of values in the remaining data set after applying certain features. Therefore, the quality of the features that are given as an input to the Random Forest Regression algorithm can greatly influence the quality of the predictions of the algorithm [30].

When forecasting the value of an overnight rate, we chose the number of transactions and the total volume of transactions as input features. The example of a decision tree that was described was not adequate in predicting the value of the interest rate. This shows that the selection of features greatly influences the quality of the split. If the split is able to divide the data set into more impure data sets such as in the example with the colours of the objects, then the prediction in the end will be much more adequate. Assuming that triangles are always green, circles are always blue and squares are always yellow, we will always be able to identify the colour green based on the shape of the object. Hence, prediction depends heavily on the splits of the decision trees. Figure 8 summarizes the above information on how the Random Forest works as a basic visualization. The total data set consists of features and outcomes.

For example, the price of a selected stock could be the outcome that is supposed to be predicted and the features could include mathematical representations such as the price rate of change [34]. Then, the data set is split into training and test data as described above. The training data is fed to the machine learning algorithm. In this case, it is a Random Forest regression algorithm. This results in a calibrated model that can be used to make predictions. The test data is fed into the calibrated model. Finally, the predicted values can be compared to the actual values to assess the performance of the calibrated model.

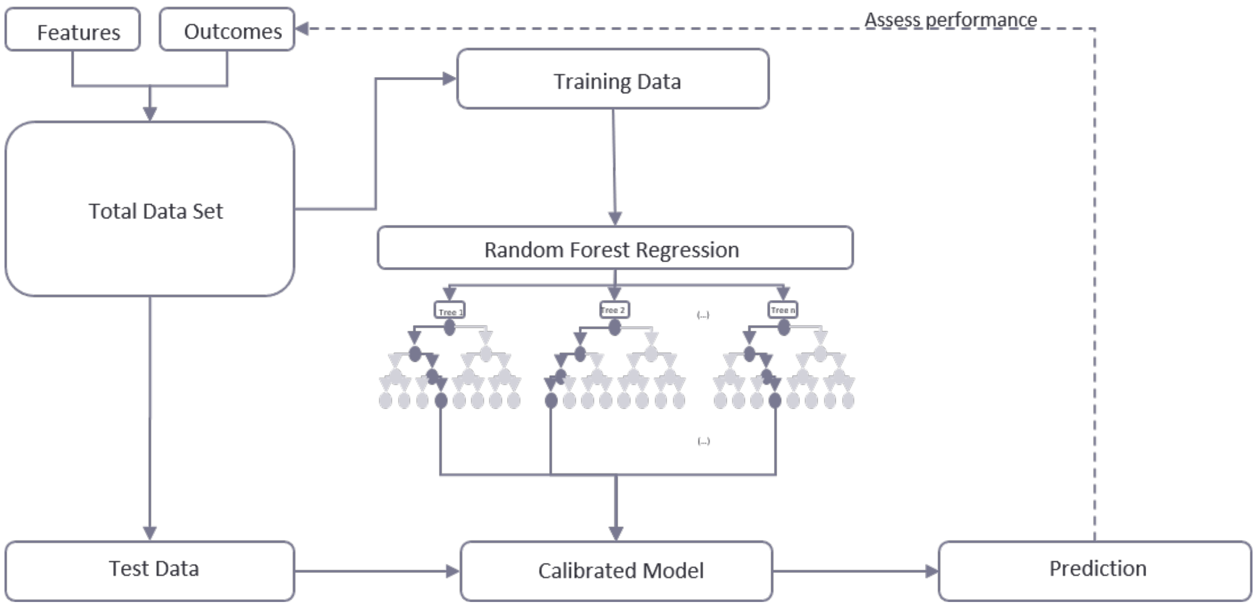


Figure 8: Visualization of a basic Machine Learning framework.

We discussed different methods to forecast interest rates in this chapter and arrived at two promising models, namely ARIMA(x) and Random Forest. Next, we will present the research method in Section 3 followed by presenting the results in Section 4 and 5. Then, we will move on to a comparison of the results to a simple prediction model in Section 6. Finally, we will arrive at a discussion and recommendation for further research in Section 7.

3 Methodology

The goal of this research is to develop a Random Forest Regression algorithm and compare its forecasting performance on the €STR, SOFR and SONIA to the forecasting performance of an Autoregressive Integrated Moving Average model. First, we will focus on creating and optimizing base models that do not include input features and are solely based on historical values. Second, we will extend the models to include input features. The ARIMA model will then become an ARIMAX model.

We chose to use Python as a programming language for this project because it is open source with a large support and the methods are suitable to implement in the given time frame. Furthermore, Python is used within EY which makes it possible to reuse the calculations and algorithm in the future. Historical data for SONIA, €STR, and SOFR are openly accessible through the internet [41, 42, 43]. Data for SONIA is available since 23 April 2018, €STR data since 01 October 2019, and SOFR data since 02 April 2018. Data points are gathered until 17 May 2021. Furthermore, the training to testing data split ratio is relevant for both ARIMA and Random Forest. Common splits are 60:40, 70:30 or 80:20 depending on the size of the data set [30]. For the €STR, the data set is relatively small. Therefore, we will use a training to testing split ratio of 70:30. We will use a 80:20 split for SOFR and SONIA. The data will not be shuffled before applying the split. This means, that the most recent 30% (for €STR) and 20% (for SOFR and SONIA) of interest rate values will be assigned to the testing data set.

3.1 Procedure for ARIMA calibration

As described in the section on Theoretical Background, the ARIMA process consists of an Autoregressive (AR), an Integrated(I) and of a Moving Average (MA) term. The degree of these terms determine the order of the ARIMA process. Let p be the degree of the AR part, d the degree of the difference term, and q the degree of the MA part. Then, p, d, q have to be determined such that ARIMA(p,d,q) fits our data set. The ARIMA model assumes that the time series is in the form of equation 2.5. Hence, the coefficients $a_0, ..., a_p$ and $b_0, ..., b_q$ need to be determined as well. The values for p, d and q can be determined by examining the autocorrelation function and the partial autocorrelation function. The autocorrelation and partial autocorrelation functions can be plotted using the module *statsmodels* that is integrated in Python [44]. The value at which the plots are cut-off can be used as an initial Autoregressive order. Then, the values for p, q and d will be reduced slightly to check whether a different combination fits better to the data. Let X be the size of the training data set as used in the Random Forest model per Alternative Reference Rate. We will cut off $(1 - X)\%$ of the latest data points before fitting the model. Then, the ARIMA(p,d,q) will be fitted to $X\%$ of the data. This embedded function estimates the coefficients $a_0, ..., a_p$ and $b_0, ..., b_q$ using maximum likelihood estimation. Next, we will predict the values of $(1 - X)\%$ of the data by using a technique called walk-forward validation. After the model is trained a one-step prediction is made. Then, the prediction is stored and the training data is extended by the actual value of that predicted time step. Next, prediction is repeated with the new time window.

3.2 Input features for ARIMAX and Random Forest

The input features are used by ARIMAX and Random Forest Regression to make predictions and are expected to have a high influence on the quality and performance of the algorithms. There are two types of features that we will consider in the scope of this paper: Market features and mathematical features. There are different market features readily available for the €STR and SOFR that are published daily alongside the respective rates by the European Central Bank and the Federal Reserve Bank of New York [41], [42]. For the €STR, the total volume, number of active banks, number of transactions, 25th percentile volume, 75th percentile volume and share of volume of the five largest active banks is available per date. For the SOFR, the 1st, 25th, 75th and 99th percentiles are available as well as the volume. Furthermore, the SOFR is published alongside the Tri-Party General Collateral Rate (TGCR) and the Broad General Collateral Rate (BGCR) which are secured overnight interest rates as well. Furthermore, we will use the value of the previous business data of the interest rate as input feature. The €STR and the SONIA as shown in Figure 2a and 2c seem to be closely related. Therefore, we will also include the values of the previous business day from the SONIA for the €STR. It is not suitable to include €STR values in the SONIA prediction as the €STR began to be published later than the SONIA and thus, no corresponding data points for each SONIA data point exists.

We include the following mathematical features in our models. The Relative Strength Index (RSI) given by

$$RSI = 100 - \frac{100}{1 + A} \tag{3.1}$$

where A is the average increase of the interest rate over the past n days divided by the average decrease of the interest rate over the past n days. The Stochastic Oscillator (SO) given by

$$SO = 100 \times \frac{x_{t-1} - l}{h - l} \tag{3.2}$$

where x_{t-1} is the interest rate value at time $t-1$, l is the lowest price over the past n days and h the highest price over the past n days [34]. We will select the number n according to the autocorrelation and partial autocorrelation plots. The RSI and SO with different values for n will be used as input features. Furthermore, we will use the Exponential Moving Average (EMA) at time t given by

$$EMA(t) = (1 - \alpha)EMA(t - 1) + \alpha x_t \tag{3.3}$$

where x_t is the value of the interest rate at time t and $EMA(t_0) = x_{t_0}$. The parameter $\alpha \in [0, 1]$ is called the decay parameter and reflects the weight that is given to past observations [45]. The EMA with different values for α will be used as input features. In the testing and improving phase of the project, the input features will be evaluated according to their relative contribution to the performance of the Random Forest Regression.

3.3 Procedure for Random Forest calibration

The Random Forest Regression algorithm is somewhat more complex to calibrate as compared to ARIMA. The algorithm will be implemented using the machine learning library *scikit learn* [46]. The function requires several arguments that we will take into account which include

i. *n_estimators*: The number of decision trees that is grown.

ii. *criterion*: The criterion that is used for splitting. The choice is between mean squared error (mse) and mean absolute error (mae).

iii. *max_depth*: The maximum amount of nodes in one decision tree. The danger of using too many nodes is an over-fitting of the data set. The procedure of setting a limit to this number is called pruning the tree.

iv. *min_samples_split*: The minimum amount of data in a node that is necessary to split the node.

v. *min_samples_leaf*: The minimum amount of data that is necessary to form a final decision node.

vi. *max_features*: The maximum number of indicators at each split. By default, this value is $\sqrt{m}$, where m is the total number of indicators.

vii. *max_leaf_nodes*: The maximum number of leaf nodes.

viii. *min_impurity_decrease*: The minimum impurity value that is required to split a node.

ix. *bootstrap*: Relates to the sampling of data for growing the decision trees. The choice is between with and without replacement.

Table 3 summarizes the initial configuration of the input arguments. We chose to use the default values as given by the machine learning library since the arguments will be tuned in the improving phase of the project.

Argument	Setting
n_estimators	100
criterion	mse
max_depth	None
min_samples_split	2
min_samples_leaf	1
max_features	Number of features
max_leaf_nodes	None
min_impurity_decrease	0
bootstrap	True

Table 3: Initial settings for the input arguments for the RFR algorithm.

3.4 Testing and improving

To test the algorithms, we will run the programs on the test data and use the mean absolute error (mae), mean squared error (mse) and the root mean squared error (rmse) as performance measures. The smaller the errors, the better the model predicts the testing data. Let $(x_t)_{t\in\mathbb{N}}$ be the values of the test data set per interest rate and $(y_t)_{t\in\mathbb{N}}$ be the corresponding predicted values. Then, the mean absolute error is given by

$$mae = \frac{\sum_{t=1}^{n} |x_t - y_t|}{n} \tag{3.4}$$

and the mean squared error is given by

$$mse = \frac{\sum_{t=1}^{n} |x_t - y_t|^2}{n} \tag{3.5}$$

where n is the total number of data points in the test data set.

As we are comparing the performance of Random Forest and ARIMA, we will mainly be concerned which model yields lower mae and mse as compared to the other. We will further report the root mean squared error (rmse) which is given by

$$rmse = \sqrt{mse} \tag{3.6}$$

and reflects the standard deviation of the residuals.

