Enhancing Credit Risk Prediction in Retail Banking: Integrating Time Series and Classical ML Algorithms

by

Sebastian Hendrikus Goldmann

A thesis
presented to the University of Twente
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Industrial Engineering and Management

Enschede, Netherlands, 2024

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Author

S.H. Goldmann

Supervisors University of Twente

Dr. J.R.O. Osterrieder (Jörg)
Dr. M.R. Machado (Marcos)

Supervisor ING Bank

V. Etesse (Voichita)

Research institute: University of Twente
Faculty: Behavioural, Management and Social Sciences
Educational program: Industrial Engineering & Management
Specialization: Financial Engineering & Management

Colloquium date: 23 February 2024
Pages: 88

Abstract

This thesis investigates the application of Time Series Classification (TSC) algorithms to enhance credit risk models, focusing on the historical balance data of retail customers. The primary aim was to explore how TSC models could improve the discriminatory power of these risk models. Employing various TSC techniques, including deep learning, shapelets, and Canonical Interval Forest (CIF) algorithms, the study rigorously evaluated their performance in predicting the Probability of Default (PD). Additionally, this thesis contributes to the evolving field of credit risk modeling by introducing a novel approach that integrates TSC with traditional credit risk assessment methods.

The research revealed that TSC algorithms, particularly when applied to end-of-day balance data, have the potential to significantly enhance the predictive accuracy of credit risk models. The CIF model, in particular, demonstrated notable efficacy, rivaling the performance of existing credit risk models. However, applying TSC algorithms to multivariate monthly data showed limited effectiveness, suggesting the removal of critical information in such data aggregation. The study also highlighted the interpretability challenges with complex TSC models and the need for more holistic data inclusion for a comprehensive credit risk assessment.

Keywords: Time Series Classification, Credit Risk Modeling, Machine Learning, Financial Data Analysis, Probability of Default, Canonical Interval Forest, Shapelets, Deep Learning.

Acknowledgements

I want to extend my heartfelt gratitude to my university supervisors, Jörg Osterrieder and Marcos Machado, for their invaluable support and guidance throughout the process of writing this thesis. Their confidence in my research abilities has been instrumental in achieving these results.

I am also immensely grateful to Voichita Etesse for providing me with the opportunity to write this thesis within ING. The access to crucial data and the trust bestowed upon me have been essential for my research and have enlightened me on the principles of effective leadership and management.

A special thanks to Daniel Prins, whose role as a model developer made him an excellent sparring partner. His technical insights greatly enhanced my modeling choices, contributing significantly to the depth of my work.

I also appreciate Busra Cikla's readiness to answer all my queries regarding using Google Cloud Platform and data handling. Her support was pivotal in navigating these complex aspects of my research.

Lastly, I would like to express my gratitude to Stefano Guastamacchia for introducing me to the project that forms the basis of this thesis. His efforts connecting me with every necessary stakeholder at the beginning of my research journey were fundamental in setting the stage for my work.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence 10, 28, 39
AUC	Area Under Curve 11, 16, 27, 37, 41, 42, 44, 46, 48, 49
CIF	Canonical Interval Forest 14, 22, 35, 36, 41–49, 52
CNN	Convolutional Neural network 24
DSRM	Design Science Research Methodology 5, 6, 8, 18, 40
IFRS 9	International Financial Reporting Standards 9 3
IRB	Internal Ratings-Based 3
LIME	local Intepretable Model-Agnostic Explanations 10, 28
LR	Logistic Regression 9, 10, 19
LSTM	Long-Short Term Memory 9, 12, 14, 23, 28, 36, 41, 42
ML	Machine Learning 2, 3, 5, 9, 10, 12, 25, 37
PD	Probability of Default 19, 27, 37–40, 48–51
PSD2	Payment Services Directive 2 2, 41
ROCKET	Random Convolutional Kernel Transform 15, 24, 28, 36, 41–44
SHAP	SHapley Additive exPlanations 10, 16, 19, 28, 39, 45, 49, 50
SMOTE	Synthetic Minority Oversampling Technique 11, 35
STSF	Supervised Time Series Forest 14, 21, 22
TSC	Time Series Classification 3, 4, 8–10, 12–15, 17–22, 24, 25, 29, 30, 37–40, 42, 48–54

Chapter 1

Introduction

This chapter starts our research journey by introducing the current credit risk models in the literature. We then introduce a case study from a large Dutch bank, using it as a specific example to explore broader issues in the field. This case study helps us clearly define the problem as it is seen in the broader industry. We then introduce our research methodology and describe our research design based on this methodology. By the end of this chapter, we aim to give a clear and complete picture of what this research is about, how we plan to do it, and why it is important, both for the specific case of the Dutch bank and the field of credit risk in general.

1.1 Credit Risk Models

Credit risk, defined by the Cambridge dictionary (2023) as 'the degree to which it is possible that a person, company, or government will not be able to pay back borrowed money', is a fundamental concern in the field of banking and finance. The magnitude of this financial risk is illustrated by the current outstanding household debt in the European Union (EU), which stands at a staggering €6.907 trillion as of June 2023, accounting for 48% of the EU's nominal GDP (CEIC, 2023a). In the Netherlands, the scenario is even more pronounced with household debt totaling €974 billion, constituting 94% of the country's nominal GDP (CEIC, 2023b).

The importance of accurate and robust credit risk models plays a pivotal role in banking. It helps distinguishing between creditworthy and high-risk customers, safeguarding banking institutions from the potential financial risks associated with inappropriate credit

granting decisions (Kao et al., 2012). The dynamic nature of the financial sector has prompted banking institutions to continually seek ways to optimize their loan approval processes, recognizing the substantial financial implications that may arise from inaccurate risk assessments (Antonio Bahillo et al., 2016).

The advancements in cost-effective data storage and computational power have already enabled the banking industry to implement Machine Learning (ML) in credit risk models and enable the automation of the loan approval process. Additionally, the introduction of the Payment Services Directive 2 (PSD2) has been pivotal. PSD2 facilitates banks in sharing customer data, provided they have the necessary consent from the customers themselves (McKinsey, 2018). This regulatory change and the increasing adoption of digital payment systems have significantly contributed to integrating big data analytics into the loan approval process.

The journey of utilizing statistical models for credit risk assessment dates back to the pioneering research conducted by Durand (1941). This study laid the groundwork for a plethora of subsequent research to develop quantitative models to assess consumer credit behavior for making informed credit-granting decisions (Wiginton, 1980). One significant milestone in this evolution was the comprehensive review by Hand and Henley (1997), which highlighted how the surge in credit demand, combined with increased commercial competition and advancements in computational capabilities, had collectively fostered the application of diverse statistical models designed to streamline and enhance the credit-granting process.

Ensemble models have become increasingly popular across a variety of fields. Informally defined, these advanced methods in ML enhance prediction accuracy by combining outputs from multiple simpler ML algorithm (Sagi & Rokach, 2018). The effectiveness of ensemble models in credit risk assessment is particularly notable, as evidenced by various studies, including those by Abellán and Castellano (2017), Lessmann et al. (2015), Tripathi et al. (2022), and Yu et al. (2008). As Thomas (2010) pointed out is that in credit risk modeling there is a search for the 'silver bullet' in optimally constructing credit risk models. Showing that there is no definite solution.

In parallel, hybrid models have emerged as a compelling approach, comparable to ensemble models but distinct in their methodology. These models are characterized by their integration of diverse single-classification models. Combining different classifiers allows hybrid models to capture a broader spectrum of features and insights, which is especially advantageous in complex applications like credit risk assessment (Zhang & Yu, 2024). The efficacy of this approach is also evidenced in the literature, as demonstrated by Gao et al. (2021).

This thesis marks a significant contribution to the literature on credit risk assessment by delving into advancing cutting-edge models. It specifically hones in on integrating the ML models that consider the temporal time aspects of data, an area not thoroughly explored in existing studies. These models belong to a specialized category within classification models, known as Time Series Classification (TSC) models. We propose that merging these models could notably enhance predictive accuracy. The core hypothesis of our research is that combining an ensemble model with a ML approach, which is designed to have the time dimension of data in mind, will result in more effective predictions of credit risk defaults compared to using an ensemble model alone.

1.2 Research Context

1.2.1 Company Problem Background

This study is situated within the Dutch Bank ING. This is one of the largest banks in the Netherlands and is offering service to retail and wholesale customers. The department we are situated in is the model risk management department, which manages model-related risks within the bank. The credit risk models are vital tools to predict potential credit losses from default events, essential for compliance with International Financial Reporting Standards 9 (IFRS 9), Internal Ratings-Based (IRB) calculations, and credit approval decisions. The department focuses on model validation, robustness, alignment with intended purposes, and proposing enhancements.

ING is currently focusing on a strategic objective to enable instant lending to retail customers through automated credit decision models. These models assess credit risk—essentially, the likelihood of a customer defaulting—using customer historical information. Based on this assessment and the bank's risk appetite, these models autonomously approve or reject loan applications and apply risk-based pricing. The model specifically addresses term loans, which are typically used for purposes like buying a car or home renovations. During my thesis work at ING, this model was undergoing a validation process, a vital step given its role in approving loans without human intervention. Additionally, the model estimates defaults on overdrafts and credit cards. Although overdrafts aren't officially considered as disbursed loans by the bank, it is still essential for the bank to estimate the risks of overdrafts.

There are plans to evolve this model into an application programming interface (API) for wholesale clients. This would allow these clients, with customer consent, to use our

default estimates based on customer data. To contextualize the model's scale, the training dataset currently encompasses about €800 million in loans disbursed in the Netherlands over six years.

1.2.2 Problem Description

In credit risk modeling, a significant scientific challenge lies in integrating historical transactional data of retail customers. For example, how do you ensure the data's time dimension is understood in the extracted features? Traditionally, this involves a feature engineering process that relies heavily on expert judgment. This method, while effective, could potentially overlook crucial insights within time series data due to its reliance on human experience and subjective decision-making.

Emerging automated techniques in feature engineering present a promising avenue for research. These methods offer the possibility of streamlining the feature engineering process, reducing the reliance on manual input, and potentially uncovering more nuanced patterns within the time series data.

The central scientific problem in credit risk modeling can be articulated as follows:

"Credit risk modeling heavily depends on labor-intensive feature engineering, which may miss critical insights from time series data."

This issue extends to exploring algorithms that can operate without traditional feature engineering or have this integrated in the model design, thereby capturing more intricate patterns and relationships in the balance data of retail customers. The ultimate goal is to enhance the predictive accuracy, granularity, and risk-based pricing in lending decisions. While this research is initially conducted in the context of ING Bank, its implications are far-reaching, potentially benefiting other banks and industries facing similar challenges in handling complex feature engineering and selection in classification problems.

1.3 Research Approach

1.3.1 Research Objective

The research aim is to explore alternative methods to extract information from the historical balance data of retail customers, specifically, we will be focusing on TSC algorithms.

These algorithms are designed to incorporate the time aspect and do not require us to create features based on expert judgement and can directly interpret the raw time series data of the historical balance account. We expect that this reduces reliance on expert-judgement, minimize manual feature engineering, and still ensure comprehensive information capture for accurate credit risk assessment.

1.3.2 Research Questions

To address the stated research objective, the primary research question guiding this study is as follows:

How can time series classification algorithms on historical balance data of retail customers enhance credit risk models' discriminatory power?

The following sub-research questions are formulated to answer the main research questions.

1. What are the current best practices for time series classification of historical balance data to predict the default of retail customers?
   (a) What current methods used for time series data in credit risk prediction?
   (b) What time series classification algorithms exist that could work on credit risk predictions?
2. What evaluation metrics are most suitable for assessing and comparing predictive performance among different ML models?
3. How can the decisions made by the credit risk prediction model be explained to stakeholders, ensuring transparency and understanding of the model's behavior?

1.3.3 Research Methodology

We have adopted a well-defined research methodology to ensure our research is structured and directed effectively. Our chosen approach is the Design Science Research Methodology (DSRM), as introduced by Peffers et al. (2007), the overview of which is illustrated in Figure 1.1. We selected this methodology for several compelling reasons. First, the DSRM is renowned for its evidence-based approach, ensuring that our problem-solving strategy

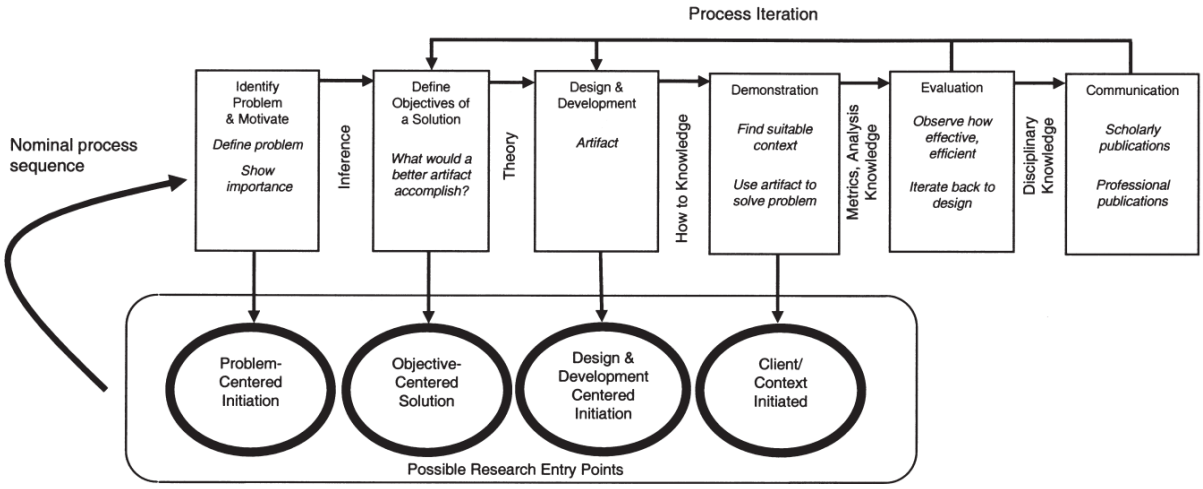


Figure 1.1: Design Science Research Methodology (Peffers et al., 2007)

is grounded in a robust, well-established method. Second, each step in the DSRM process aligns closely with the needs of our study, ensuring a comprehensive approach to system design without overlooking critical aspects. Lastly, the DSRM offers an iterative process with flexible entry points for research, making it particularly well-suited to address our specific problem.

1.3.4 Research Design

Having established our research objective, formulated the research questions, and selected our research methodology, we are now poised to integrate these elements into a cohesive research design. It's important to note that while our approach is structured, as depicted in Figure 1.1, it also incorporates a feedback loop. This loop allows for flexibility and iterative refinement throughout the research process. The steps we will follow in our research, as guided by the DSRM methodology, we outline it as follows:

1. **Problem identification:** This stage involves recognizing and defining the central problem our research aims to address. This is done at the beginning of our research as we have already done in Chapter 1.

2. **Solution objectives:** Here, we outline our research goals. The focus is to answer our research questions in the literature study. This enables us to have a good theoretical basis in creating a solution objective.

3. **Design and development:** This phase is about creating a blueprint for our solution. It involves translating theoretical solutions we found in literature into a practical implementation, where ideas are transformed into a tangible model.

4. **Demonstration:** During this stage, we showcase how our solution works in an experimental setup. It's about providing a practical example of our research in action, illustrating its application and effectiveness.

5. **Evaluation:** Evaluation is critical to assess the success of our solution against defined criteria. This involves testing and analyzing the results to determine if the objectives have been met and to what extent our solution has addressed the problem.

6. **Communication:** The final step is effectively communicating our research findings, methodology, and conclusions. This includes the overall thesis, presenting results, and discussing the implications and potential for future research in the field.

The structure of the thesis is as follows: Chapter 2 reviews the literature on credit risk prediction models, focusing on classification methodologies and the handling of time series data. Chapter 3 proposes a model improvement that outlines our methodology, including evaluation metrics and techniques for model explainability, aligning with our research questions. Chapter 4 details our dataset and experimental setup, setting the foundation for empirical work and ensuring transparent data handling. Chapter 5 presents our experimental results, comparing existing models with our proposed enhancements. Finally, Chapter 6 concludes the thesis, discussing our findings, their implications, and potential areas for future research.

Chapter 2

Literature Review

This chapter delves into the existing literature on credit risk modeling, emphasizing the nuances of TSC. Our exploration begins with thoroughly examining the current state-of-the-art credit risk models, followed by analyzing how these models are explained and their performance quantified in existing research papers. We then shifted our focus to time series data, closely examining the classification algorithms that could be employed to derive meaningful insights from them. The solution objective is this chapter's design step in the DSRM. Creating guidance and a foundation for model enhancements.

2.1 Methodology

The research questions we are trying to answer require us to search for unbiased information. We accomplish this by using Scopus[1], an Elsevier search engine created to search for academic literature. We used the search term '(consumer AND credit AND risk AND prediction) OR (consumer AND default AND prediction) OR (consumer AND scorecard)'. This resulted in 344 search results on the Scopus search engine. To further specify, we selected only English articles published between January 2018 and October 2023. This resulted in 93 search results. We then selected the areas of - economics, econometrics and finance, - business management and accounting, - computer science, and - decision sciences. This resulted in a total of 72 papers. From this point, a selection was made that was most suitable for this research by reading the titles and abstracts, resulting in 30 papers that will be read thoroughly. Appendix A gives an overview of the papers used in this literature.

[1] https://www.scopus.com/

We uncover the three sub-research questions in each paper and whether it was addressed. Namely, what classification method was used, how was the performance evaluated, and how did the authors explain the model. We then looked at any links between TSC and credit risk predictions. Afterward, we purely focus on the domain of time series classification that is unrelated to credit risk but could be applied to historical balance time series data.

2.2 Credit Risk Models

The literature on credit risk models is vast and complex, encompassing various methodologies and approaches (Dastile et al., 2020). Credit risk models in the literature employ supervised ML techniques, such as Logistic Regression (LR) (Brygała, 2022; Li et al., 2023; Liang & Cai, 2020). However, as Li et al. (2023) points out, these linear models often make unrealistic assumptions about the distribution and relationships in the data and require creating highly discriminant features. Common input features for credit risk models include application data (e.g., age, marital status, time with the bank), aggregated balance data on a monthly or yearly basis and feature engineering on the transactional data (Brygała, 2022; Correa Bahnsen et al., 2016; Gao et al., 2021; Kvamme et al., 2018; Ma & Lv, 2019; Mahbobi et al., 2021; Tripathi et al., 2022). In recent years, we see in the research that ensemble models, such as XGBoost, outperform single classifiers in credit risk modeling, thanks to their ability to capture non-linear relationships in the data (Abellán & Castellano, 2017; Alam et al., 2020; Lessmann et al., 2015; Tripathi et al., 2022; Yu et al., 2008). Furthermore, H. Wang (2021) and Merćep et al. (2021) highlight the potential for further improvements in the application of big data and non-linear models in consumer finance credit risk management.

Various studies have demonstrated the successful application of advanced ML techniques that surpass traditional credit models discriminatory power (Gao et al., 2021; Kvamme et al., 2018; Mahbobi et al., 2021; Merćep et al., 2021; van Thiel & van Raaij, 2019; Yu et al., 2008). The research of Gao et al. (2021) compared the performance of a combined XGBoost model with Long-Short Term Memory (LSTM) network model and a standalone XGBoost model, both applied to credit information data. The results indicated that the deep learning approach improved performance and eliminated the need to construct 81 features. The study found that features related to transaction flow are the most powerful for predicting credit card defaults. It also highlighted that extracting useful features from transaction data requires deep understanding. But this type of deep learning application is not uncontroversial as Gunnarsson et al. (2021) points out that standalone

deep learning does not outperform XGBoost on their data sets.

Our research identified the standalone application of XGBoost as a leading method in consumer credit risk modeling. This assertion is supported by extensive literature reviewed in Appendix A. Notably, our review reveals a scarcity of studies integrating classical ML techniques with TSC. However, the singular study of Gao et al. (2021) we identified in this context demonstrates promising outcomes. This finding suggests a significant opportunity for enhancing existing models by incorporating TSC methodologies into traditional ML frameworks.

2.2.1 Explainability

The current challenge with advanced ML models is their 'black-box' nature, which makes them difficult to interpret and explain. This is a critical issue, especially considering the implementation of the Basel II and 'general data protection regulation' acts that mandate European banks to maintain a certain level of explainability in their data-based decision-making models (de Lange et al., 2022). Despite this requirement, most of the literature we investigated in Appendix A on credit decision models tends to overlook the explainability of their proposed models. Common metrics utilized in studies that do address model explainability include feature importance (Brygała, 2022; Gao et al., 2021; Kao et al., 2012; Ma & Lv, 2019), and SHapley Additive exPlanations (SHAP)-values (Bücker et al., 2022; de Lange et al., 2022). The paper of Bücker et al. (2022) posits that interpreting modern complex ML models can be comparable to more traditional ones, such as LR. The author proposes a framework to facilitate this process and underscores that data preparation and explicit feature engineering will become increasingly costly and complex as data grows, making their proposed framework essential for compliance to keep understanding advanced models. There is an ongoing effort within the realm of explainable Artificial Intelligence (AI) to render black-box models more transparent and trustworthy, ultimately facilitating our comprehension of these advanced models (Adadi & Berrada, 2018). A notable method also introduced to explain AI is local Intepretable Model-Agnostic Explanations (LIME) (Ribeiro et al., 2016). Although not seen in the literature on credit risk models, we do not include LIME since improving the explainability of the models is not the main focus of our research.

2.2.2 Model Evaluation

A standard method to test model performance involves splitting the data into training and test set. The splitting can be a single or cross-validation split where the test set varies over iterations. Most studies in our literature study credit risk models prefer the single train-test split (Addo et al., 2018; Alam et al., 2020; Gao et al., 2021; Kvamme et al., 2018). In contrast, some performed a cross-validated evaluation of the model (Liang & Cai, 2020; Yu et al., 2008). The test set, serving as an independent observation, helps evaluate the model's performance because the model did not see this data. As noted by Cawley and Talbot (2009), a decrease in accuracy on the test set could suggest overfitting to the training data.

Addressing the challenge of imbalanced datasets in credit risk modeling is crucial, as the class of defaulted customers is typically underrepresented. Such imbalance can skew performance indicators, leading to a bias towards the majority class at the expense of the minority class. Several techniques in literature are employed to counteract this. For instance, Alam et al. (2020) uses undersampling, oversampling and Synthetic Minority Oversampling Technique (SMOTE) to balance the dataset, effectively mitigating this issue.Addo et al. (2018) and Alam et al. (2020) showed for his dataset that the specific K-means SMOTE technique resulted in the best accuracy gain. Alternatively, employing performance metrics that account for class imbalance is another approach. A majority of studies in our literature review, including works by (Addo et al., 2018; Correa Bahnsen et al., 2016), utilize the confusion matrix and the Area Under Curve (AUC) score as evaluation metrics. The confusion matrix sheds light on the true positive/true negative and false positive/false negative. The AUC score represents the area under the receiver operating characteristic curve, indicating the model's ability to distinguish between classes. Other metrics like the Brier score (Kvamme et al., 2018; Mahbobi et al., 2021), the H measure (Kvamme et al., 2018), and the F-measure (Correa Bahnsen et al., 2016; Dastile et al., 2020) are also used. The Kolmogorov-Smirnov (KS) statistic, as suggested by Y. Tan and Zhao (2022), is particularly beneficial for models dealing with imbalanced datasets.

2.3 Time Series

Banks hold time series data on their customers, such as the end-of-day balances of the current and savings accounts and the daily debited/credited amounts. These time series are often aggregated to monthly/quarterly/yearly statistics to create and tabulate highly discriminant features (Gao et al., 2021; Mahbobi et al., 2021). We also mentioned earlier

that creating and testing these features requires effort and domain knowledge. In recent years, the research in TSC has seen a rise in publications. Two literature review studies of Bagnall et al. (2017) and Middlehurst et al. (2023) looked at state-of-the-art TSC models on a large number of different time series datasets of the archives accessible at the University of California (no financial data was present in any of the datasets). These algorithms show good predictive accuracy on various applications in TSC challenges. To our knowledge, we have not seen any research in combining TSC and credit risk models, except for deep learning methods in, for example, the study of Kvamme et al. (2018) where a neural network was applied, or in the study of Liang and Cai (2020) where an LSTM was implemented. These methods eliminate the requirements of creating discriminant features and directly classify time series. In the next section, we will look at interesting TSC models that could be applied to time series data of customers in the banking industry. We will not test all the possible TSC methods as done in the study of Bagnall et al. (2017) as we do not have the resources for that. Consequently, our selection of methods within the TSC algorithm taxonomy was strategic and aligned with our research objectives. We opted for interval methods, drawing inspiration from the current model used by ING, which also incorporates intervals. The choice of deep learning methods was influenced by the successful implementations found in the studies by Kvamme et al. (2018) and Liang and Cai (2020). We included shapelets for their unique ability to identify distinct time series shapes, thereby providing a highly explainable approach to classification. Lastly, hybrid-based methods were chosen for their capacity to amalgamate the strengths of various methodologies, thereby optimizing overall performance.

2.3.1 Time Series Classification

TSC has emerged as a prominent area of study in ML, particularly due to the unique attributes of time series data (Chakraborty & Yoshida, 2017; Middlehurst et al., 2023). In TSC, as defined by Löning et al. (2019), we deal with N instances of training data, each comprising a pair of features and labels (x_i, y_i), where $i = 1, \ldots, N$. Each feature instance x_i in this context is a time series, which is denoted as $x_i = (x_i(t_1), \ldots, x_i(t_T))$. The aim in TSC is to utilize this training data to construct a predictive model $\hat{f}$ that can accurately forecast a target class label $\hat{y}$ for a new, unseen time series x^*, with the relationship expressed as $\hat{y} = \hat{f}(x^*)$.