The input features that are used for the Random Forest Regression model will be tested by calculating their relative contribution to the predictions of the model. Each indicator will be assigned a relative weight that reflects their importance to the algorithm. Next, the indicators can be ranked according to their importance and we can remove the least contributing indicators if the testing phase shows that the accuracy is sufficiently high. The advantage of removing indicators is to decrease computing cost and increase interpretability of the model. Finally, the parameters for the Random Forest Regression can be tuned. We will define a parameter grid that the function uses to perform a cross validation. This means that the program will go through all parameter settings that are specified in the parameter grid to see which setting yields the best Random Forest Regression algorithm performance. The function will give an optimized set of input argument settings. Since this type of grid search can be computationally expensive, we decide to define a parameter grid that the program randomly samples from to test the optimal parameter settings and run 500 iterations. Table 4 shows the parameter grid that we defined for parameter searching.

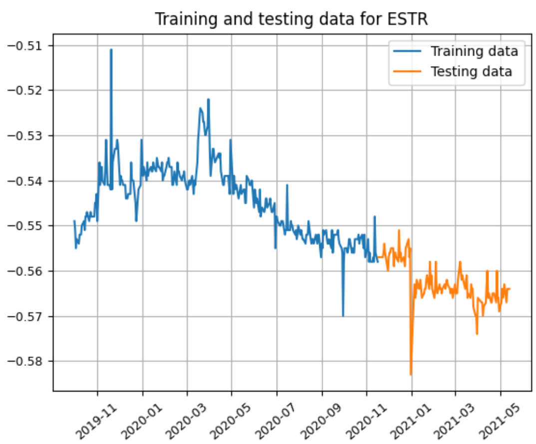
Argument	Setting
n_estimators	[1, 1500]
criterion	mse, mae
max_depth	[1, 100]
min_samples_split	[2, 10]
min_samples_leaf	[1, 10]
bootstrap	True, False

Table 4: Grid used for parameter optimization by randomly sampling 500 times.

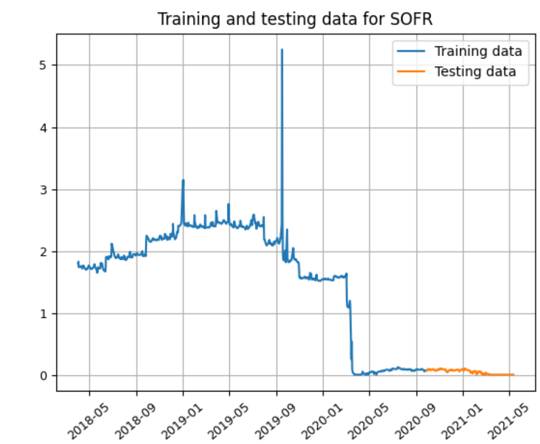
In this section, we discussed how we will calibrate the ARIMA, ARIMAX and Random Forest models to predict the €STR, SOFR and SONIA. We further discussed how we will assess the forecasting performance and improve the models once they are calibrated. The following two sections contain the results of our analysis. In Section 7, we will discuss the results and formulate recommendations and further research.

4 Prediction based on historical values

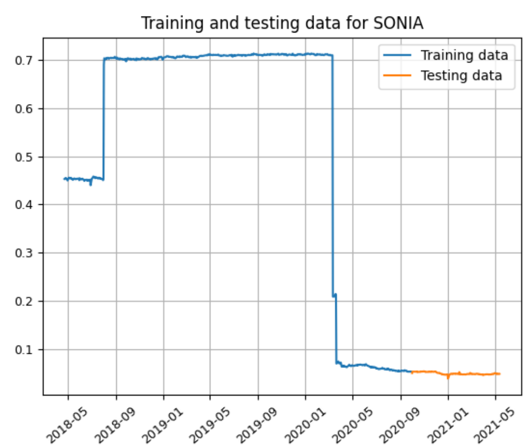
We defined the research goal of this thesis to be to develop a Random Forest Regression algorithm and an Autoregressive Integrated Moving Average model and compare their forecasting performances on the overnight interest rates €STR, SOFR and SONIA. As described in Section 3, we set the split ratio of training and testing date for the €STR to 70 : 30 and for the SOFR and SONIA to 80 : 20. Figure 9 shows the visualization of the splits for the respective Alternative Reference Rates (ARR). There are 289 data points for the €STR for the training data and 124 data points for the testing data. The training data set for the SOFR consists of 624 data points and the testing data set of 156 data points. For the SONIA, the training and testing data sets consist of 618 and 155 data points, respectively.



(a) 70:30 split for €STR.



(b) 80:20 split for SOFR.



(c) 80:20 split for SONIA.

Figure 9: Splits of data into training and testing data for the ARRs.

Table 5 shows descriptive statistics of the training and testing data set of the respective overnight rates. There are differences in mean and variance for training and testing data set for both SOFR and SONIA. The mean of the training data for SOFR is 5.25 whereas it is 0.11 for the testing data. The variance is 0.7885 for the training data and 0.0013 for the testing data.

The training data of the SONIA has a mean of 0.5334 whereas the testing data has a mean of 0.0502. Furthermore, the variance for the SONIA training data is 0.0697 and for the testing data $6.2500e-06$. The reason for these differences is that the testing data set is taken after the downward jump of the interest rates in March 2021 due to Covid-19. The interest rates continued on a much lower level as compared to before the jump.

Statistic	Training €STR	Testing €STR	Training SOFR	Testing SOFR	Training SONIA	Testing SONIA
Number of data points	289	125	624	156	618	155
Minimum (in %)	-0.57	-0.58	0.01	0.01	0.05	0.04
Maximum (in %)	-0.51	-0.55	5.25	0.11	0.71	0.06
Mean (in %)	-0.55	-0.56	1.62	0.05	0.53	0.05
Variance (in %)	6.74E-5	2.10E-5	0.78	1.3E-3	6.97E-2	6.25E-6
Standard Deviation (in %)	0.0082	0.0046	0.8824	0.0354	0.2641	0.0025

Table 5: Summary statistics for training and testing data set.

In the following subsections, we will first present and discuss the calibration and prediction of the ARIMA models and the Random Forest models per interest rate including walk-forward validation. This addresses the first research question on the effectiveness of a Random Forest Regression algorithm in predicting €STR, SOFR and SONIA based on historical values as compared to ARIMA. Subsequently, we will present the calibration and predictions of ARIMAX and Random Forest including various input features. This part addresses the second research question on the forecasting effectiveness of ARIMAX and Random Forest using these input features.

4.1 ARIMA calibration

The first step in the analysis is to calibrate an ARIMA(p,d,q) model for each of the interest rates €STR, SOFR and SONIA. First, we need to determine the values for the orders p, d, and q. We test the order of differencing d by fitting different ARIMA(0,d,0) models to the training data interest rates and assessing the optimal order of differencing according to the standard deviation of the residuals. The residuals are the difference between the interest rate values as calculated by the fitted ARIMA model and the actual interest rate values. Table 6 summarizes the standard deviations of the residuals for various ARIMA(0,d,0) models for €STR, SOFR and SONIA. The ARIMA(0,1,0) models yields the smallest standard deviation for all three interest rates. Therefore, we choose $d = 1$ for €STR, SONIA and SOFR.

Model	€STR	SOFR	SONIA
ARIMA(0,1,0)	0.0326	0.1893	0.0294
ARIMA(0,2,0)	0.0368	0.2939	0.0383
ARIMA(0,3,0)	0.0478	0.5116	0.0618
ARIMA(0,4,0)	0.0652	0.9322	0.1078

Table 6: Standard deviations for different ARIMA(0,d,0) models for €STR, SONIA and SOFR.

Next, we need to find orders p and q for the ARIMA(p,1,q) models by considering the autocorrelation and partial autocorrelation plots for the training data as shown in Figures 10, 11, and 12 and considering the significance of the correlations. In all plots, the correlation of the series with itself when no lag is applied is always equal to 1. The blue shaded areas illustrate whether the correlations are significant. Points outside of this area are statistically significant correlations. The partial autocorrelation plots for €STR, SONIA and SOFR all show positive partial autocorrelations at lag 1. Therefore, we set $q = 0$ for all three rates and continue further with assessing the order p in the ARIMA(p,1,0) models [47]. In Figure 10 we see the autocorrelation and partial autocorrelation for the training data of €STR. The autocorrelations up until a lag of 22 are significant. The partial autocorrelations are significant until a lag of 5. Therefore, we start with an ARIMA(5,1,0) for the €STR.

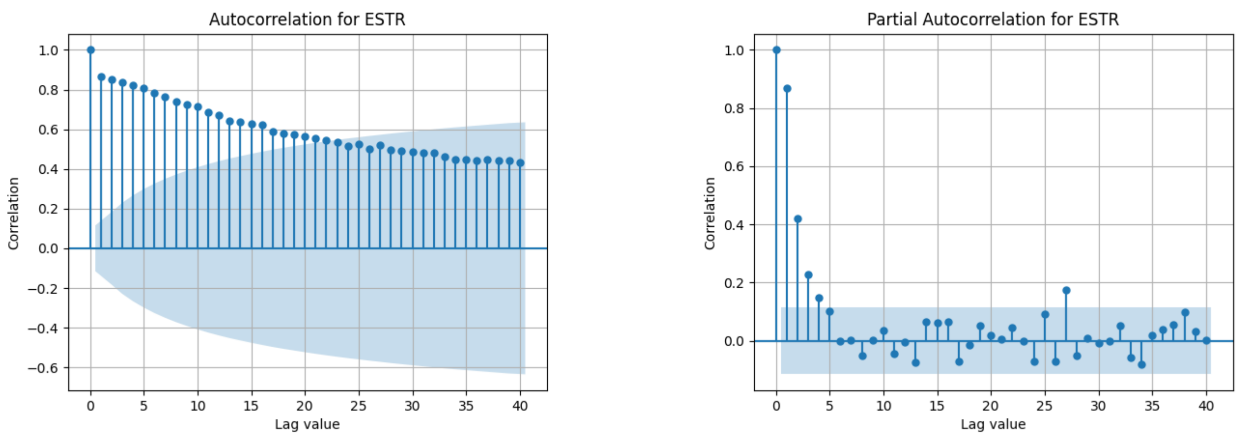


Figure 10: Autocorrelation (left) and Partial Autocorrelation (right) for the €STR.

Table 7 shows different ARIMA(p,1,0) models fitted to the training data set of the €STR and the standard deviations of the residuals. ARIMA(5,1,0) and ARIMA(4,1,0) have the lowest standard deviations. Since the parameter of the fifth coefficient in an ARIMA(5,1,0) is not statistically significant ($p = 0.735$), we choose to use ARIMA(4,1,0) for fitting and predicting the €STR.

Model	Mean	Standard deviation
ARIMA(5,1,0)	-0.001954	0.03248
ARIMA(4,1,0)	-0.001953	0.03248
ARIMA(3,1,0)	-0.001948	0.032484
ARIMA(2,1,0)	-0.001943	0.032490

Table 7: Different ARIMA(p,1,0) models and the corresponding standard deviations of the residuals for €STR.

Figure 11 shows the autocorrelation and partial autocorrelation for the SOFR. The autocorrelation is significant until a lag of 54 and the partial autocorrelation until a lag of 6. We choose the ARIMA(p,1,0) model for SOFR in the same way as for the €STR.

Table 8 shows that the standard deviation is lowest for ARIMA(6,1,0).