In contrast to standard classification tasks, TSC specifically addresses the classification of data points that are sequentially ordered. While typical classification methodologies operate under the assumption that data points are independent and identically distributed, TSC assumes a dependence between data points based on their order in time (Löning et al.,

2019). This key distinction highlights the necessity of considering the entire sequence of data points in TSC, rather than evaluating them in isolation.

The mathematical representation in TSC considers these sequential dependencies. For a univariate time series, this is represented as (Löning et al., 2019):

$$x_i = (x_i(t_1), x_i(t_2), \ldots, x_i(t_T)) \tag{2.1}$$

In cases of multivariate time series, where multiple observations are recorded at each time point, the representation is:

$$x_i = \begin{pmatrix} x_{i,1}(t_1) & x_{i,1}(t_2) & \ldots & x_{i,1}(t_T) \\ x_{i,2}(t_1) & x_{i,2}(t_2) & \ldots & x_{i,2}(t_T) \\ \vdots & \vdots & \ddots & \vdots \\ x_{i,M}(t_1) & x_{i,M}(t_2) & \ldots & x_{i,M}(t_T) \end{pmatrix} \tag{2.2}$$

In equation 2.2, $x_{i,m}(t)$ signifies the observation of the m-th variable at time t for the i-th series. The length of the time series, T, may vary across different instances. The goal in TSC is then to define a model $\hat{f}$ that maps a time series $\mathbf{X}$ to a discrete class label Y. This model is formally described as:

$$\hat{f} : \mathbf{X} \to Y \tag{2.3}$$

In equation 2.3, $\mathbf{X}$ represents the domain of all possible time series data, either univariate or multivariate, and Y denotes the set of discrete class labels $Y = \{y_1, y_2, \ldots, y_k\}$. The model $\hat{f}$, thus, encapsulates the essence of TSC, transforming the temporal patterns within the time series data into meaningful class predictions (Löning et al., 2019).

Feature Interval Based

In the domain of TSC, identifying meaningful patterns often hinges on strategically extracting relevant features. Christ et al. (2018) emphasized that feature engineering for time series data can be an arduous process aimed at isolating salient attributes. One common strategy in handling transactional data, as highlighted by Correa Bahnsen et al. (2016), is to aggregate historical transactions. This aggregation simplifies the observation of customer spending behavior and addresses the challenge of uneven time series lengths frequently found in transactional data sets. Building on this, Alam et al. (2020) aggregated

credit card data on a monthly basis. Creating features based on how much is billed to the credit card and how much is paid off. Thereby, 24 distinct features of the last 12 months to enhance credit card default predictions.

While feature extraction forms the foundation of TSC, the methods used for classification are equally pivotal. Deng et al. (2013) introduced the time series forest, a paradigm-shifting approach that combines the principles of entropy gain and a distance measure, termed 'Entrance' gain. This method aligns computational efficiency with the complexity of time series. In performance comparisons, it has displayed an edge over traditional techniques, such as one-nearest-neighbor classifiers paired with dynamic time warping.

In pursuit of concise yet informative time series feature sets, Lubba et al. (2019) unveiled Catch-22. This method's genius lies in its ability to condense a vast 4791 feature set down to 22 features. Such drastic dimensionality reduction ensures a monumental drop in computation time. Remarkably, this reduction comes at only a slight compromise on classification accuracy, rendering Catch-22 an efficient solution for real-world challenges (Lubba et al., 2019).

Noteworthy methods in the TSC landscape include the Supervised Time Series Forest (STSF) and the Canonical Interval Forest (CIF). While the former, as explored by Cabello et al. (2020), accentuates certain sub-series within the original time series through a top-down search mechanism, the latter, as done by Middlehurst et al. (2020), synthesizes STSF and Catch-22 methodologies into a CIF model. This merger results in a better-performing classification performance as evidenced by the benchmark of Middlehurst et al. (2023).

Deep Learning

Deep learning presents an advantage in eliminating the need to create explicit time series data features. Neural networks can autonomously extract and identify complex relationships. For instance, Kvamme et al. (2018) implemented a convolutional neural network on historical consumer transaction data, demonstrating good predictive accuracy using only end-of-day balance information. Another study by Gao et al. (2021) utilized an ensemble model, XGBoost for applicant information, combined with a LSTM neural network trained on transaction data, resulting in strong classification performance for default prediction. Additionally, they implemented a deep neural network and achieved high accuracy with multiple network designs. A comprehensive review on deep learning methods for TSC was conducted by Fawaz et al. (2019) emphasizing that TSC is one of the most challenging problems in data mining. However, a consensus regarding using artificial neural network models in default prediction modeling is yet to be established. Concerns regarding these

models' stability, performance, and transparency were raised by Addo et al. (2018), indicating that they may not always provide the best performance. Similarly, Gunnarsson et al. (2021) concluded that deep neural networks may not be optimal models for credit scoring and are outperformed by XGBoost. Dempster et al. (2020) showcased that linear classifiers equipped with Random Convolutional Kernel Transform (ROCKET) can achieve state-of-the-art accuracy. Notably, these classifiers manage this with only a fraction of the computational burden associated with their counterparts.

Shapelets

Shapelets, initially introduced by Ye and Keogh (2009), are defined as 'subsequences within time series data that are maximally representative of a particular class'. Ye and Keogh (2009) highlights the substantial advantages of utilizing shapelets in TSC. One significant advantage is their ability to yield interpretable results by showcasing the specific shapelet within the time series, facilitating comprehension for domain experts. Furthermore, shapelets demonstrate superior accuracy and robustness on certain data sets due to the minimal number of hyper-parameters requiring tuning. And lastly, they offer improved computational efficiency compared to existing state-of-the-art approaches in TSC. Grabocka et al. (2014) introduced an effective algorithm that searches for near-optimal shapelets based on stochastic gradient descent and does not need to try out many candidate solutions. Shapelets have seen a rise in literature in the past decade (J. Chen et al., 2022; Eamonn, 2022). However, no literature is found on the application of financial data.

Hybrid Models

The study of Bagnall et al. (2017) concluded that the HIVE-COTE hybrid model significantly outperforms other models in the classification task. While this model boasts remarkable accuracy, it demands substantial computational resources, leading some to favor simpler alternatives. Nevertheless, its prominence led to an enhanced version, HIVE-COTEv2, introduced by Middlehurst et al. (2021). This newer model further elevates classification accuracy but also intensifies computational challenges. The study of Middlehurst et al. (2023) recently reproduced the study of Bagnall et al. (2017) with new and expanded UCR archives and new TSC algorithms introduced. This study again demonstrated the prevailing performance of the HIVE-COTE-v2.

2.4 Conclusion

In addressing our main research question, "How can time series classification algorithms on historical balance data of retail customers enhance credit risk models' discriminatory power?", our literature review identified several key findings on our sub-research questions:

1. **Current Best Practices for Time Series Classification (Sub-Question 1):** We identified XGBoost as the current state-of-the-art model in credit risk modeling, offering robust performance across various datasets. Our exploration into the time series classification of historical balance data highlighted three promising methods: feature interval-based, shapelet-based, and deep learning-based algorithms. These methods, particularly when integrated into existing models like XGBoost, might have the potential to enhance the discriminatory power of credit risk models significantly.

   (a) *Methods Used for Time Series Data in Credit Risk Prediction (Sub-Question 1a):* Current practices predominantly utilize feature engineering, creating summary statistics on intervals of time series data. However, studies like those of Kvamme et al. (2018) and Liang and Cai (2020) demonstrate the efficacy of deep learning methods in analyzing transactional data, suggesting a shift towards more complex approaches.

   (b) *Time Series Classification Algorithms for Credit Risk Predictions (Sub-Question 1b):* While traditional methods focus on feature engineering, our review indicates a growing potential in algorithms that directly classify time series data which could be applied to banking customer time series data.

2. **Evaluation Metrics for Predictive Performance (Sub-Question 2):** The literature consistently utilizes the AUC score. Particularly useful in addressing class imbalances commonly encountered in credit risk data. Studies mostly use a single train and test split, while some apply cross-validation to enhance the robustness of their outcomes. Beyond the AUC score, other metrics like the Brier score and the H measure are also used, underscoring the diversity of evaluation approaches in current research.

3. **Explainability of Credit Risk Prediction Models (Sub-Question 3):** Despite the critical importance of model explainability for regulatory and stakeholder trust, our review notes a general oversight in this area. However, methods like SHAP analysis and feature importance metrics are employed in some studies.

In summary, our research reveals significant opportunities for enhancing credit risk models through the integration of TSC algorithms. By incorporating TSC techniques into current models like XGBoost, there is potential for deeper insights into customer behavior and an improvement in predictive accuracy. Our study suggests that future research should focus on the practical implementation of these advanced TSC methods in credit risk models, aiming not only to improve discriminatory power but also to ensure models remain interpretable and transparent to meet regulatory requirements and stakeholder expectations. Embracing these methodologies could lead to more accurate, efficient, and trustworthy credit risk assessment tools in the banking industry.

Chapter 3

Methodology

This chapter delves into the methodology adopted for our study, laying the groundwork for the model design and the selection of TSC models. Our approach is twofold, beginning with an in-depth analysis of the existing model currently utilized at ING. By doing so, we aim to provide a comprehensive understanding of how our envisioned model design can be integrated and utilized effectively in the broader industry.

Following the analysis of the current model, we transition to exploring the TSC models that we propose to apply in this context. The selection of these models is a critical component of our research, and as such, we will provide detailed reasoning behind the choice of each model.

This chapter offers a clear and detailed exposition of our methodological approach. From the analysis of existing models to the rationale behind selecting specific TSC models, we endeavor to present a well-rounded and robust methodology that underpins the subsequent stages of our research. This chapter corresponds to the design and development step in the DSRM process.

3.1 Model Design

The explored literature on credit risk models and advanced TSC algorithms sets the stage for discussing our proposed implementation strategy. ING currently employs automated credit decision models, which leverage customer application details and historical transaction/balance information to enable prompt lending solutions for retail clients. This process utilizes an ensemble approach, integrating two distinct sub-models (X and Y) to predict

Probability of Default (PD) for term loans and overdraft facilities. Figure 3.1 shows an overview of the model architecture. In the proposed framework, Model X is designed to process application data using a LR approach. In contrast, Model Y is specifically tailored to leverage historical transaction data through the application of the XGBoost algorithm. The innovation lies in the ensemble model that combines the PD estimates derived from both Model X and Model Y. This integration is further refined by incorporating a cut-off mechanism aligned with ING's risk appetite. The culmination of this process is the final loan approval decision, which is determined by an advanced LR model, effectively integrating the insights garnered from both models.

Figure 3.1 provides a comprehensive overview of the current model architecture within ING. It details the design of models X and Y, illustrating the specific data flow processes for each. Our proposed TSC model will utilize balance information as its primary input to estimate PD, subsequently feeding into model Y. Here, 'balance information data' refers to the historical balance records of a customer's current and savings accounts. In contrast, 'transaction information' encompasses all transaction details, such as transaction type, amount, date, etc. 'Application data' includes all applicant-specific information.

To understand the logic of why we believe this addition to the model could lead to a great accuracy improvement, we need to take a closer look at model Y.[1] The design of model Y is encompassed with a great effort of feature engineering to tabulate the balance data and transaction data. We will not dive into the details of all these features, as this is outside the scope of this research. But we give an overview for understanding. 310 features are created on the balance data, and 2472 features are created on the transaction data. The balance features have been aggregated over 1, 3, 6, and 12 months with various indicators. Also, various ratios between these features have been computed. Appendix B.1 gives a high-level overview of the features.

The feature selection pipeline of the model minimized this to a final features set to 10 balance features and 30 transaction features. Notably, a relatively higher number of features originating from the balance account data have survived the selection procedure compared to the transaction data features. The feature selection pipeline can be seen in Appendix B.2. Furthermore, the model explanation was investigated using the mean absolute SHAP values to rank the features; it turns out that four out of the top 5 features were balance account data features. However, interpretation and conclusions should be carefully taken on the mean absolute SHAP values. It is an interesting observation and justifies the increased focus on historical balance data. The final feature set and their absolute mean SHAP values of model Y can be seen in Appendix B.2.1.

[1] This analysis is based on internal documentation and expert opinion

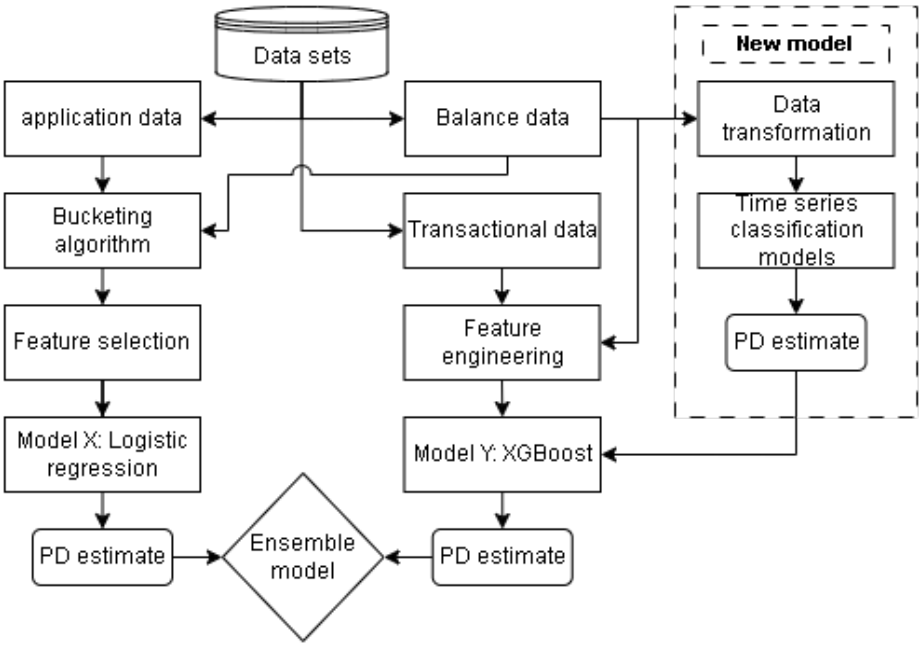


Figure 3.1: Flowchart model architecture.

We argue that the model can be improved by using TSC models on the historical balance data of customers. The current features on the balance data summarize information to aggregated terms of 1, 3, 6, and 12 months. This does indeed give high discriminatory power to the features. But also remove much information from the original data that could contain important information. This model setup also addresses one of the limitations of Kvamme et al. (2018): it incorporates more data than daily aggregated balance data. We assume it can improve the model by applying models designed to understand the time aspect of the classification problem.

3.2 TSC Algorithms

This section presents the selected TSC algorithms for testing on historical consumer balance data. Our literature review explored interval-based, deep learning, and shapelet-based methods. Figure 3.2 provides an overview of the proposed model. We have implemented two types of TSC algorithms: interval-based and deep learning-based.

We hypothesize that interval-based methods are promising for integration into our

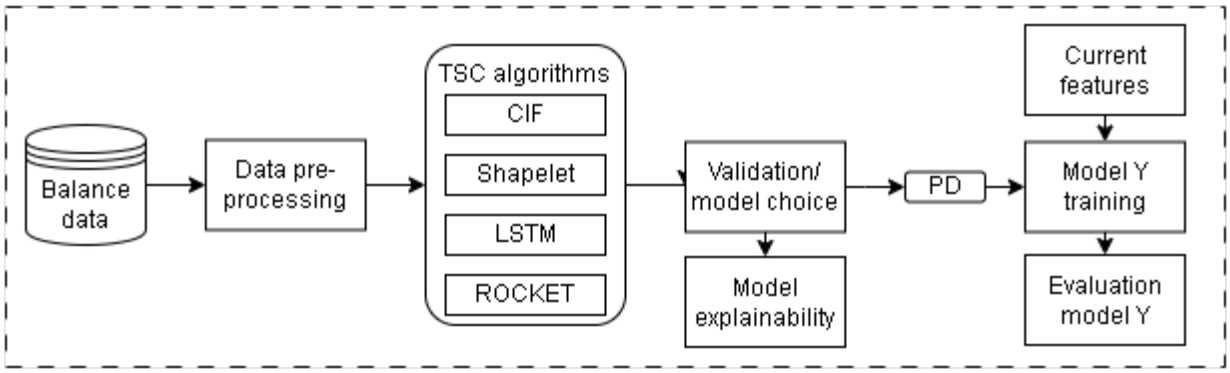


Figure 3.2: Time series classification model design.

model as they align with the approach used in the current model. In the existing framework, intervals of 1, 3, 6, and 12 months are selected, and discriminant statistics for these intervals are computed and utilized as features in the XGBoost model. Unlike this manual selection of intervals, the interval-based TSC algorithms in our study automatically identify intervals, eliminating the need for expert judgment and testing. Further questioning with the model developers also showed that these intervals are chosen, because it has a proven track record in previous models and it is generally accepted in the business interpretation.

Furthermore, deep learning methods have demonstrated effectiveness in credit risk prediction models, as indicated by existing literature. This motivates their inclusion and testing in our proposed TSC framework. An overview of all the TSC literature investigated in this research is shown in Table A.2.

3.2.1 Interval Based

Supervised Time Series Forest

STSF, proposed by Cabello et al. (2020), is an adaptation of the random forest algorithm optimized for time series data. STSF builds on the principles of ensemble learning by generating a multitude of decision trees, where each tree is trained on a randomly selected subset of the data and features.

In the classification process, each decision tree within the STSF makes an independent prediction on the class label for a given time series input. The features used for making these predictions are the mean, standard deviation, minimum, maximum, median, interquartile range, and the slope of the time series subset. STSF distinguishes itself by

efficiently searching for the optimal subset of these interval features that contribute to the most accurate classification.

One notable advantage of STSF, highlighted by Cabello et al. (2020), is the interpretability of its output by visualizing the regions of interest. This is particularly beneficial in practical applications. The final classification is derived from a majority voting scheme among the individual trees, which tends to yield more accurate and reliable results. The simplicity and efficiency of STSF make it particularly appealing for handling time series data, which can vary greatly in length and other properties. Its effectiveness has been demonstrated in several time series classification tasks, proving the approach's utility introduced by Cabello et al. (2020).

Canonical Interval Forest

CIF shares its foundational architecture with the STSF. CIF diverges from STSF by employing the Catch-22 feature set, a powerful toolset introduced by Lubba et al. (2019) for capturing a broad and informative range of time series characteristics.

CIF begins its process by systematically dissecting each time series into predefined canonical intervals. Leveraging the insights provided by Middlehurst et al. (2020), CIF utilizes the Catch-22 features within each segment, encompassing a diverse array of descriptive statistics and characteristics that provide a rich, multidimensional view of the data.

The construction of the CIF's decision trees follows, with each tree being trained on a specific subset of canonical intervals. These intervals, alongside their Catch-22 calculated features and associated class labels, form the basis of the CIF's classification mechanism (Middlehurst et al., 2020).

In the broader context of TSC, CIF's application has proven innovative and advantageous. Its ability to integrate detailed feature sets with interval-based classification strategies allows CIF to deliver superior performance, as evidenced by its benchmarking against other TSC methods in the study of Middlehurst et al. (2020).

While the STSF algorithm applies a recursive interval search and selection based on the Fisher score, the interval search in CIF uses a random interval generation as seen in the original paper of Middlehurst et al. (2020) and in algorithm 1 which shows the implementation in Löning et al. (2019) package that is used in this study. We will also argue here that implementing a more intelligent search could improve this interval search algorithm.

Algorithm 1 CIF Random Interval Creation (Löning et al., 2019)

```
Input: n_intervals, series_length, min_interval, max_interval

FOR each interval j from 0 to n_intervals - 1:
    StartSelected <- random choice between True or False
    IF StartSelected THEN
        start <- random number from 0 to series_length - min_interval
        maxEnd <- minimum of series_length and start + max_interval
        end <- random number from start + min_interval to maxEnd
    ELSE
        end <- random number from min_interval to series_length
        maxStart <- maximum of 0 and end - max_interval
        start <- random number from maxStart to end - min_interval
    END IF
    intervals[j] <- (start, end)
END FOR
```

3.2.2 Deep Learning

LSTM

An LSTM network can learn long-term dependencies in the sequential data. The study of Gao et al. (2021) already combined the LSTM network with an XGBoost model and showed superior performance as compared to a standalone XGBoost model. Implying that the standalone model could not capture the temporal dependencies in the sequential data. The framework we will implement for our LSTM model is sourced from Karim et al. (2019) designed for time series classification tasks.

An LSTM network can learn long-term dependencies in sequential data. The study of Gao et al. (2021) already combined the LSTM network with an XGBoost model and showed superior performance compared to a standalone XGBoost model, implying that the standalone model could not capture the temporal dependencies in the sequential data. The framework we implement for our LSTM model is sourced from Karim et al. (2019), designed for time series classification tasks.

ROCKET

The ROCKET technique, introduced by Dempster et al. (2020), offers a significant leap in the classification of time series data by leveraging the potency of convolutional kernels, traditionally the backbone of Convolutional Neural network (CNN), while introducing an element of randomness to each kernel property. This innovative approach equips ROCKET with the ability to transform time series data into an optimally classifiable format with an unprecedented blend of efficiency and scalability since we do not have to learn all the optimal hyper-parameters of the convolutional kernels.

As Kvamme et al. (2018) demonstrated the successful application of CNNs in default predictions of mortgages, we have a well-founded expectation for ROCKET to perform well in the same setting. Its computational expediency is a notable advantage of ROCKET. Compared to the exhaustive training demands of a full CNN, ROCKET's approach to feature extraction markedly reduces computation time, presenting a swift and streamlined alternative without sacrificing the depth and robustness of the analysis. Once ROCKET has distilled time series data into a rich feature set, subsequent classification can be undertaken by models such as linear classifiers. These classifiers interpret ROCKET's complex feature map, enabling precise predictions while maintaining the model's overall computational lightness.

3.2.3 Shapelets

As introduced in Chapter 2 of our literature study, shapelets are defined as 'subsequences within time series data that are maximally representative of a particular class' Ye and Keogh (2009). We did not find any literature on applying shapelets in credit risk models for consumer loans. Shapelets are particularly interesting for these models due to their direct interpretability and explainability. However, there are some drawbacks; for example, finding shapelets is computationally complex. Multiple algorithm proposals have been made to approximate the top k shapelets, as introduced by Hills et al. (2014). An illustration of a shapelet is given in Figure 3.3. From this, we can extract two features: the location of the best-fitted shapelet and the total absolute distance between all the datapoints in the time series T and the shapelet S

3.2.4 Hybrid Model

The two benchmark studies by Middlehurst et al. (2023) and (Bagnall et al., 2017) showed that HIVE-COTE-v2 is currently the best performing TSC algorithm on a wide range of

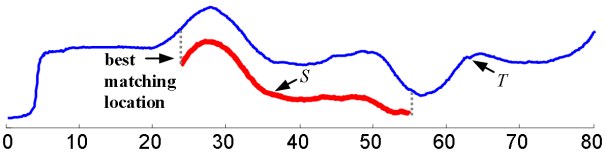


Figure 3.3: Illustration of the best matching location in time series T for shapelet S. (Ye & Keogh, 2009)

classification problems. This model represents a culmination of multiple different types of TSC models, each with its own strengths and weaknesses. While each sub-model can pick up signals from the data with its strengths, the hybrid model combines all these signals into a final classification. However, as the literature also points out, this model is computationally expensive, as it requires training multiple TSC algorithms and complicates interpretation. For these reasons, we chose not to implement this model, as it would lead to high computational costs and provide limited interpretive insights. Nonetheless, this approach does highlight that integrating diverse model types on time series data can enhance performance.

3.3 Model Performance Metrics

In evaluating ML models, especially in imbalanced datasets, selecting the right performance metrics is crucial for accurately assessing model effectiveness and reliability. To address this, we have implemented stratified k-fold cross-validation in our study. This method is particularly beneficial for imbalanced datasets as it ensures that each fold of the validation process maintains the same proportion of classes as the original dataset. The use of stratified k-fold cross-validation aligns with best practices in the field, as it not only provides a more robust and reliable evaluation but also reflects real-world application scenarios, thereby enhancing the practical relevance of our findings (Berrar, 2018).

3.3.1 Confusion Matrix

The confusion matrix is a tabular representation that provides detailed insight into a model's classification performance. It distinguishes between correct predictions, depicted as True Positives (TP) and True Negatives (TN), and incorrect predictions, represented as False Positives (FP) and False Negatives (FN), also referred to as type I and type II

errors, respectively (Branco et al., 2015). The matrix is instrumental for diagnosing and refining the model's capabilities, as visualized in Table 3.1.

Table 3.1: Confusion matrix (Branco et al., 2015).