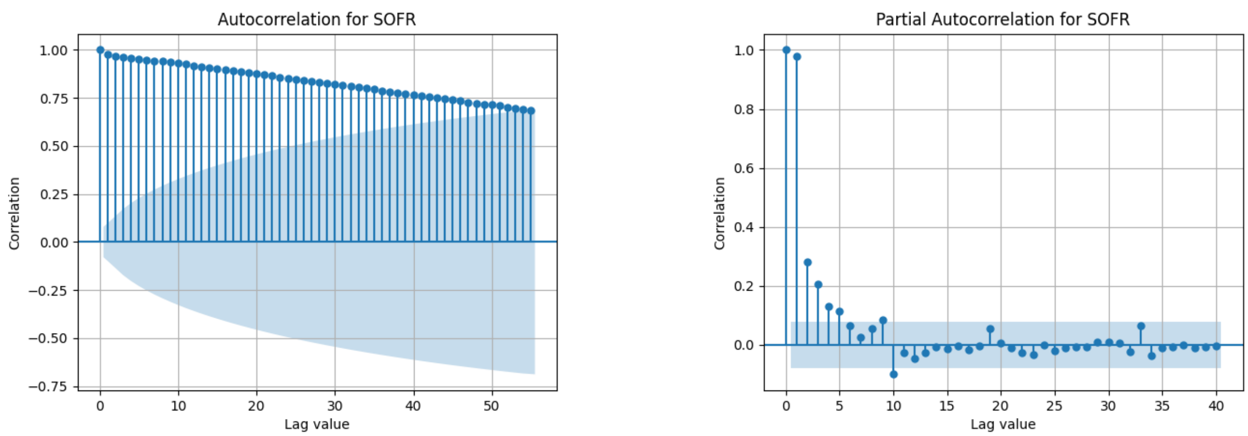


Figure 11: Autocorrelation (left) and Partial Autocorrelation (right) for the SOFR.

Model	Mean	Standard deviation
ARIMA(6,1,0)	-0.003650	0.173852
ARIMA(5,1,0)	-0.003399	0.173977
ARIMA(2,1,0)	-0.001548	0.177624

Table 8: Different ARIMA(p,d,q) models and the corresponding standard deviations of the residuals for SOFR.

Figure 12 shows the autocorrelation and partial autocorrelation plots for SONIA. The autocorrelation is significant until a lag of 46 and the partial autocorrelation until a lag of 2. Table 9 shows that the standard deviation is lower for ARIMA(2,1,0) as compared to ARIMA(1,1,0). Therefore, we choose ARIMA(2,1,0) as a model for the SONIA.

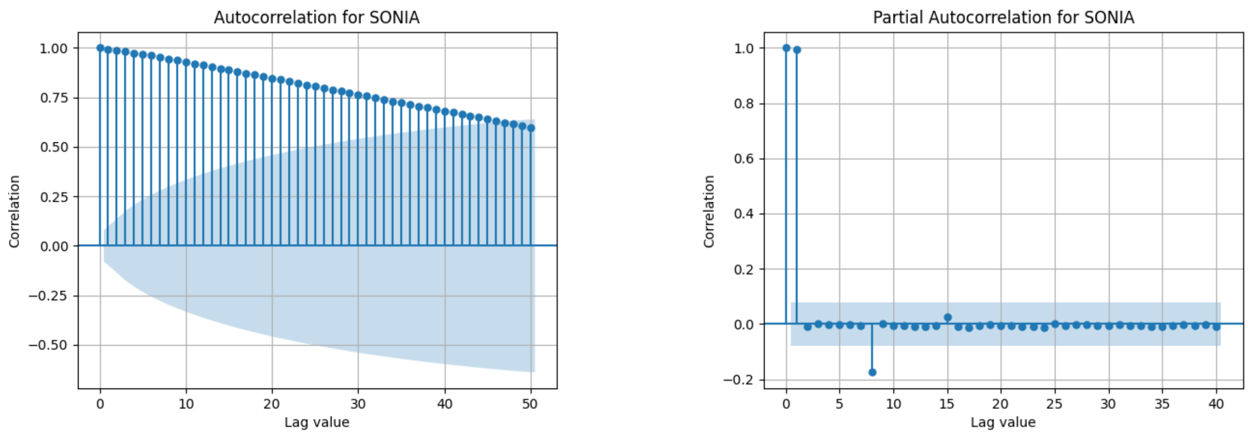


Figure 12: Autocorrelation (left) and Partial Autocorrelation (right) for the SONIA.

Model	Mean	Standard deviation
ARIMA(2,1,0)	0.000089	0.029394
ARIMA(1,1,0)	0.000092	0.029395

Table 9: Different ARIMA(p,1,0) models and the corresponding standard deviations of the residuals for SONIA.

We have established that we will use ARIMA(4,1,0) for €STR prediction, ARIMA(6,1,0) for SOFR prediction and ARIMA(2,1,0) for SONIA prediction. Next, we will present the calibration of the Random Forest model. Table 4.1 shows the coefficients of the ARIMA models for €STR, SOFR and SONIA, respectively.

Argument	€STR	SOFR	SONIA
1	-0.6418	-0.4442	0.0093
2	-0.3853	-0.3489	-0.0048
3	-0.2605	-0.2444	
4	-0.1350	-0.1857	
5		-0.1019	
6		-0.0404	

Table 10: Coefficients of ARIMA (4, 1, 0) for €STR, ARIMA(6,1,0) for SOFR and ARIMA(2,1,0) for SONIA.

4.2 Random Forest calibration

To calibrate the Random Forest model without input features, we first transformed the data sets of the overnight rates into supervised learning data sets by creating an input column that consists of a lagged version of the time series. We further ran the randomized parameter grid search to find an improved set of parameters for this model. Table 11 shows the parameters settings that yielded the most accurate mae and mse optimal parameter settings using 500 iterations in the randomized search function.

Argument	€STR	SOFR	SONIA
n_estimators	209	1213	446
criterion	mae	mae	mae
max_depth	13	17	95
min_samples_split	5	4	6
min_samples_leaf	5	4	3
bootstrap	True	False	True

Table 11: Parameters as suggested by randomized search with 500 iterations.

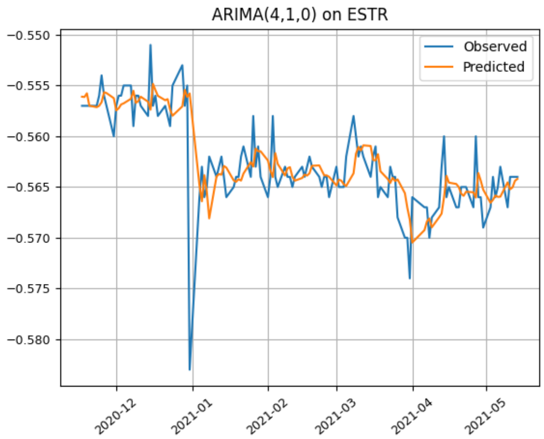
4.3 €STR prediction

We used an ARIMA(4,1,0) model and a Random Forest with two different sets of parameter settings to make a one-step prediction of the €STR. Table 12 shows the mae, mse and rmse as measured for the €STR prediction. The values are reported in basis points.

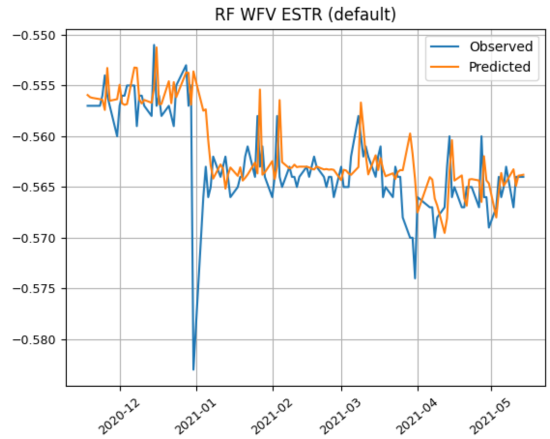
Measure	ARIMA	Random Forest (default)	Random Forest (randomly searched)
Mae	19.9	252	240
Mse	0.11	0.12	1.6
Rmse	0.34	0.34	1.27

Table 12: Mae, mse and rmse for €STR prediction using ARIMA and Random Forest models based on historical values.

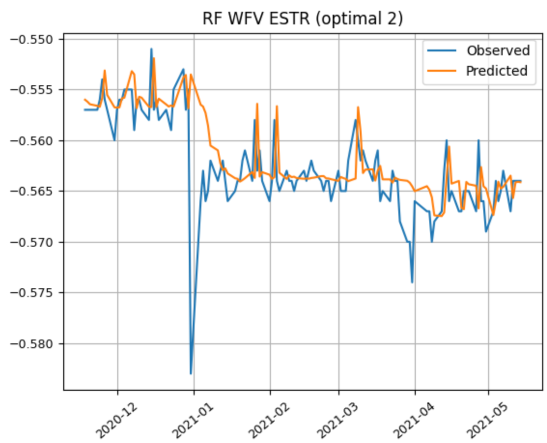
The mse is lowest for €STR prediction using ARIMA, followed by the Random Forest with default settings. The mse is highest for €STR prediction using the Random Forest with parameter settings as searched by a random grid. The mae for €STR prediction is lowest with ARIMA and is highest for Random Forest with default settings. Figure 13 shows the observed and predicted €STR values for the three different models.



(a) ARIMA



(b) Random Forest with default settings



(c) Random Forest with randomly searched settings

Figure 13: Observed and predicted values for €STR using walk-forward validation.

The ARIMA predicts the general trend of the rate well but does not capture the spikes accurately. The Random Forests capture the general trend less well but the spikes more accurately as compared to the ARIMA. None of the models capture the downward spike at the end of December 2020 or March 2021. The predicted values of the Random Forests capture the bigger spikes whereas it is more flat around the smaller spikes. The ARIMA does not capture the big spikes but is also not flat around the low spikes. This could be an explanation for why the mse and mae are lower for ARIMA as compared to the Random Forests.

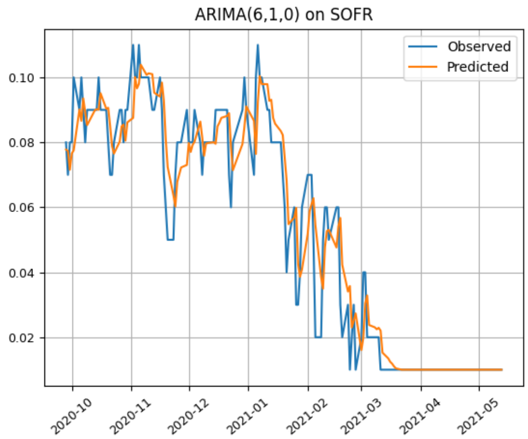
4.4 SOFR prediction

Next, we used an ARIMA(6,1,0) and a Random Forest with two different sets of parameter settings to make a one-step prediction of the SOFR. Table 13 shows the mae, mse and rmse as measured for the SOFR forecast. The values are reported in basis points.

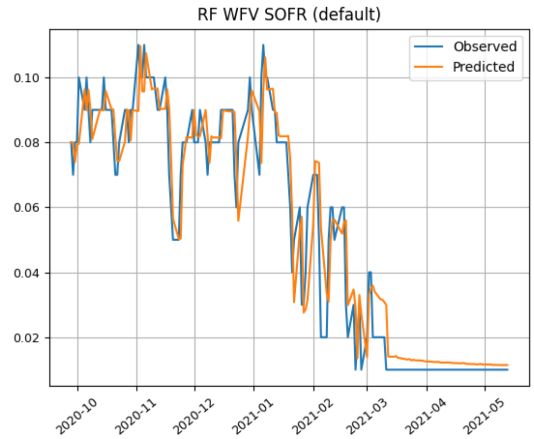
Measure	ARIMA	Random Forest (default)	Random Forest (randomly searched)
Mae	68.8	72.1	67
Mse	1.07	1.02	1.08
Rmse	1.04	1.01	1.04

Table 13: Mae, mse and rmse for SOFR prediction using ARIMA and Random Forest models based on historical values.