		Predicted	
		Positive	Negative
True	Positive	TP	FN
	Negative	FP	TN

Derived from these metrics, a set of performance indicators are computable and crucial for evaluating model efficacy. These are delineated from equations 3.1 through 3.6. Notably, equation 3.5 combines precision and recall into a singular metric, offering a harmonized evaluation that mitigates the respective trade-offs that are introduced by equations 3.1 to 3.4 that make it hard to choose a model based on one metric. Furthermore, equation 3.6 is tailored for scenarios involving imbalanced datasets, providing a more nuanced performance measure (Branco et al., 2015).

$$\text{accuracy} = \frac{TP + TN}{TP + FN + TN + FP} \tag{3.1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.3}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{3.4}$$

$$F1 = 2 \cdot \frac{TP}{2TP + FP + FN} \tag{3.5}$$

$$G_{mean} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \tag{3.6}$$

The confusion matrix offers a granular view of a model's performance, allowing us to identify areas of improvement, assess the impact of different decision thresholds, and make informed decisions regarding model deployment and optimization, which we have also seen being applied in our literature literature study.

3.3.2 Area under Receiver Operator Curve

The receiver operating characteristic curve represents a model's performance, plotting the True Positive Rate (Sensitivity) against the False Positive Rate (1-Specificity) as the decision threshold varies. AUC represents the area under this receiver operating characteristic curve, ranging from 0 to 1. AUC is a widely used metric for evaluating the discriminative power of a binary classification model (Branco et al., 2015). It is also widely used in the literature on credit risk models for retail lending. The metric comprehensively measures a model's ability to distinguish between positive and negative instances across various decision thresholds.

While the AUC is an invaluable metric for model evaluation, its application in imbalanced datasets requires a cautious approach due to its tendency to present an overly optimistic view, especially for the minority class (Branco et al., 2015). Comparing different models solely on AUC can be misleading, as it favors models that perform well on average, potentially overlooking those that excel at specific threshold settings. This can be seen in figure 3.4 where curves A and B perform better in different regions where appropriate thresholds can be chosen. Selecting a model purely under the AUC does not give the whole picture, and the ROC should be plotted to ensure the optimal model choice. Generally, a model with a higher AUC is considered more effective, with 0.5 denoting a performance no better than random chance and a score above 0.5 reflecting an ability to discriminate between classes more effectively. The pinnacle of performance is an AUC of 1.0, which indicates flawless discrimination, meaning the model can perfectly differentiate between positive and negative instances. It should be noted that even though the AUC metric has significant shortcomings in imbalanced data-sets it is still widely used within the literature as a standard metric for comparing (credit risk) models.

The AUC and the confusion matrix are indispensable tools for assessing the performance of machine learning models in time series classification and binary classification tasks. AUC provides a holistic measure of discrimination, while the confusion matrix offers a detailed breakdown of classification outcomes, helping us make informed decisions about model performance and improvements. One downside of the confusion matrix is the choice of a threshold. Within banking, it is reasonable to accept loans if the PD estimate is below 10%, while the confusion matrix typically uses the threshold of 50%. For this reason, the AUC does not require such business decisions to be made.

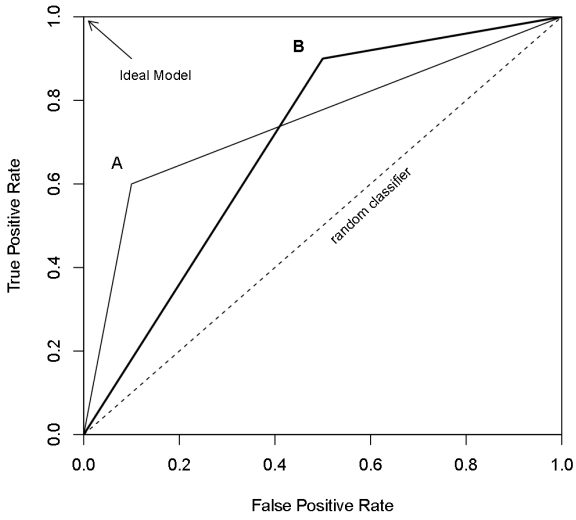


Figure 3.4: Receiver operator characteristics curve (Branco et al., 2015).

3.4 Explainability

SHAP values are a powerful tool for interpreting machine learning models. They provide a way to understand and quantify the contribution of individual features to a model's predictions on a local level (Adadi & Berrada, 2018). SHAP values are based on from cooperative game theory and offer a principled approach to attributing the prediction of a particular instance to its constituent features (Bücker et al., 2022). By calculating SHAP values for each feature, we can unravel the "black box" of complex models and gain insights into why a model makes specific predictions (de Lange et al., 2022). These values enhance model interpretability and facilitate feature selection, debugging, and model improvement, making them a valuable asset in machine learning interpretability.

While interval models generate discriminative features that can be explored using the SHAP method, and the approach outlined in Cabello et al. (2020) identifies and visualizes regions of interest directly on the time series, thereby enhancing model interpretability. The same cannot be said for deep learning approaches involving LSTM or ROCKET. These methods process raw time series data to predict default flags but do not offer a straightforward way to interpret or explain their predictions. This characteristic makes extracting meaningful explanations from these models an infeasible task. Although other methods like LIME exist, our study does not focus on multiple explainable AI techniques;

thus, we will not delve into them further.

3.5 Final Model

The predicted output from the superior TSC framework will serve as an input feature for the XGBoost model. We chose this method for two reasons. Namely, it is currently the best-performing algorithm in the literature, as highlighted by our literature study. Secondly, it is the method currently used by ING. To maintain a consistent benchmark for comparison, the feature set and hyperparameters, as architected by the original model developers, will be left unaltered. This approach directly evaluates our model enhancements relative to the established norms. The introduction of a preliminary classification model to precede the XGBoost framework is not a novel concept but is instead adapted from precedents in the literature, such as the methodology outlined by Gao et al. (2021), albeit the specific TSC algorithms selected for our implementation represent a pioneering step in this domain.

Chapter 4

Experimental Setup

This chapter delves into the comprehensive experimental setup of our study. We begin by providing an in-depth overview of the dataset utilized in this research. This includes a detailed description of the data attributes, the nature of the data, and its relevant characteristics, with particular emphasis on the target variable.

Following the data overview, the chapter discusses the data preprocessing methodologies employed in our study. We elaborate on the various steps taken to prepare the data for analysis. This crucial stage encompasses data cleaning, normalization, specific transformations. The quality and suitability of the pre-processing directly influence the performance and reliability of the TSC models.

Lastly, The utilized TSC models are sourced from the 'sktime' package (Löning et al., 2019), a well-regarded toolkit in the Python ecosystem for TSC.

4.1 Data

In the existing framework, Model Y requires specific criteria for customer transactions: [Classified criteria]. If a customer does not meet these criteria, Model X is employed instead as a standalone model. This approach primarily ensures sufficient data availability for feature creation and operates under the assumption that the customer primarily banks with this institution. For our study, we will adhere to these established guidelines. This allows for a direct comparison between the current operational model and our proposed enhancements. Consequently, our initial data will utilize the same anonymous customer data as the original model developers selected and filtered.

Regarding data sources, we have access to two key datasets for our research: a small univariate dataset detailing end-of-day account balances and a multivariate dataset with monthly account balance data. Appendix C shows the distribution shape and summary statistics. We will delve into these two datasets more comprehensively later in this chapter.

Target Variable

The core of our model is a binary target variable designed to indicate customer default status. It is assigned a value of 1 in instances of customer default and 0 if no default occured. The criteria for defining a default are stringent: a customer is considered to have defaulted if they are 90 days overdue on a loan or overdraft payment or under certain conditions such as early termination, bankruptcy, financial instability, fraud, refinancing, or restructuring. These default triggers are assessed within 12 months post-disbursement of a loan or overdraft.

Our dataset consists of XXX customers, with a default rate of X.XX% from 2017 to the end of 2021. The temporal evolution of the default rate and the customer base are depicted in Figure 4.1. A significant increase in the number of customers with loans is observable in 2020, which aligns with the moment overdrafts are included in the dataset. Overdrafts, while not traditionally classified as loans, are treated as such under legislative frameworks and are consequently incorporated into the model. This highlights the changing behavior of our data and suggests close monitoring while it is being used for business decisions.

An intriguing trend is the sharp decline in default rates observed towards the end of 2021. This phenomenon may be attributable to various factors, including the economic and social impacts of the COVID-19 pandemic. Additionally, Figure 4.2 reveals fluctuations in total defaults over time for term loans and overdrafts, highlighting the dynamic nature of financial behaviors and market conditions that could influence the model performance over time.

Time Interval Per Customer

A critical aspect of our data analysis involves understanding the temporal dimension. Ideally, data from all customers would span the same time interval. However, this uniformity is not present in our dataset. We have data from one year prior to each loan application date. Since these applications occurred at various times across different years, the time intervals for customers are not uniform.

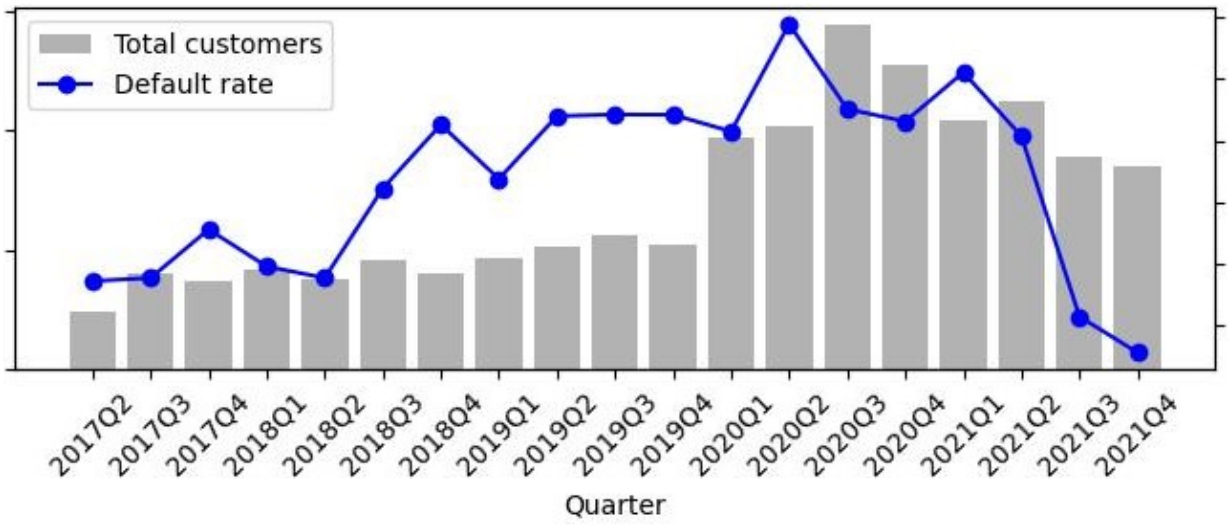


Figure 4.1: Default rate and number of customers per quarter.

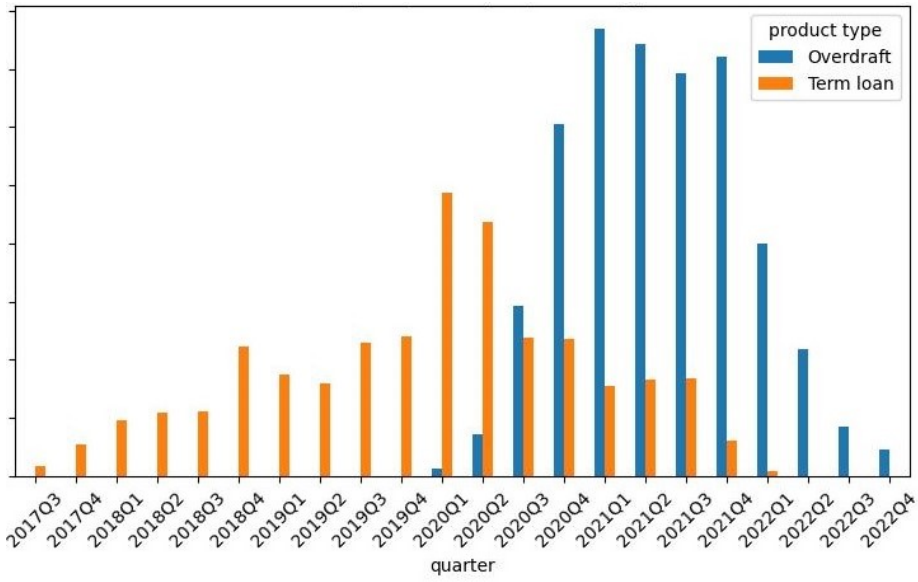


Figure 4.2: Defaults per quarter per product type.

Table 4.1: Example of current accounts data of monthly data

party_nbr	date	feature 1	...	feature 6
XXX	2020-01-01	619.21	...	13.15
XXX	2020-02-01	-4.00	...	-5.64
XXX	2020-03-01	430.90	...	3.07
XXX	2020-04-01	0.66	...	-5.63

Our model adopts a standardization approach for the time series data to mitigate this inconsistency. Instead of aligning the data with calendar dates, we index it based on a fixed 12-month period leading to the loan disbursement. This method ensures that all customer data are consistent and comparable, enhancing the accuracy of our analysis.