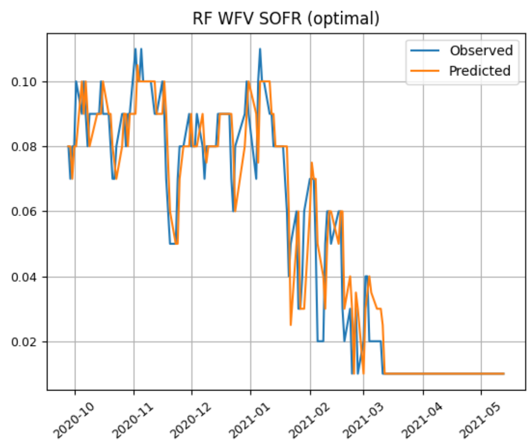
The mse is lowest for the SOFR using the Random Forest with default settings and it is highest using the Random Forest with randomly searched parameter settings. The mae is lowest using the Random Forest with randomly searched settings and highest using the Random Forest with default settings. The values for the ARIMA are in the middle for both mse and mae. Figure 14 shows the comparisons of the predictions of the different models.



(a) ARIMA



(b) Random Forest with default settings.



(c) Random Forest with settings as randomly searched by the program.

Figure 14: Observed and predicted values for SOFR using walk-forward validation.

All models capture the general trend of the SOFR whereas the ARIMA captures the spikes less accurately as compared to the Random Forests. The flatness of the SOFR since the beginning of March 2021 is captured well by ARIMA and Random Forest with randomly searched parameter settings. However, it is not captured well by Random Forest with default parameter settings.

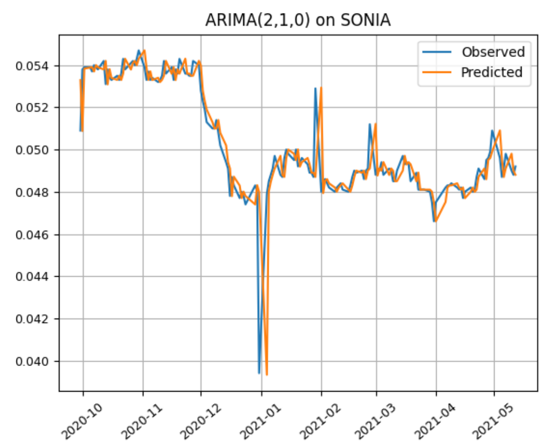
4.5 SONIA prediction

We used ARIMA(2,1,0) and a Random Forest with two sets of parameter settings to make one-step predictions of the SONIA. Table 17 shows the mae, mse and rmse as measured when forecasting the SONIA. The values are reported in basis points.

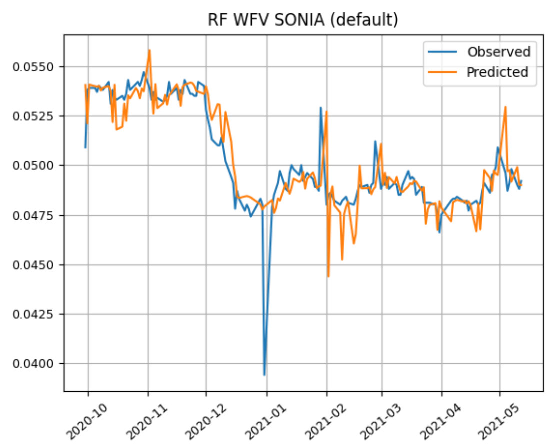
Measure	ARIMA	Random Forest (default)	Random Forest (randomly searched)
Mae	5.99	8.02	6.22
Mse	0.02	0.02	0.01
Rmse	0.13	0.13	0.11

Table 14: Mae, mse and rmse for SONIA prediction using ARIMA and Random Forest models based on historical values.

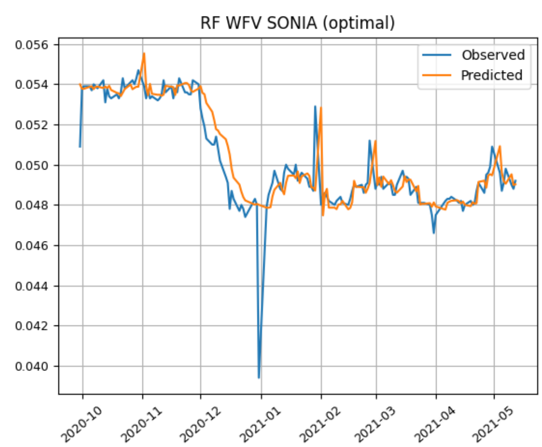
The mse for SONIA prediction is lowest for Random Forest with randomly searched parameter settings, followed by ARIMA. The mae for SONIA prediction is lowest for ARIMA and highest for Random Forest with default settings. Figure 15 shows the comparison of predicted and observed values of the SONIA using the three different models.



(a) ARIMA



(b) Random Forest with default settings



(c) Random Forest with randomly searched settings

Figure 15: Observed and predicted values for SONIA using walk-forward validation.

The ARIMA model is the only model that captured the downward spike at the end of December 2020. However, the spike has some delay in the predicted values. The Random Forest with default settings predicts spikes - mostly downward - that are not observable in the actual values. The general trends are captured well by all models.

4.6 Conclusion

In this section, we presented the results for the first research question. We showed the accuracy of €STR, SOFR and SONIA prediction by means of rmse, mse and mae for different ARIMA and Random Forest calibrations. Models follow the general trends of the interest rates generally well whereas the spiking is captured less well with ARIMA in SOFR prediction as compared to Random Forest. The spiking is captured better with ARIMA for SONIA prediction than with Random Forest. The prediction for €STR is in general less accurate with all models as compared to SOFR and SONIA prediction. One reason for that might be that there are less data points available for the calibration of the models for €STR forecasting.

In Section 5, we will present the results for the second research question. In Section 7, we will discuss the findings.

5 Prediction based on market and mathematical features

The second research question of this thesis focuses on forecasting €STR, SOFR and SONIA with ARIMAX and Random Forest using several input features. In this section, we will first describe the input features that we used followed by the presentation of the prediction accuracy and plots.

5.1 Input features and calibration

The input features for the interest rates can be divided into market features and mathematical features. The market feature for the €STR are the number of active banks, total volume, share of volume of the five largest banks using €STR, number of transactions, 25^{th} and 75^{th} percentiles. Figure 16 shows a visualization of these market features.

Figure 16: Market features on €STR.

We see that the number of active banks, share of volume, number of transactions and total volume fluctuate with the market. There are some clear downward spikes around January 2020, August 2020 and January 2021 for these features. These are the same times that the €STR experienced downward spikes. Furthermore, we can see that the 25^{th} and 75^{th} percentiles move with the €STR.

We also calculated mathematical input features for each interest rate. The autocorrelations for the €STR are significant until a lag of 22 whereas the partial autocorrelation is significant until a lag of 5 as established above. Therefore, we calculate the Relative Strength Index and the Stochastic Oscillator for the €STR with $n = 5$ and $n = 22$.

Figure 17 shows a visualization of the input features for the €STR.

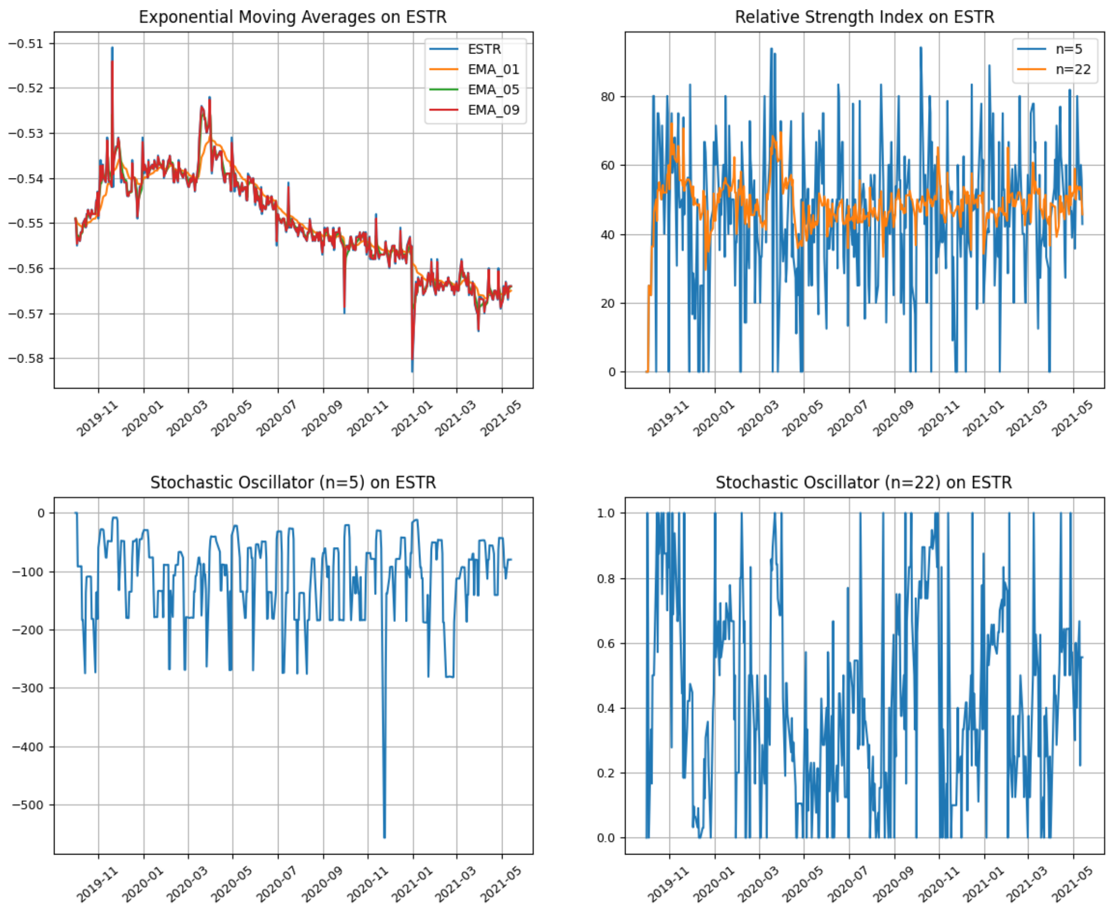


Figure 17: Input features on €STR.

The exponential moving average with $\alpha = 0.9$ follows the €STR most closely whereas the exponential moving average with $\alpha = 0.1$ only follows the general trend but less the spiking of the rate. The Relative Strength Index with large n fluctuates less much as compared to the Relative Strength Index with small n. A similar pattern holds true for the Stochastic Oscillator.

The market features that we used for the SOFR are the 1^{st}, 25^{th}, 27^{th} and 99^{th} percentiles as well as the Tri-Party General Collateral Rate (TGCR) and Broad General Collateral Rate (BGCR). Figure 18 shows a visualisation of these market features. The unsecured rates TGCR and BGCR are very similar to the SOFR. However, they are not completely the same.

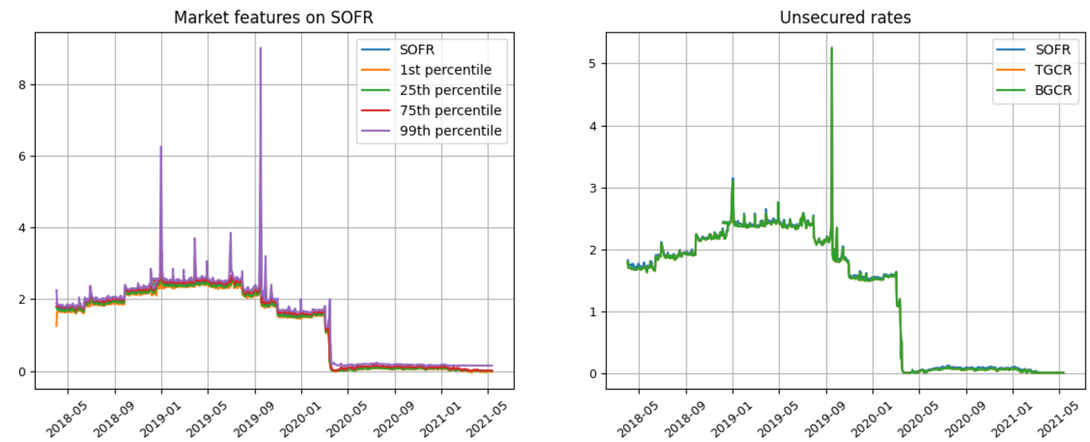


Figure 18: Market features on SOFR.

The autocorrelation function of the SOFR is significant until a lag of 54 whereas the partial autocorrelation function of the SOFR is significant until a lag of 6. Therefore, we chose $n = 6$ and $n = 54$ for the calculations of the Relative Strength Index and the Stochastic Oscillator. Figure 19 visualizes the RSI, SO and exponential moving average.

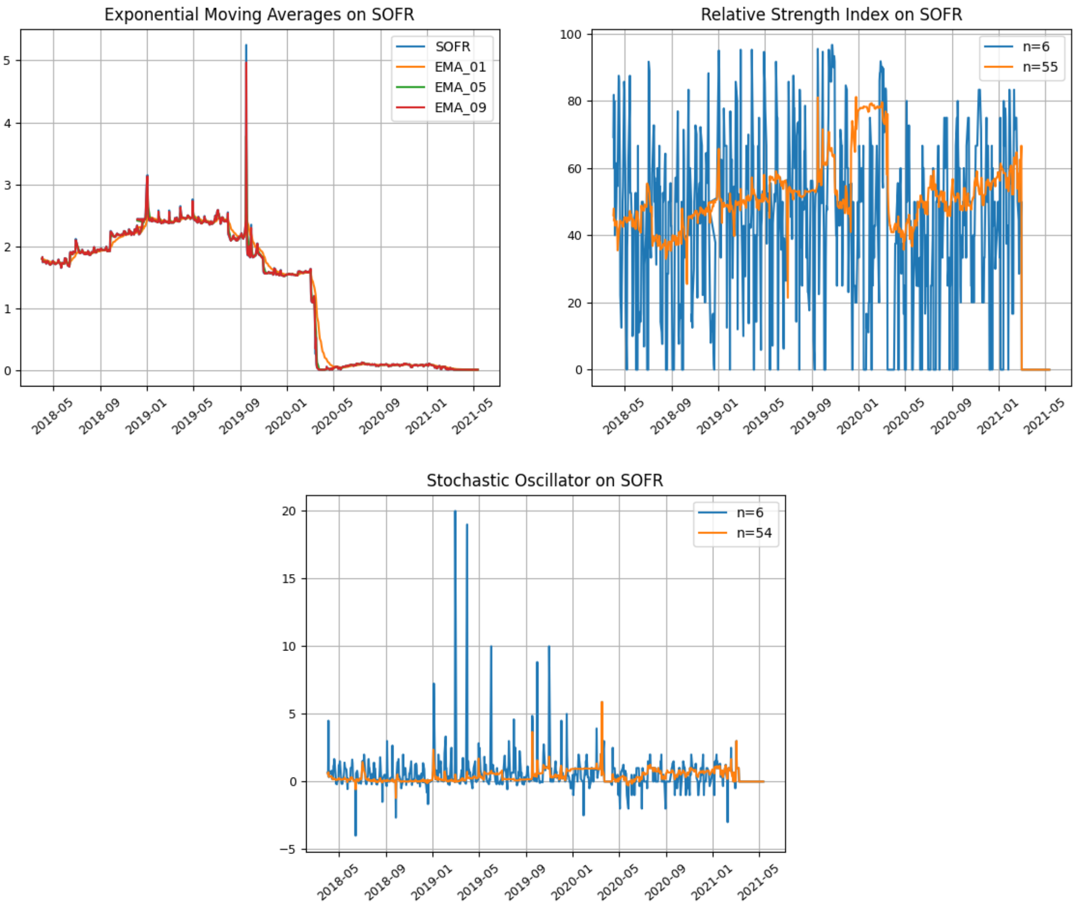


Figure 19: Mathematical features on SOFR.

We only include the mathematical indicators for the SONIA. The autocorrelation function is significant until a lag of 46 and the partial autocorrelation function until a lag of 2. Therefore, we use $n = 2$ and $n = 46$ for feature calculation for the SONIA. Figure 20 shows a visualization of the features for the SONIA.

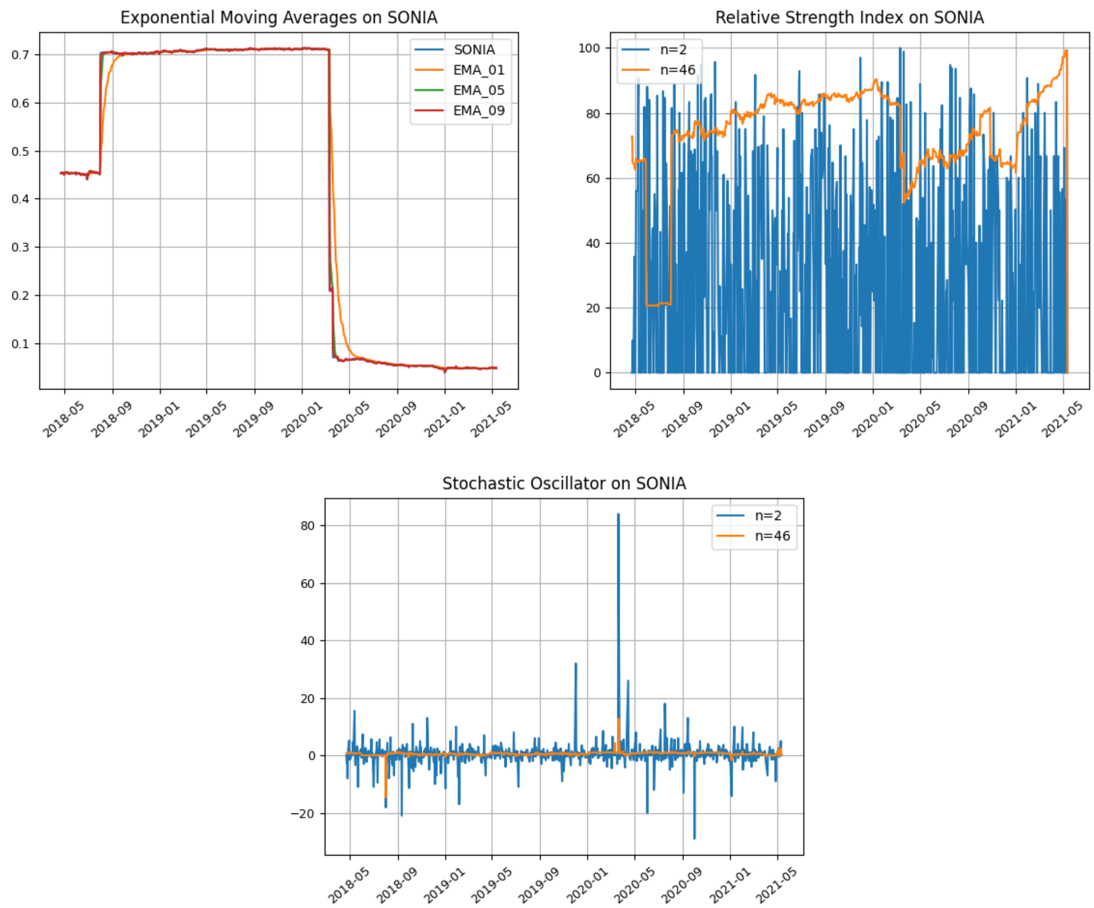


Figure 20: Input features on SONIA.

We first ran the ARIMAX including all input features that are mentioned above and viewed the summary of the model. Only the three different exponential moving averages and the stochastic oscillator with $n = 22$ For the €STR were significant features. The three exponential moving averages, 25^{th} percentile, TGCR, BGCR and stochastic oscillator with $n = 54$ were significant for the SOFR. The exponential moving averages, relative strength index with $n = 46$ and stochastic oscillator with $n = 46$ were significant for the SONIA. We removed the input features that were not significant from the ARIMA models to test whether the performance increases. Furthermore, we assessed the feature importance of the Random Forest. Figures 15, 16 and 17 show the relative contributions of the features in percent for €STR, SOFR and SONIA, respectively.

Feature	Importance
EMA ($\alpha = 0.9$)	96.85%
EMA ($\alpha = 0.5$)	0.73%
RSI $n = 22$	0.66%
SONIA	0.62%
75^{th} percentile	0.38%
Share of volume	0.2%
SO $n = 22$	0.15%
SO $n = 5$	0.13%
25^{th} percentile	0.09%
RSI $n = 5$	0.08%
EMA ($\alpha = 0.1$)	0.05%
Number of transactions	0.04%
Total volume	0.01%
Number of active banks	0.01%

Table 15: Feature importance Random Forest €STR.

The feature that contributed most to €STR prediction in the Random Forest with default settings is the exponential moving average with $\alpha = 0.9$. All other features contribute by less than 1% to the model. However, we will not take these features out. It is not computationally expensive to run the model including all input features. If we remove the features that have a lower relative contribution, then the model performs slightly less accurately while only saving slightly more time in computation.

Feature	Importance
EMA ($\alpha = 0.1$)	16.44%
TGCR	15.25%
EMA ($\alpha = 0.9$)	14.58%
1^{st} percentile	13.41%
25^{th} percentile	11.41%
75^{th} percentile	11.08%
BGCR	7.18%
EMA ($\alpha = 0.5$)	6.32%
99^{th} percentile	3.7%
RSI $n = 54$	0.46%
RSI $n = 6$	0.17%
SO $n = 54$	< 0.01%
SO $n = 6$	< 0.01%

Table 16: Feature importance Random Forest SOFR.

The contribution of the features is more evenly distributed for the SOFR as for the €STR. The exponential moving averages with $\alpha = 0.1$ and $\alpha = 0.9$, TGCR and 1^{st}, 25^{th} and 75^{th} percentile have the highest contribution to predictions whereas the relative strength indices and stochastic oscillators have the lowest contribution. Similarly to the features of the €STR, we do not exclude any features from further prediction.

Feature	Importance
EMA ($\alpha = 0.9$)	68.68%
EMA ($\alpha = 0.5$)	22.33%
EMA ($\alpha = 0.1$)	8.78%
RSI $n = 46$	0.18%
SO $n = 46$	0.011%
SO $n = 2$	0.01%
RSI $n = 2$	$< 0.01\%$

Table 17: Feature importance Random Forest SONIA.

The exponential moving average with $\alpha = 0.9$ has the highest contribution to SONIA prediction, followed by the exponential moving average with $\alpha = 0.5$ and $\alpha = 0.1$. The lowest contributions have the stochastic oscillator and relative strength index with $n = 2$. We do not exclude any features for further prediction. Furthermore, we ran a randomized parameter search algorithm on the Random Forest including input features similarly to the Random Forest based solely on historical values of the three interest rates. Table 18 shows the parameters as suggested by the random search function with 500 iterations.

Argument	€STR	SOFR	SONIA
n_estimators	819	1276	18
criterion	mse	mae	mae
max_depth	69	70	90
min_samples_split	5	5	5
min_samples_leaf	1	2	5
bootstrap	True	True	False

Table 18: Parameters as suggested by randomized search with 500 iterations.