It is important to note that our dataset encompasses a one-year interval. An argument could be made for a two-year interval to observe each month twice, potentially capturing income seasonality more effectively. However, we contend that spending behavior from two years prior may not accurately reflect current financial behaviors. Consequently, while a one-year interval may limit our ability to observe long-term seasonal trends, it likely offers a more relevant depiction of recent financial activities.

Multivariate Monthly Balance Data

The multivariate monthly balance dataset encompasses seven summary statistics calculated monthly for each customer's current account. These statistics are detailed in Table 4.2. A key advantage of using summary statistics over daily data, discussed in the subsequent section, is the reduction in data dimensionality. The daily dataset's 365 data points are condensed into 84 data points per customer ($months \times nbr\_features$), significantly reducing the input dataset.

However, it is essential to recognize that the computational complexity of algorithms scales differently with the length of the time series and the number of time series per sample.

Table 4.2: Multivariate Monthly Time Series Data

Time series feature	Description
Classified	Classified

A significant challenge is the account-specific nature of the data. Customers may have

multiple current accounts for different purposes, such as shared or individual spending accounts. The current model employs heuristics to determine which account to use for feature calculation. The variability in account opening and closing times also leads to incomplete data. We further refine our dataset to include only customers with a single current account and at least 12 months of data. This decision is based on the fact that such customers constitute the majority of the data (66.6%) and we are certain that this is used as a current account. However, this approach introduces a limitation by excluding customers with multiple accounts.

The final dataset comprises XXX individuals, with a default rate of X.XX%. This limitation underscores an opportunity for future research to enhance the model's robustness and applicability. Incorporating individuals with multiple accounts and varying account durations could expand the model's scope and improve its predictive accuracy. Table 4.1 presents an illustrative monthly current account data example.

Univariate End-Of-Day Balance Data

The univariate dataset in our study consists of daily end-of-day balance data for customers' current accounts. This dataset is not as comprehensive as the multivariate one because it is not used within the current model. In aligning with the selection criteria of the multivariate data, we included only those customers with a continuous record of at least 365 days. This rigorous selection process results in a dataset comprising 7,905 customers, with a default rate of 36 %—significantly smaller than the multivariate dataset. But this smaller sample also helps us in reducing our dataset. The 7 905 persons have 365 datapoints, resulting in 2.8 million datapoints ($Nbr\_customers \times nbr\_days$).

The primary aim of using this univariate dataset is to explore the potential for enhancing our model's accuracy by leveraging more granular data. While promising in terms of model refinement, this approach introduces certain challenges. The increased data granularity leads to higher computational demands for training the models. Processing and analyzing a detailed dataset of this nature requires substantial computational resources, posing limitations on our methodology.

4.2 Data Preprocessing

We anticipate the need to downscale the training sample to mitigate these computational challenges in our model. Additionally, the selection of hyper-parameters for the model

will be strategically tailored to simplify the model's complexity. These adjustments are crucial for balancing model precision and computational efficiency. Despite the constraints, the univariate dataset offers valuable insights into customer behavior patterns at a more detailed level, providing a nuanced perspective that complements the broader analysis conducted with the multivariate data.

4.2.1 Handling Imbalanced Datasets

Imbalanced datasets are a prevalent challenge in machine learning, impacting various fields such as face recognition, software engineering, and financial modeling (Fernandez et al., 2018). In addressing imbalanced datasets, the primary objective is to balance the trade-off between enhancing predictions for the minority class and controlling false positives (Fernandez et al., 2018). To this end, sampling techniques have emerged as prominent solutions, predominantly categorized into undersampling and oversampling.

Undersampling reduces the size of the majority class to balance the class distribution. Its benefit lies in creating a compact and balanced training set, thus reducing the computational cost of the learning stage (Fernandez et al., 2018). However, it can increase the classifier's variance and may discard useful examples, potentially affecting the classifier's generalization ability, especially in high imbalance scenarios. Conversely, oversampling increases the minority class size, with methods ranging from simple replication to more complex techniques like the SMOTE (Fernandez et al., 2018).

For our study, undersampling was chosen over other methods like oversampling and SMOTE. Our dataset's specific characteristics and research objectives informed this decision, prioritizing computational efficiency and model performance while minimizing overfitting risks. In the context of our large-scale, high-dimensional data, undersampling offered an optimal balance between simplicity and effectiveness.

4.2.2 Data Normalization

Data normalization is a crucial preprocessing step in machine learning, especially for models sensitive to the scale of input data. Our study employed different normalization strategies for the models under consideration based on their inherent characteristics and requirements.

Data normalization is unnecessary for the CIF model. The CIF model, leveraging a random forest algorithm that operates on features calculated on canonical intervals, is

inherently robust to variations in the scale and distribution of input data. Therefore, the CIF model processes the raw time series data without normalization.

However, normalization is a standard and essential practice for the ROCKET, LSTM and shapelet models. These models are sensitive to the scale of the input data, require normalized inputs to function effectively. We have chosen to implement Z-score normalization for these models, as indicated by Equation 4.1:

$$X_{\text{normalized}} = \frac{X - \mu}{\sigma} \tag{4.1}$$

Here, $X_{\text{normalized}}$ represents the normalized data, X is the original time series, μ is the mean of the time series, and σ is the standard deviation.

Moreover, we split the normalization process into two distinct approaches: absolute and relative normalization. Relative normalization, which involves normalizing data per customer, is particularly interesting in our context. This approach is necessitated by customers with significantly high incomes, which could otherwise skew the overall mean (μ) of the dataset. Such a method aligns with the findings of Kvamme et al. (2018), who noted the trade-off inherent in this approach: while it helps in managing outliers, it limits the deep learning models only to discern relative patterns within the data.

In summary, our data normalization strategy is tailored to the specific requirements of each model used in our study. While the CIF model works effectively with raw data, the ROCKET, LSTM and shapelet models benefit from the Z-score normalization technique, with a special emphasis on relative normalization to accommodate the unique distribution of our dataset.

4.2.3 Out-Of-Sample Validation

To ensure the robustness and reliability of our model, we employed a 10-fold stratified cross-validation technique. This method involves partitioning the dataset into ten equally sized folds or subsets of data. In each iteration of the validation process, one fold is used as the test set, and the remaining nine folds are used as the training set. This process is repeated ten times, with each fold serving as the test set exactly once.

Stratified cross-validation ensures that each fold represents the whole well by maintaining the same proportion of classes as in the original dataset. This approach is important in our study due to the imbalanced nature of our dataset, with the default class being significantly underrepresented.

This rigorous validation method allows us to assess the performance and generalizability of our model accurately. It also helps mitigate overfitting, ensuring that our model's predictions are not merely artifacts of the particular sample of data used for training. The entire training, testing, and validating process is summarized in Algorithm 2.

4.2.4 Hyper-Parameter Tuning

Hyper-parameter tuning is critical to building robust ML models. Hyper-parameters, distinct from model parameters, are not learned from the data but are set before the training process. These parameters play a pivotal role in controlling the behavior of the learning algorithm and can significantly impact the model's performance.

Proper selection of hyper-parameters is essential. Setting them too high can lead to overfitting, where the model becomes overly complex and performs well on the training data but fails to generalize to new, unseen data. Conversely, setting them too low may result in an underfitted model, lacking the complexity to capture the underlying patterns in the data. Additionally, inappropriate hyper-parameter values can lead to unnecessarily long training times, which is especially critical in scenarios dealing with large datasets.

The final set of hyper-parameters chosen for each model was based on a balance between accuracy and computational efficiency. A list with all the hyper-parameters is provided in Appendix C.

4.3 Final Model

In the final model, the study identifies the most effective TSC model by utilizing the AUC metric, a widely recognized standard for measuring the accuracy of predictive models. The model with the highest average AUC over the cross folds is selected as the superior model for this study.

An intriguing aspect of the methodology involves revisiting the previously excluded data points - those rejected during the undersampling process. The model's performance on these samples offers valuable insights, especially since undersampling is commonly employed to address class imbalance in datasets. By estimating these samples with the trained TSC, the study sheds light on the model's effectiveness across a broader data spectrum.

Moreover, the PD for each customer is computed and recorded. The current model used by ING is trained on the selected sample. This training is conducted twice: once with the

Algorithm 2 Pseudocode for training, testing, and validating TSC models

```
input: Preprocessed datasets (multivariate, univariate), hyper-parameters
FOR each dataset in [multivariate, univariate]:
    undersample dataset until balanced
    Perform 10-fold stratified cross-validation:
    FOR each fold:
        1. Train the CIF model without normalization
        2. Normalize the data:
           a. Train the LSTM-FCN model on normalized data
           b. Train the ROCKET model on normalized data
           c. Train Shapelet model on normalized data
        3. Relatively normalize the data:
           a. Train the LSTM-FCN model on relatively normalized data
           b. Train the ROCKET model on relatively normalized data
        4. Test the models on current fold and record performance metrics
    END FOR
    Aggregate and analyze the performance metrics across all folds
END FOR
```

inclusion of the PD estimates derived from the TSC model as a feature, and once without these estimates as shown in Figure 3.2. This comparative approach aims to evaluate the added value, if any, provided by the PD estimates from the TSC model.

This comparison is vital to understanding the incremental benefit of integrating the TSC model's insights into the existing model model. By evaluating the performance of the ING model with and without the PD estimates, the study can demonstrate the practical implications of the TSC model in a real-world banking environment.

4.4 Explainability

The explainability of our models within our framework, particularly the best-performing TSC model, is investigated whenever a feasible method is available. Such exploration offers deeper insights into our model choices and enhances our understanding of the data. As we highlighted in our literature study in Chapter 2, explainability is often found to be lacking in most research. In response, we adopt a more holistic approach by incorporating explain-

ability into our framework. This strategy extends beyond merely assessing performance; it encompasses considerations of usability, addressing the need for models that stakeholders can trust. However, it should be noted that our focus is not on expanding the research area of explainable AI.

The final model is examined using mean absolute SHAP values to evaluate the contribution of the TSC-PD estimate. This technique highlights the importance of our newly introduced feature in the XGBoost framework, giving us insight into whether it is contributing.

Chapter 5

Results

This chapter presents the performance of evaluating the TSC model on both multivariate and univariate datasets. We will choose the best-performing model and examine its interpretability. Then, we train the XGBoost, both with and without the PD estimate as a feature, and evaluate its performance. This chapter corresponds to the demonstration step in the DSRM process.

5.1 Time Series Classification Model Results

This section delves into all the results of the TSC models. Understanding the creation steps of our model is essential, so we summarize the experiment procedures as described in Chapters 3 and 4. Initially, we selected customers with at least one year of historical balance information, marking a deviation from the currently used model. The existing model calculates features on data without requiring a minimum one-year historical balance, making our model more stringent. We then focused only on customers with a single current account, disregarding those multiple current accounts and removing the savings accounts. This approach is a notable limitation of our model as it needs to provide a complete customer profile. However, the current model excludes savings accounts and applies predefined rules for feature calculation on selected current accounts. We have created two datasets: 1) a multivariate monthly dataset and 2) a univariate end-of-day current account dataset. We undersampled the data in each set to achieve balanced datasets. We then trained and tested these datasets using stratified 10-fold cross-validation, saving the results from each test fold. For models incorporating neural networks, we applied both absolute and relative

z-normalization to the data. In summary, we introduced two additional assumptions in our model, deviating from the model currently in use:

- The customer has at least 12 months' worth of historical balance data of the current account
- the customer has exactly one current account

These assumptions allow for cleaner data, providing a clearer insight into customers' spending behavior. We have not tested the impact of applying zero padding to missing data. It would also be interesting to merge all the current accounts into one time series or even combine all current and savings accounts into a single time series to evaluate the model's performance. Additionally, all banks can inquire the end-of-day balance data of the current account from customers outside the bank using the PSD2 system.

5.1.1 Multivariate

The CIF model leads in performance with an AUC of 0.68, surpassing all other models. The LSTM model also demonstrates notable results on normalized data, achieving an AUC of 0.64 and ranking among the top performers. However, models trained on relatively normalized data, such as ROCKET and LSTM, exhibit bad performance differences, with AUC scores of 0.58 each. In contrast, the ROCKET model on normalized data achieves a slightly higher AUC of 0.59. Overall, the performance on the multivariate dataset is modest. For comparison, ING's current model Y achieves a significantly higher AUC of 0.XX. Despite the advances in deep learning methods, including LSTM, they fall short in performance compared to the other techniques on our dataset.

The CIF model's performance is notable as it closely resembles the current industry-standard models. Its robustness to outliers is due to calculating summary statistics on intervals and creating numerous intervals. The LSTM model shows a performance gain, aligning with expectations based on the Gao et al. (2021) study. However, this performance diminishes on relatively normalized data, suggesting that this relative normalization technique removes critical information as expected, which is important for the model. The ROCKET model's performance on both relatively normalized and just normalized data is also subpar, leading us to conclude that it is not well-suited for multivariate monthly datasets. We argue that the prevalence of outliers in the data, as shown in the distribution plots in Appendix C, complicates the performance of deep learning methods.