5.2 €STR prediction

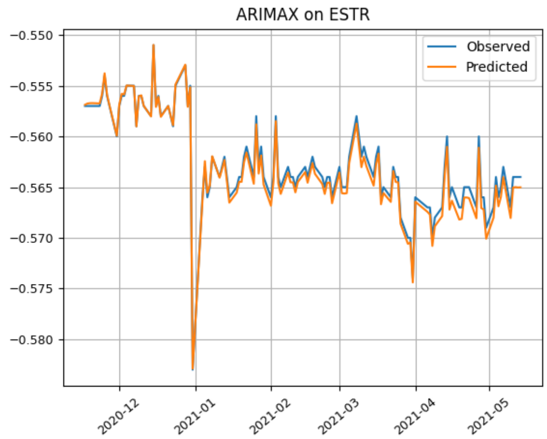
We used ARIMAX with all input features, ARIMAX removing insignificant input features, Random Forest with all input features and default parameter settings and Random Forest with all input features and parameter settings as suggested by a random search for prediction of the €STR. Table 19 shows the mae, mse and rmse as measured for one-step €STR prediction with these models. Values are reported in basis points.

Measure	ARIMAX	ARIMAX (excl. features)	Random Forest (default)	Random Forest (randomly searched)
Mae	5.54	5.48	16.9	18.9
Mse	0.0043	0.0042	0.0685	0.0759
Rmse	0.0655	0.0648	0.2617	0.2755

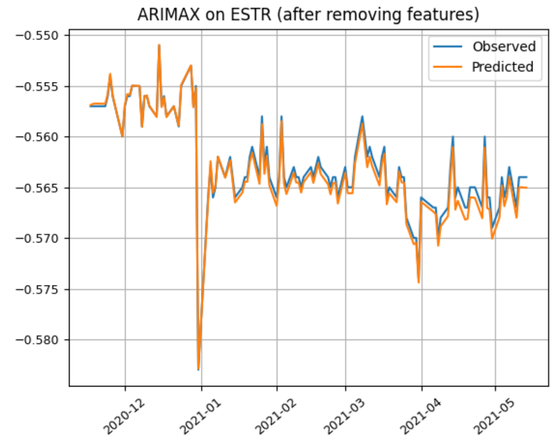
Table 19: Comparison of mae, mse and rmse for €STR predictions using ARIMAX and Random Forest.

The mae is lowest for ARIMAX excluding insignificant input features and amounts to 5.48 bps, closely followed by ARIMAX model including all input features with 5.54 bps. The mae is higher for both Random Forest models with 16.9 bps and 18.9bps. The mse and subsequently the rmse are also lower for both ARIMAX models as compared to the Random Forest models. All in all, ARIMAX without input features has lowest mae and rmse for the €STR prediction.

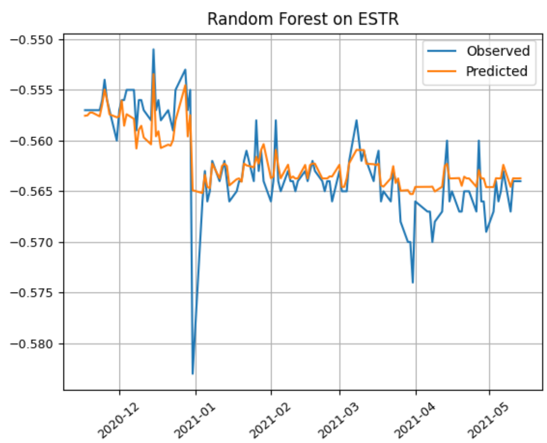
Figure 21 shows the observed and predicted values for €STR prediction using the four different models.



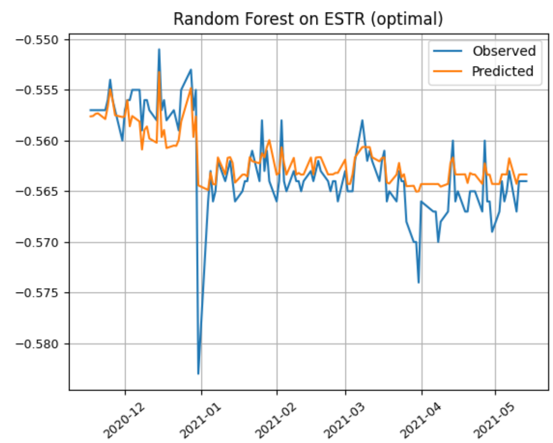
(a) ARIMAX including all input features.



(b) ARIMAX removing insignificant input features.



(c) Random Forest (default).



(d) Random Forest (randomly searched).

Figure 21: Observed and predicted values €STR.

It can be observed that the predicted values of both ARIMAX models indeed follow the observed process more closely as compared to Random Forest. The Random Forest does follow the general trend of the interest rate well but fails to capture the spikes accurately.

5.3 SOFR prediction

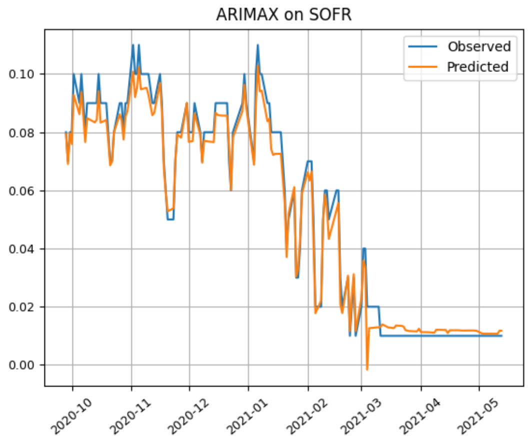
We used ARIMAX with all input features, ARIMAX removing insignificant input features, Random Forest with all input features and default parameter settings and Random Forest with all input features and parameter settings as suggested by a random search for prediction of the SOFR. Table 20 shows the mae, mse and rmse as measured for one-step SOFR prediction with these models. Values are reported in basis points.

Measure	ARIMAX	ARIMAX (excl. features)	Random Forest (default)	Random Forest (randomly searched)
Mae	34.6	47.1	9.25	24.1
Mse	0.19	0.29	0.03	0.1
Rmse	0.44	0.54	0.18	0.32

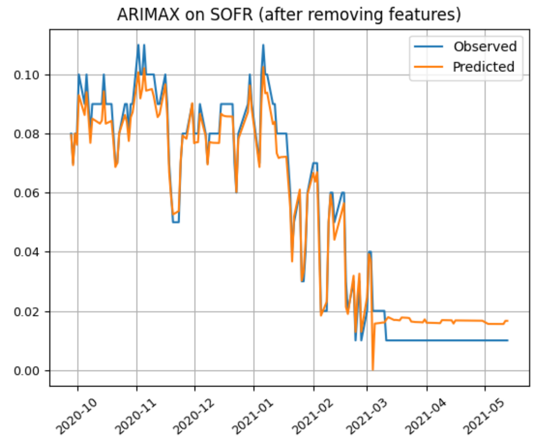
Table 20: Comparison of mae, mse and rmse for SOFR predictions using ARIMAX and Random Forest.

Both Random Forests outperform the ARIMAX models in terms of mae. The Random Forest with default parameter settings yields a mae of 9.25 bps whereas the Random Forest with randomly searched parameter settings yields a mae of 24.1 bps. The ARIMAX models have a mae of 34.6 bps and 47.1 bps. The standard deviation as shown by the rmse are also lower for the Random Forest models as compared to ARIMAX.

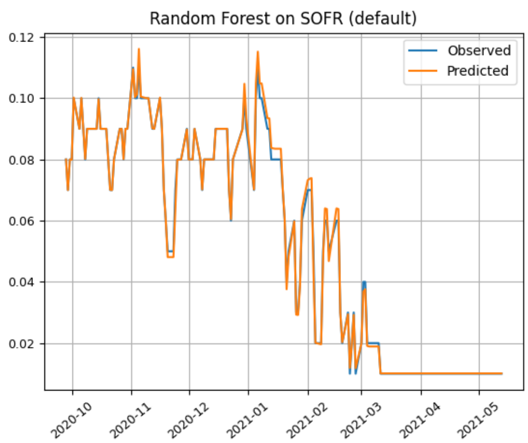
Figure 22 shows the observed and predicted values for SOFR prediction using the four different models.



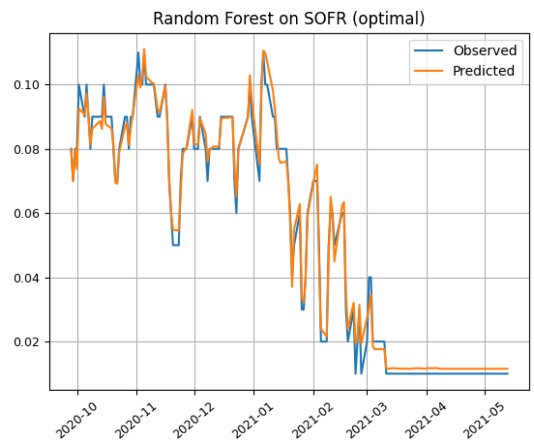
(a) ARIMAX including all input features



(b) ARIMAX after removing features



(c) Random Forest (default)



(d) Random Forest (randomly searched)

Figure 22: Observed and predicted values SOFR.

It can be observed that the predicted SOFR values using the Random Forest with default parameter settings follow the observed process most accurately. This is inline with the reported mae and rmse. As previously, all models follow the general trend of the SOFR well. Furthermore, all models register the spiking of the SOFR. The flatness of the SOFR since beginning of March 2021 is not captured well by both ARIMAX models. It is, however, captured by the Random Forest with default parameter settings.

5.4 SONIA prediction

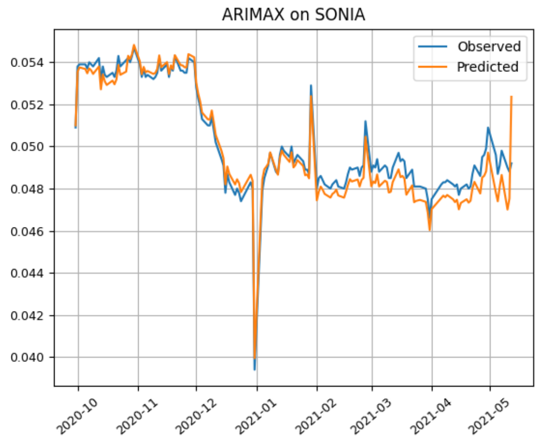
We used ARIMAX with all input features, ARIMAX removing insignificant input features, Random Forest with all input features and default parameter settings and Random Forest with all input features and parameter settings as suggested by a random search for prediction of the SONIA. Table 21 shows the mae, mse and rmse as measured for one-step SONIA prediction with these models. Values are reported in basis points.

Measure	ARIMAX	ARIMAX (excl. features)	Random Forest (default)	Random Forest (randomly searched)
Mae	4.91	4.92	35.5	39.7
Mse	0.0038	0.0039	0.183	0.217
Rmse	0.062	0.0621	0.4278	0.4658

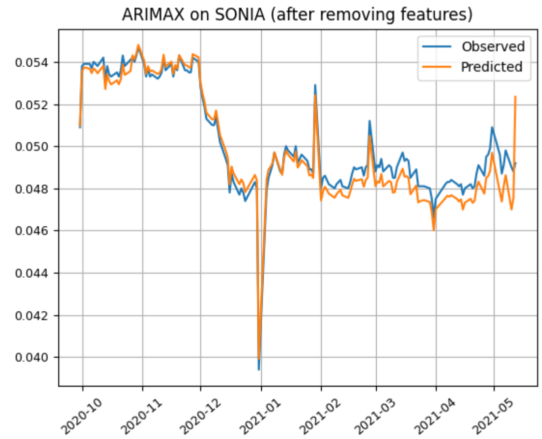
Table 21: Comparison of mae, mse and rmse for SONIA predictions using ARIMAX and Random Forest.