5.1.2 Univariate

The CIF model again outperforms deep learning methods, achieving an AUC of 0.80. The shapelet-based approach also shows excellent performance on normalized data, with an AUC of 0.79. These results are particularly impressive [Classified], as they are achieved using only end-of-day balance data.

Among the other models, the LSTM model on normalized data scores the highest with an AUC of 0.72, followed by its performance on relatively normalized data with an AUC of 0.70. The ROCKET model shows a slightly lower performance with AUC scores of 0.67 on normalized data and 0.68 on relatively normalized data.

The CIF model's robustness is evident in this application, and the shapelet's strong performance, despite lengthy training times, suggests that the two classes are distinguishable by their shapes. The LSTM model performs better than the ROCKET model, consistent with the observations in the multivariate case. Models trained on relative data underperform in both univariate and multivariate settings, indicating that relative normalization may not be the best approach for handling this data. The high AUC score of 0.80 is indicative of strong model performance, although it should be noted that defaults can occur unexpectedly due to unforeseen circumstances like illness or job loss.

We conclude that aggregating data into a monthly format significantly diminishes valuable information, thereby weakening the performance of TSC models. This result is logical, as these models are designed for time series data, and the aggregation process transforms these into statistical monthly features.

The underperformance of deep learning models in both multivariate and univariate datasets can be attributed to the significant number of outliers, as observed in our data exploration and illustrated in Appendix C. The distribution x-axis had to be scaled with log to visualize the plots, indicating the high outliers. These outliers alter the mean and standard deviation, affecting most customers' time series. Applying relative normalization resolves this issue by facilitating comparisons of relative movements in end-of-day balances, but it sacrifices the ability to make absolute comparisons. Alternatively, removing outliers is an option, but this disregards customers who might be significant outliers.

Enhancing the model could involve adding more time series data to the end-of-day series. For future research, we propose aggregating all current accounts into one time series, all savings accounts into another, and creating separate daily credited and debited amount time series, which can then be fed into multiple TSC models.

Based on these results, our continued analysis will focus on CIF and shapelets using the univariate end-of-day balance data.

5.1.3 Computation Time

This section presents a comparative analysis of the computation times from a theoretical and empirical perspective of the CIF and the Shapelet algorithms.

Theoretical Time Complexity

The time complexity of an algorithm can be measured in the amount of time it takes to run it as a function of the length of the input, called the big O notation.

The time complexity of the CIF algorithm, as derived from the work of Middlehurst et al. (2020), incorporates several key parameters: the number of trees generated r, the number of intervals per tree k, the number of cases per tree n, the number of attributes subsampled per tree a, and the computation of Catch-22 features $m^{1.16}$. These parameters combine to yield a total time complexity for the CIF model of $O(rknam^{1.16})$. This formulation indicates that the computation time increases linearly with the number of trees, intervals, cases, and attributes. However, the computation of the Catch-22 features, which scales with m to the power of 1.16, introduces a superlinear factor into the complexity. It is important to note that the algorithm does not compute all features for each interval. Instead, it selects a random subset of features for computation, which mitigates the computational burden as detailed by Middlehurst et al. (2020). Given that it scales linearly with the input size the algorithm is desirable for practical applications.

The time complexity of the Shapelet algorithm, as calculated by Bagnall et al. (2017), is $O(p^2q^4)$, where p is the number of time series and q is the length of each series. This significant computational complexity arises from the exhaustive search for the optimal shapelets within the dataset. Due to this complexity, considerable research is in developing methods to approximate the shapelet discovery process. One such approach involves using heuristics to find near-optimal shapelets, reducing computational demands while retaining the quality of the identified shapelets. An example of this line of research can be found in the work of Grabocka et al. (2014), which proposes a heuristic-based method for shapelet discovery. We used the same approach as Grabocka et al. (2014).

Empirical Training Time

From a practical standpoint, we have analyzed the training times of various algorithms on our univariate end-of-day training set, with the results displayed in Figure 5.1. The visualization illustrates the fast computation time of the ROCKET algorithm. In stark

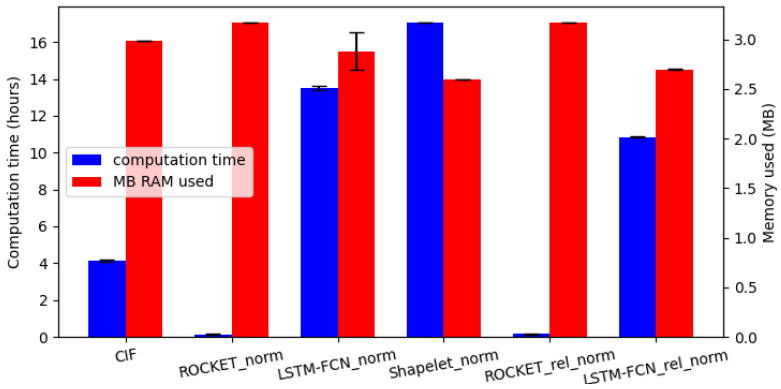


Figure 5.1: Average Computation time and RAM used in the univariate cross-validation process.

contrast, the shapelet algorithm exhibits the longest training duration, underscoring the implications of its time complexity on practical application. The CIF algorithm, while not as swift as ROCKET, demonstrates a more moderate computation time, offering a viable compromise between speed and the granularity of interval analysis. By constructing trees in parallel, significant reductions in computation time can be achieved, which is a testament to the CIF algorithm's adaptability to multi-core processing environments and practicality in business.

Imbalanced Dataset

To optimize computation time and enhance model performance, we balanced our dataset, a strategy supported by existing literature. However, it's notable that most credit risk model studies do not employ dataset balancing (Dastile et al., 2020). To explore the effects of this, we trained the CIF model on an unbalanced dataset, considering the limitations of our computational resources. Consequently, we had to reduce the model's complexity, which involved halving the number of trees in the CIF model.

The test yielded an average AUC of 0.71 (std=0.02), a considerable decrease compared to the balanced dataset. This decline in performance suggests that using an imbalanced dataset impacts the model's effectiveness. Additionally, we believe that the further reduction in the AUC score is partly due to the simplified model architecture, which included reducing the number of trees from 200 to 100 to maintain computational feasibility.

5.1.4 Explainability

CIF Algorithm and Its Explainability

The CIF algorithm, which integrates a random forest model with an interval-based approach, enhances interpretability through methods such as SHAP. This combination provides a nuanced method for interpreting time series data, enabling ways to understand individual sample-level decisions. To illustrate how the model makes decisions and which features it prioritizes, we have visualized the decision process using a Partial Dependence Plot. This approach is demonstrated in Figure 5.2, which displays the top 10 features across the entire interval in the classification process.

Central to the CIF algorithm's performance is the high information gain of the periodicity of the Wang feature.[1] The feature effectively measures time series periodicity, distinguishing series with consistent patterns from those with rapid fluctuations and noise. Higher values are assigned to series with more significant periodicity, while lower values indicate more irregularities in the time series. The Wang feature thus plays an important role in analyzing the periodic nature and consistency of customers' financial behaviors, offering a sophisticated lens for observing time-related consistency. For a more personalized model explanation, SHAP can be exploited on the random forest, demonstrating the individual contribution of each feature to the decision-making process.

In conclusion, the CIF algorithm, illustrated in Figure 5.2, effectively delivers a holistic perspective on utilizing the entire interval in feature analysis. We observe a slight increase in information gain in the year's second half, but it is essential to consider the entire period for a comprehensive analysis. For a detailed understanding of all the Catch-22 features, one can refer to Appendix D. The Wang feature emerges as one of the most crucial elements, providing an efficient method for credit risk modeling on the end-of-day current account data. This prominence of the Wang feature underscores the potency of both the CIF algorithm and the Catch-22 methods in the context of credit risk models, highlighting their effectiveness in capturing essential dynamics in financial data.

Imbalanced Dataset

To optimize computation time and enhance model performance, we balanced our dataset, a strategy supported by existing literature. However, it is notable that most credit risk model studies do not employ dataset balancing (Dastile et al., 2020). To explore the effects of this,

[1] Name in the legend: $PD\_PeriodicityWang\_th0\_1$ of Figure 5.2, as presented by X. Wang et al. (2007)

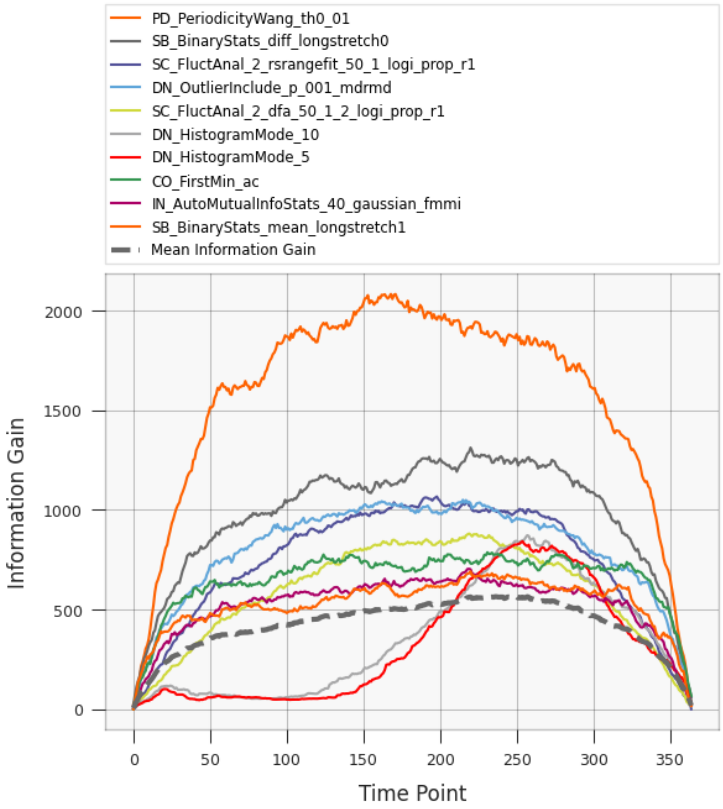


Figure 5.2: Partial dependence plot of the top 10 most important features from the Catch-22 features in the trained CIF model on the univariate end-of-day dataset.

we trained the CIF model on an unbalanced dataset, taking into account the limitations of our computational resources. Consequently, we had to reduce the model's complexity, which involved halving the number of trees in the CIF model.

The test yielded an average AUC of 0.71 (std=0.02), a considerable decrease compared to the balanced dataset. This decline in performance suggests that using an imbalanced dataset impacts the model's effectiveness. Additionally, we believe that the further reduction in the AUC score is partly due to the simplified model architecture, which included reducing the number of trees from 200 to 100 to maintain computational feasibility.

Shapelet Algorithm and Its Explainability

Unlike the CIF algorithm, the Shapelet algorithm is distinguished by its exceptional explainability, largely attributed to its utilization of shapelets. Shapelets can be directly plotted over a customer's end-of-day balance data, allowing for a clear graphical representation of divergence between lines, if the loan is not approved. This approach leverages data patterns directly for classification, fostering transparency and simplifying interpretation.

The found shapelets, as illustrated in Figure 5.3, all identified an upward trend in the current account. The weekly and monthly shapelets still show a noisy shape, whereas the quarterly and half-yearly shapelets follow linear trends. This upward trajectory aligns well with business insights, particularly in loan approval scenarios, where a positive income trend in a current account suggests a healthy financial status. The algorithm calculates the distance between these desirable upward shapelets and the actual time series, utilizing this measurement as a feature.

The shapelets identified and illustrated in Figure 5.3 consistently indicate an upward trend in the current account data. The weekly and monthly shapelets exhibit some variability, contrasting the quarterly and half-yearly shapelets, demonstrating more linear trends. In loan approval, this upward trend is insightful as it suggests a stable and growing financial status in the current account. The algorithm capitalizes on this insight by calculating the distance between these preferred upward-trending shapelets and the actual time series data at the best fitting position, employing this measurement as a key feature for analytical purposes.

A key advantage of shapelets is their autonomous determination of the optimal fitting period, which does not require pre-set instructions for the duration of the time series, such as one or three months. This algorithm feature effectively tackles the challenges outlined in Chapter 4.1 - 'Time Interval Per Customer,' enabling shapelets to adjust to the most suitable period. However, we observed an unusual pattern in the yearly shapelet, with an unexpected flattening in income. To maintain the efficiency of the shapelet's automatic timeframe adjustment, we advise against using shapelets longer than half the length of the time series. This strategy ensures preserving the shapelets' inherent strength in adapting to varied time frames.

The strength of the Shapelet algorithm lies in its ability to identify and create complex shapelet shapes, providing a distinct advantage in scenarios where pattern recognition is paramount. Although there were no complex shapes found in our dataset. Compared to the CIF algorithm, which offers a macro-level understanding, the Shapelet algorithm's direct interpretability allows for a more precise and insightful analysis of time series data.