The mae is lower for one-step SONIA prediction using both ARIMAX models as compared to the Random Forests. The ARIMAX models yield a mae of 4.91 bps and 4.92 bps. The Random Forests yield a mae of 35.5 bps and 39.7 bps. The rmse is higher for both Random Forest models as compared to the ARIMAX models as well. The error measures for the ARIMAX model with all input features and for the model excluding insignificant features are very close to each other.

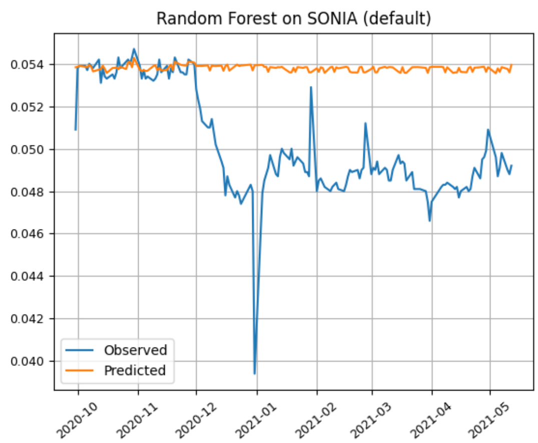
Figure 23 shows the observed and predicted values for SONIA prediction using the four different models.



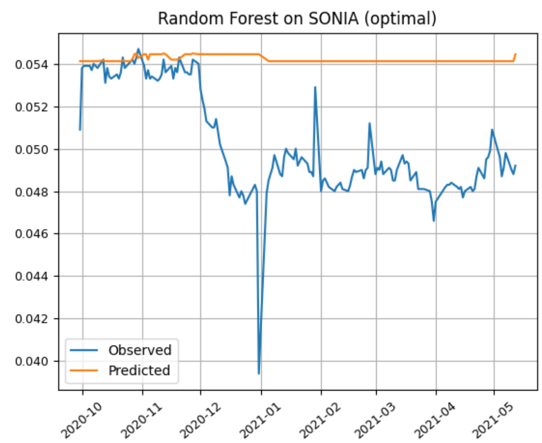
(a) ARIMAX with all input features.



(b) ARIMAX after removing input features.



(c) Random Forest (default).



(d) Random Forest (randomly searched).

Figure 23: Observed and predicted values SONIA

The ARIMAX models capture the general trend of the SONIA well and register the downward spike at the end of December 2020. After around February 2021, both models capture the trend and the spikes of the overnight rate well but have an observable offset. The Random Forests fail to capture the trend or the spikes of the SONIA. The prediction is rather flat with some small fluctuations. The predicted values do not seem to follow the observed process.

By experimenting, we found that decreasing the testing data set to from 20% to 10% of the most recent SONIA values results in a Random Forest that can capture the trends and spiking for the predictions. We ran the random parameter searching function with 500 iterations again for this model. Table 22 shows the resulting randomly searched parameter grid.

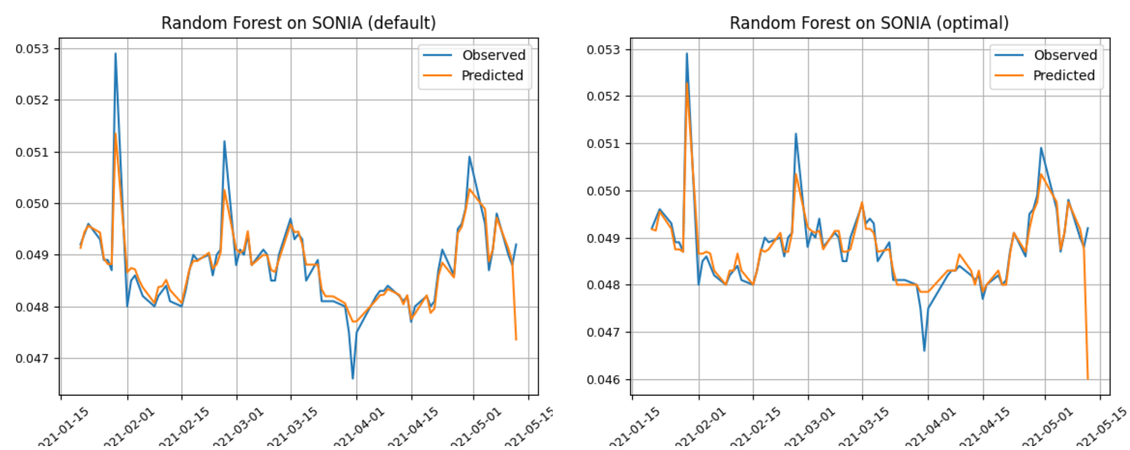
Argument	SONIA
n_estimators	1086
criterion	mse
max_depth	87
min_samples_split	3
min_samples_leaf	1
bootstrap	False

Table 22: Parameters as suggested by randomized search with 500 iterations for SONIA with a training to testing data set split of 90 : 10.

Figure 23 shows the mae, mse and rmse for the Random Forest with default parameter settings and with randomly searched settings on SONIA prediction with a training to testing data set ratio of 90 : 10. Values are reported in basis points.

Measure	Random Forest (default)	Random Forest (randomly searched)
Mae	1.86	0.002
Mse	0.0013	2.05
Rmse	0.04	1.43

Table 23: Comparison of mae, mse and rmse for SONIA predictions using ARIMAX and Random Forest.



(a) SONIA prediction with Random Forest with default parameter settings with a testing data set of 10%.

(b) SONIA prediction with Random Forest with randomly searched parameter settings with a testing data set of 10%.

Figure 24: Observed and predicted values SONIA

5.5 Conclusion

In this section, we presented the results for the second research question. We made one-step predictions of €STR, SOFR and SONIA using two different ARIMAX models and two different Random Forest models. The €STR and SONIA are more accurately predicted using ARIMAX as compared to Random Forest. On the other hand, SOFR is more accurately predicted using Random Forest as compared to ARIMAX.

Furthermore, predictions are more accurate in terms of mae, mse and rmse for the models including input features than for the predictions based solely on historical values as discussed in section 4. The exception to this statement is the Random Forest with input features used to predict SONIA. The models fail to capture the trends or spiking of SONIA in general.

In Section 6, we will discuss the significance of the one-step predictions in comparison to simply using the value of the previous day of the interest rates. Section 7 contains the conclusion and discussion of the research that we conducted including recommendations for theory and practice.

6 Comparison to a simple model

We developed a Random Forest algorithm to make one-step predictions of the overnight interest rates €STR, SOFR and SONIA. We further applied ARIMA and ARIMAX models to make the same predictions and compared the results to the Random Forest models.

We were able to develop multiple models that can accurately make a one-step prediction for the overnight interest rates. We have seen that the models are in general more accurate when input features are incorporated as compared to models without input features.

To put the results into context, we also calculated the mae, mse and rmse of a simple prediction model which simply takes the value of the previous business day as prediction value. That is, if the value of the €STR was -0.565 yesterday, then the model predicts that the value of the €STR today is -0.565. Table 24 shows the mae, mse and rmse for prediction using this simple model based on the previous business day. Values are reported in basis points and are calculated using only the test data set. This data set consists of the most recent 30% of data points for €STR and 20% for SOFR and SONIA.

Rate	Mae	Mse	Rmse
€STR	22.86	0.1	0.32
SOFR	62.42	0.6	0.78
SONIA	5.99	0.01	0.11

Table 24: Comparison of mae, mse and rmse for €STR, SOFR and SONIA using a simple prediction model.

None of the models as presented in Section 4 which predict based on historical values yield lower error measures for SOFR or SONIA. The ARIMA model based on historical values yields a lower mae for €STR as compared to the simple model but not in terms of mse or rmse.

The one-step predictions as presented in Section 5 are made with ARIMAX and Random Forest models which use multiple input features for model fitting. All models including input features yield lower mae, mse and rmse for €STR and SOFR as compared to the simple prediction model.

The least accurate of the prediction models including input features for €STR prediction is Random Forest with randomly selected parameter settings with an mae of 18.9 bps and rmse of 0.28 bps. Hence, the mae is lower by 3.96 bps and rmse is lower by 0.04 bps. The most accurate prediction model based on input features for €STR is the ARIMAX excluding insignificant features with an mae of 5.48 bps and rmse of 0.06 bps. This is a decrease in mae by 17.38 bps and in rmse by 0.26 bps.

Similarly, the mae for SOFR predictions using ARIMAX excluding input features is 15.32 bps lower than the mae of the simple model. The rmse is lower by 0.24 bps. The best performing model for SOFR prediction is Random Forest with default settings which yields an mae which is 53.17 bps lower than the mae of the simple model. The rmse is 0.6 bps lower than the rmse of the simple model.

The ARIMAX models outperform the simple model with regard to SONIA prediction. The mae of ARIMAX is 1.08 bps lower than the mae of the simple model while the rmse is 0.05 bps lower than the rmse of the simple model. The Random Forests predict the SONIA less accurate than the simple model. The mae is higher by 29.51 bps for the Random Forest with default parameter settings and by 33.71 bps for the Random Forest with randomly searched parameter settings.

We can conclude from these results that we developed ARIMAX and Random Forest models which can predict €STR and SOFR more accurately than a model using a simple approach. Furthermore, we developed ARIMAX model which can predict SONIA more accurately as compared to a simple model.

In Section 7 we will discuss these results and make recommendations about overnight interest rate predictions for theory and practice.

7 Conclusion and discussion

The goal of this thesis was to develop a Random Forest Regression algorithm and compare its forecasting performance on the Alternative Reference Rates (ARRs) €STR, SOFR and SONIA to the forecasting performance of an Autoregressive Integrated Moving Average (ARIMA) model.

Firstly, we assessed how effective a Random Forest Regression algorithm is in making a one-step prediction of €STR, SOFR and SONIA based on historical values as compared to an ARIMA(p,d,q) model. To answer this question, we first calibrated a suitable ARIMA model for each interest rate based on the autocorrelation and partial autocorrelation plots. Next, we applied a Random Forest algorithm with two different sets of parameters which were based on default settings as suggested by literature and randomly selected settings from a pool of applicable parameters, respectively. We used the mean absolute error (mae), the mean squared error (mse) and the root mean squared error (rmse) to assess the forecasting accuracy of the models. The mae measures the average of the absolute difference between the observed and the predicted values. The mse measures the average squared error between observed and predicted values and can be understood as the variance of the differences. Subsequently, the rmse can be understood as the standard deviation of the residuals. Next, we present the main results for the first research question.

We compared the mae, mse and rmse for €STR, SOFR and SONIA prediction using one ARIMA model and two Random Forest models with different parameter settings each. The ARIMA outperforms the Random Forest for SONIA in terms of mse and rmse while Random Forest outperforms ARIMA for €STR and SOFR. The ARIMA yields a lower mae for €STR and SONIA than the Random Forest while the Random Forest yields a lower mae for SOFR. This result meets our expectations as previous literature suggested that the performance of forecasting models heavily depends on the data set. Our result yields additional evidence for that claim.

Furthermore, there seems to be a trade-off between decreasing mae and mse. The mae is lower for Random Forest than for ARIMA on SOFR whereas the mse is lower for ARIMA than for Random Forest. Similarly, the ARIMA yields a lower mae but a higher mse as compared to the Random Forest. The Random Forest with randomly selected parameter settings predicts all three overnight rates more accurately than the Random Forest with default parameter settings. The Random Forest with randomly selected parameter settings decreases the mae for €STR from 252 bps to 240 bps, for SOFR from 72.1 bps to 67 bps and for SONIA from 8.02 bps to 6.22 bps. This suggests that prediction performance heavily depend on the parameter settings of the Random Forest.

We assessed how effective a Random Forest Regression algorithm is in making a one-step prediction of €STR, SOFR and SONIA based on input features as compared to ARIMAX for the second research question. ARIMAX yields a lower mae, mse and rmse for €STR and SONIA as compared to Random Forest. Random Forest yields a lower mae, mse and rmse for SOFR than ARIMAX. Furthermore, the ARIMAX and Random Forest models that include input features perform make in general more accurate predictions as compared to the models solely based on historical data. We included market features such as volume or number of transactions and mathematical features such as the exponential moving average. In total, we included 14 input features for €STR, 13 for SOFR and 7 for SONIA. The mae for €STR was decreased from 19.9 bps to 5.48 bps, for SOFR from 68.8 bps to 34.6 bps and for SONIA from 5.99 bps to 4.91 bps from ARIMA to ARIMAX. The mae for €STR decreased from 240 bps to 16.9 bps and from 67 bps to 9.25 bps for SOFR from the Random Forest without input features to Random Forest with input features.

The performance of SONIA is less accurate using a Random Forest with input features as compared to without input features with an increase in mae from 6.22 bps to 35.5 bps. This result stands in contract to the rest of the predictions because the Random Forest fails to capture any characteristic of the SONIA. The Random Forest neither predicts the general trend, nor the spiking of the interest rate. One reason for that might be that the prediction performance of machine learning algorithms are heavily dependent on the characteristics of the data set. However, we found that the Random Forest with input features can predict the SONIA rate to some extent when we decrease the testing data set and increase the training data set.

Finally, we compared our results to a simple prediction model which uses the value of the overnight rates on any given day as the predicted values of the rate for the next day. All our prediction models that include input features yield lower mae, mse and rmse for €STR and SOFR than the simple prediction model. The SONIA is only predicted more accurately as compared to the simple model by both ARIMAX models.

7.1 Strengths and limitations

The main strong point of the study is that we developed a solid foundation for overnight interest rate prediction. We developed ARIMA, ARIMAX and Random Forest models that can make one-step predictions of €STR, SOFR and SONIA either solely based on historical values or based on several input features. We concluded that the ARIMAX and Random Forest predict €STR, SOFR and SONIA in general more accurately including input features as compared to excluding input features. We have further seen that the choice of parameters in a Random Forest can influence the prediction accuracy.

We have seen that the prediction accuracy of the ARIMA, ARIMAX and Random Forest heavily depend on the data set. We concluded that the respective models including input features predicts €STR, SOFR and SONIA more accurately as compared to the models based solely on historical values. However, the limitation of this study is that it cannot directly be generalized to other types of data sets. We have even seen that the ARIMAX predicts €STR and SONIA more accurately as compared to Random Forest while the Random Forest predicts SOFR more accurately. Therefore, these models have to be calibrated and tested separately on additional overnight rates as presented in Table 1. Therefore, we recommend that both ARIMAX and Random Forest are considered critically for further research on the prediction of other overnight rates.

7.2 Recommendations

ARIMA is easier to implement and less computationally expensive to execute in comparison to Random Forest. In addition, fine tuning of parameters is much slower for a Random Forest than for an ARIMA model. Finding the optimal parameter combinations for a Random Forest can potentially take a lot of time. The selection and implementation of input features is similar in ARIMAX and Random Forest. There is usually a time constraint in practical applications such as an engagement of a client to make an estimate of future interest cash flows. In these situations, it is more feasible to use ARIMAX to make predictions of €STR, SOFR and SONIA as compared to Random Forest because it saves time. However, if there is no time constraint and the focus lies more on retrieving the most accurate result as possible, then it is suitable implement a Random Forest and search for optimal parameter settings for €STR, SOFR and SONIA prediction.

Theoretically, we recommend to focus further research on finding optimal parameter settings and input features. Tuning the parameters and input features significantly changed the prediction accuracy for some of the rates. We expect that the prediction accuracy can be improved further by spending time on finding the optimal parameter settings. The randomized search with 500 iterations is not enough to make a statement about the optimal parameter combinations. Furthermore, more market and mathematical features could be considered as input features. We expect that both Random Forest and ARIMAX have the potential to have higher forecasting accuracy if the input feature are fine tuned.

We have seen that the Random Forest including input features fails to predict the SONIA rate when the training to testing data ratio is 80 : 20 percent. However, when we increase the training data set to 90%, then the Random Forest prediction captures the SONIA. Therefore, we recommend that future research includes the optimization of the training to testing data set ratio when fitting the prediction models to the overnight rates.

Bibliography

[1] Corporate Finance Institute. Overnight rate: The interest rate that depository institutions charge each other for overnight lending. https://corporatefinanceinstitute.com/resources/knowledge/finance/overnight-rate/, 2018.

[2] Financial Stability Board. Overnight risk-free rates. *Report*, 2019.

[3] International Swaps and Derivatives Association, Inc. Understanding IBOR benchmark fallbacks, 2020.

[4] Prapanna Mondal, Labani Shit, and Saptarsi Goswami. Study of effectiveness of time series modeling (arima) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*, 4(2):13, 2014.

[5] John H Morgan. Forecasting exchange rates: Theory and practice. *Managerial Finance*, 1981.

[6] Francesco Rundo, Francesca Trenta, Agatino Luigi di Stallo, and Sebastiano Battiato. Machine learning for quantitative finance applications: A survey. *Applied Sciences*, 9(24):5574, 2019.

[7] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168, 2006.

[8] European Central Bank. The euro short-term rate (€STR) methodology and policies. *European Central Bank*, 2021.

[9] New York Federal Reserve Bank of New York. Transition from LIBOR. https://www.newyorkfed.org/arrc/sofr-transition#aboutsofr, n.d. accessed: May 2021.

[10] New York Federal Reserve Bank of New York. Additional information about reference rates administered by the new york fed. https://www.newyorkfed.org/markets/reference-rates/additional-information-about-reference-rates#tgcr_bgcr_sofr_calculation_methodologyr, n.d. accessed: May 2021.

[11] Federal Reserve Bank of New York. Additional information about reference rates administered by the new york fed. https://www.newyorkfed.org/markets/reference-rates/additional-information-about-reference-rates#information_about_treasury_repo_reference_rates, 2021.

[12] Bank of England. SONIA interest rate benchmark. https://www.bankofengland.co.uk/markets/sonia-benchmark, n.d. accessed: May 2021.

[13] Bank of England. SONIA key features and policies. https://www.bankofengland.co.uk/markets/sonia-benchmark/sonia-key-features-and-policies, n.d. accessed: May 2021.

[14] Sriya Anbil, Alyssa Anderson, and Zeynep Senyuz. What happened in money markets in september 2019? https://www.federalreserve.gov/econres/notes/feds-notes/what-happened-in-money-markets-in-september-2019-20200227.htm, 2020. accessed: July 2021.

[15] Schule Roth & Zabel. Overnight LIBOR and SOFR performance during the COVID-19 crisis. https://www.srz.com/images/content/1/7/v4/171291/042220-SRZ-Alert-Overnight-LIBOR-and-SOFR-Performance.pdf, 2020.

[16] Bank of England. Monetary policy summary and minutes of the monetary policy committee meeting. https://www.bankofengland.co.uk/monetary-policy-summary-and-minutes/2018/august-2018#:~:text=0.75, 2018.

[17] Szu Ping Chang, BBC. UK interest rates cut in emergency move. "https://www.bbc.com/news/business-51831004, 2020.

[18] Jonathan D Cryer and Kung-Sik Chan. *Time Series Analysis: With Applications in R*, volume 2. Springer, 2008.

[19] Rob J Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, 2018.

[20] George EP Box, Gwilym M Jenkins, and G Reinsel. Time series analysis: Forecasting and control. *BoxTime Series Analysis: Forecasting and Control*, 1970.

[21] William WS Wei. Time series analysis. In *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*. The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2: Statistical Analysis, 2006.

[22] RE Abdel-Aal and Ahmed Z Al-Garni. Forecasting monthly electric energy consumption in eastern Saudi Arabia using univariate time-series analysis. *Energy*, 22(11):1059–1069, 1997.

[23] Benoit Mandelbrot. The variation of some other speculative prices. *The Journal of Business*, 40(4):393–413, 1967.

[24] Robert F Engle. Autoregressive Conditional Heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the econometric society*, pages 987–1007, 1982.

[25] Tim Bollerslev. Generalized Autoregressive Conditional Heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

[26] S Radha and M Thenmozhi. Forecasting short term interest rates using ARMA, ARMA-GARCH and ARMA-EGARCH models. In *Indian Institute of Capital Markets 9th Capital Markets Conference Paper*, 2006.

[27] Ďurka Peter and Pastoreková Silvia. ARIMA vs. ARIMAX–which approach is better to analyze and forecast macroeconomic time series. In *Proceedings of 30th International Conference Mathematical Methods in Economics*, volume 2, pages 136–140, 2012.

[28] Andreea-Cristina Petrică, Stelian Stancu, and Alexandru Tindeche. Limitation of ARIMA models in financial and monetary economics. *Theoretical & Applied Economics*, 23(4), 2016.

[29] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[30] Vahid Mirjalili Sebastian Raschka. *Python Machine Learning: Machine Learning and Deep*

Learning with Python, Scikit-learn, and TensorFlow. Packt Publishing Ltd., Birmingham B3 2PB, UK, 2017.

[31] J. Brownlee. A tour of machine learning algorithms. https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/, 2019.

[32] XLSTAT. Ordinary Least Squares Regression (OLS). https://www.xlstat.com/en/solutions/features/ordinary-least-squares-regression-ols, n.d. accessed: July 2021.

[33] Raymond E Wright. *Logistic Regression*. American Psychological Association, 1995.

[34] Luckyson Khaidem, Snehanshu Saha, and Sudeepa Roy Dey. Predicting the direction of stock market prices using Random Forest. *arXiv preprint arXiv:1605.00003*, 2016.

[35] Jan Täppinen. *Interest Rate Forecasting with Neural Networks*. Valtion taloudellinen tutkimuskeskus VATT, 1998.

[36] J. Cornelissen. A study on forecasting sofr with a recurrent neural network using long short-term memory cells, June 2021.

[37] Jack V Tu. Advantages and disadvantages of using artificial neural networks versus Logistic Regression for predicting medical outcomes. *Journal of clinical epidemiology*, 49(11):1225–1231, 1996.

[38] Gérard Biau and Erwan Scornet. A Random Forest guided tour. *Test*, 25(2):197–227, 2016.

[39] Builtin. A complete guide to the Random Forest algorithm. https://builtin.com/data-science/random-forest-algorithm, 2019.

[40] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[41] European Central Bank. Historical €short-term rate. https://sdw.ecb.europa.eu/browse.do?node=9698302, n.d. accessed: May 2021.

[42] Federal Reserve Bank for New York. Secured overnight financing rate. https://www.newyorkfed.org/markets/reference-rates/sofr, n.d.

[43] Bank of England. SONIA data series. https://www.bankofengland.co.uk/boeapps/database/fromshowcolumns.asp?Travel=NIx&ShadowPage=1&SearchText=sonia&SearchExclude=&SearchTextFields=TC&Thes=&SearchType=&Cats=&ActualResNumPerPage=&TotalNumResults=12&XNotes2=Y&C=5JK&ShowData.x=41&ShowData.y=34, n.d. accessed: May 2021.

[44] Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

[45] Georgios Efstathopoulos. Python for finance, part 3: Moving average trading strategy, 2021.

[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[47] Robert Nau. Notes on nnnseasonal arima models. *Fuqua School of Business, Duke University*, 2014.