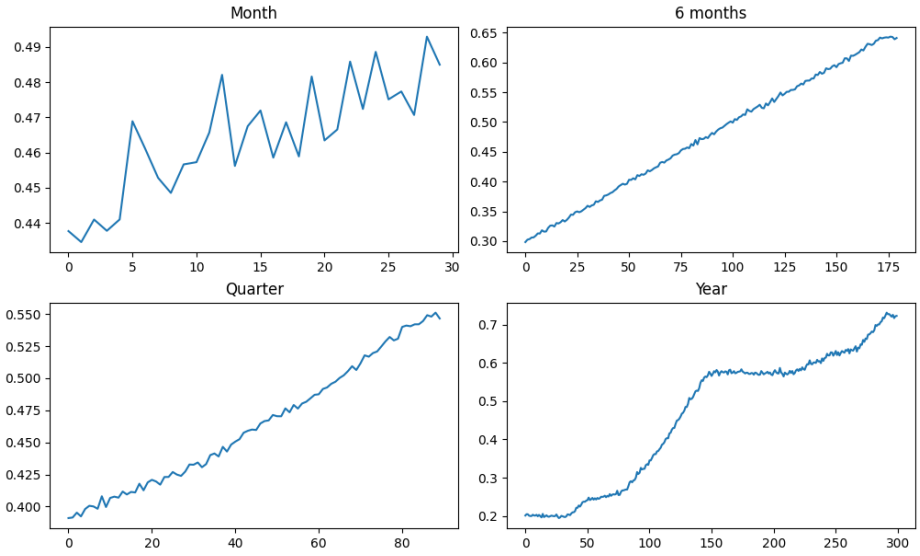


Figure 5.3: Top found normalized shapelets for different time intervals.

Both the Shapelet and CIF algorithms contribute various methods to make the models explainable and, consequently, more trustworthy. This makes them particularly suitable for applications where transparency is a critical factor. Despite their different approaches and strengths, both models offer effective pathways for ensuring explainability in complex data environments.

5.2 Final Model Results

The final XGBoost model is trained with and without the PD estimate of our CIF model as a feature. Without our estimate, the current model gives us a benchmark with the same dataset. It should be noted that there is a risk of overfitting the data since we are reusing the same samples used to train the TSC model and give this as an input for the combined model. The models are again tested on 10-fold cross-validation, and the out-of-sample fold is used to calculate the performance metrics.

The combined model shows a performance gain to an AUC of 0.81 compared to our CIF model with an AUC of 0.79. Model Y has an AUC of 0.77, while the model developers claim an out-of-sample AUC of 0.XX. The under-performance of model Y is likely caused by data reduction. We applied more stringent filtering of persons having just one current account and at least 365 data points, resulting in a sample size drop from 180k to 10k

samples, which damaged the performance of model Y. The combined model Y does show an improvement with the additional PD estimate as a feature.

Even though this model stacking approach where the TSC prediction is used as a feature in the main model was a logical choice for deep learning models, as it is also done in the research of Gao et al. (2021) because these models do not create features. The CIF model does calculate features on different time intervals. a continuation in the research could be to fully migrate the features to the XGBoost, simplifying the model architecture and enabling the XGBoost model to find relationships between the features instead of the PD estimate.

5.2.1 Random Dataset Validation

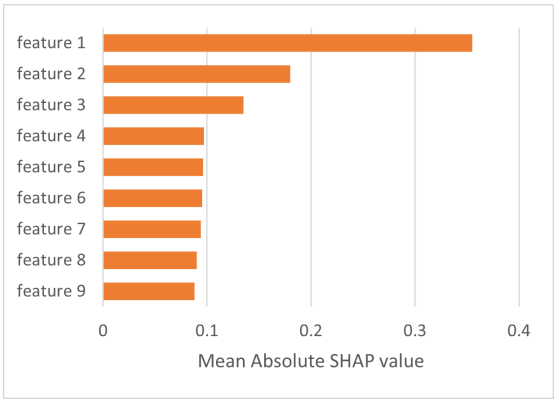
We conducted training and testing using randomly generated data to verify the unbiased nature of our model's estimates. This process involved the creation of time series data, synthesized by drawing from a normal distribution with parameters mirroring the mean and standard deviation of the per-person current account in our samples. The target variable was also generated to reflect the portfolio's default rate change (X.XX%). This approach aimed to mirror real-world variability and unpredictability.

Following this, we applied the same 10-fold cross-validation procedure as with our original model. The combined model's AUC for predictions on this random dataset was 0.503 for the balanced dataset and 0.505 for the unbalanced dataset. These results, hovering around the 0.5 mark, indicate that the model's predictions on the random dataset align closely with what would be expected by chance (Mathai et al., 2020). Such an outcome reinforces the conclusion that our model's design is inherently unbiased. Furthermore, it suggests that our dataset is sufficiently comprehensive to enable the model to converge to an estimate reflective of random chance, thereby supporting the validity of our modeling approach.

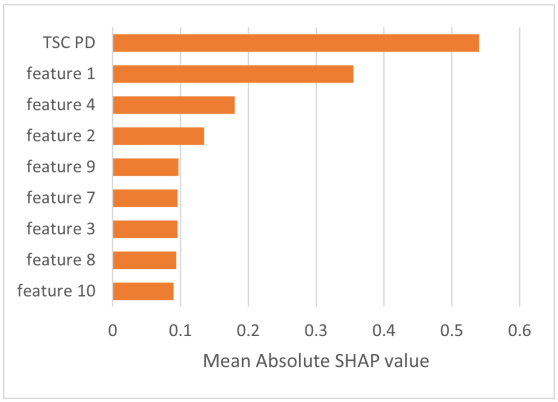
5.2.2 Explainability

Figure 5.4 presents the top 9 features in our model, ranked by their mean absolute SHAP values. The PD estimate from the TSC model is identified as the most influential feature, with an absolute mean SHAP value of 0.35, indicating a substantial impact on the model's predictions.

The analysis, however, underscores a challenge in interpretability due to the model's stacked structure. The significant role of the PD estimate in the TSC model is clear, but



(a) Model Y without TSC PD estimate



(b) Model Y with TSC PD estimate

Figure 5.4: Absolute SHAP value contribution of the top 10 features in the final model with and without our TSC estimate

its specific impact remains less transparent, suggesting a potential benefit in integrating the TSC model directly into the XGBoost framework. Such integration would enhance explainability by simplifying the feature attribution process, making it easier to understand and justify the model's predictions.

Moreover, it's important to recognize that while absolute mean SHAP values quantify the average impact of features, they do not indicate the direction of this impact. The high value of the PD estimate signifies its strong influence, but further analysis of non-absolute SHAP values is necessary to determine the nature of this influence.

Chapter 6

Conclusion

This research embarked on an innovative journey to harness TSC algorithms for enhancing credit risk models, with a special focus on the historical balance data of retail customers. The origin of our study stemmed from a notable main research problem in the field: the creation of features on time series data is labor intensive and might overlook crucial insight in time series data. This challenge shaped our central research question:

"How can time series classification algorithms on historical balance data of retail customers enhance credit risk models' discriminatory power?"

Our quest to address this question revealed a notable void in existing academic literature: there is no current literature researching the synergy between TSC and credit risk modeling for handling time series data. This discovery positioned our study at the forefront of bridging this gap, representing a novel and significant stride in the field. We delved into an array of TSC methodologies, encompassing deep learning, shapelets, and interval-based approaches. Each method underwent a rigorous assessment to evaluate its capacity to refine the accuracy and enhance the performance of credit risk predictions. The outcomes of these evaluations not only underscore the viability of TSC techniques in this context but also mark a groundbreaking advancement in the realm of financial risk assessment.

The most significant achievement of our research was the successful incorporation of the optimal TSC-based PD prediction method into a comprehensive credit risk model. This method, when used alongside traditional feature sets in an XGBoost model, demonstrated a significant enhancement in the discriminatory power of credit risk models. Our

findings not only highlight the potential of TSC techniques in credit risk assessment but also set a precedent for future research in this domain, potentially revolutionizing credit risk management strategies in the financial industry.

In our study, we investigated various TSC methodologies, including CIF, shapelet-based models, and two advanced deep learning approaches. Both standalone TSC models, namely the shapelet and CIF, demonstrated promising results, effectively competing with the existing models prevalent in the industry. However, it is important to acknowledge the interpretability aspect of the CIF model. Given that it processes a feature set derived from hundreds of intervals.

The difference in interpretability becomes particularly evident when we compare these TSC models with the current industry-standard model, which are typically developed with business-centric features. In this context, the shapelet-based model emerges as notably more interpretable. Plotting the identified shapelets with the customers' end-of-day balance data facilitates a more direct and intuitive understanding of the model's decision-making process. This attribute of the shapelet-based approach underscores its potential for providing clear, actionable insights in the domain of credit risk assessment.

An important observation from our research is the limited efficacy of deep learning classification methods when applied to both the end-of-day balance data and the multivariate monthly dataset. We attribute this underperformance to numerous outliers in retail customers' balance data. These outliers pose significant challenges in data normalization, a prerequisite for deep learning models to function effectively. The normalization process, while essential, can be significantly hampered by outliers, leading to issues in model convergence and overall performance.

Moreover, a critical limitation of deep learning models in the context of credit risk modeling is their inherent lack of transparency at an individual decision level. This lack of transparency in decision-making processes is a significant drawback, particularly in credit risk assessment, where understanding the rationale behind each credit decision is crucial for regulatory compliance and gaining stakeholder trust.

Nonetheless, Our research introduces a novel approach that connects TSC with credit risk models, making a valuable contribution to the evolving literature on credit risk modeling. Our quantitative testing, supported by a comprehensive review of current literature, confirms that TSC models effectively can discriminate between default and non-default retail customers.

6.1 Limitations and Future Research

6.1.1 Limitations

Our research journey, while fruitful, encountered several limitations that are important to acknowledge. These challenges impacted our study and provided valuable insights into the complexities of working with TSC models and large datasets.

- Limited computational resources somewhat constrained item our capabilities. This limitation restricted our ability to explore the full spectrum of TSC models, potentially leaving out some promising approaches.
- Each TSC method we examined required different amounts of computation time. This discrepancy posed a challenge in creating a fair comparison framework, as balancing the sample size and model complexity to equalize training times would make the comparison fairer.
- The dataset we utilized, specifically the univariate dataset, was incomplete compared to the multivariate dataset. Consequently, the customers in the training and testing of the models were different between these two.

6.1.2 Future Research

The scholarly dimension of our study presents numerous opportunities to broaden the research spectrum. Our investigations have unveiled promising avenues for integrating TSC with credit risk modeling frameworks. To further refine and extend the scope of our research, we propose the following recommendations:

- Investigate the inclusion of various account types (e.g., savings accounts) in the model to present a more holistic financial profile of customers.
- Consolidate how to handle clients with more than one current account.
- Distinguish between overdrafts and term loan defaults and explore their performance in the model.
- Examine the impact of extending the dataset beyond the COVID-19 period to assess the model's robustness in varying economic climates.

- Consider the integration of additional TSC algorithms that were not covered in this study, broadening the scope of algorithmic comparison.
- investigate the added benefits of using TSC in other domains in finance, such as early warning systems and fraud detection.

6.1.3 Operational Recommendations for Business Implementation

This research not only holds academic value but also exhibits significant business potential. It offers avenues for enhancing model precision, streamlining model structures, and consolidating data sources. To effectively translate these research insights into business practice, we suggest the following operational strategies:

- Develop an optimal interval search strategy for the current model to maximize its predictive accuracy. Putting more emphasis on the interval choice.
- Add the Catch-22 and shapelet features to the current model feature set. Also, discuss the acceptability of these features with stakeholders. Specifically, look into the application of the periodicity of Wang feature for intervals chosen by ING.
- Create a multivariate end-of-day dataset using the same information as the current model features (i.e., debit/credit, savings, etc.) to equalize the data of the TSC models and the current model.
- Analyze the relative positions of identified shapelets in classification algorithms to enrich the model's interpretative power and gain more insight into the data.

This research has taken significant strides in integrating TSC models with credit risk assessment, offering new perspectives and methodologies in the field. While we have unveiled promising avenues, the journey towards fully understanding and optimizing these models continues. The recommendations and limitations highlighted in this study serve as a roadmap for future research, steering toward a more inclusive, accurate, and comprehensive credit risk modeling framework.

References

Abellán, J., & Castellano, J. G. (2017). A comparative study on base classifiers in ensemble methods for credit scoring. *Expert Systems with Applications*, *73*, 1–10. https://doi.org/10.1016/j.eswa.2016.12.020

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI) [Conference Name: IEEE Access]. *IEEE Access*, *6*, 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

Addo, P. M., Guegan, D., & Hassani, B. (2018). Credit risk analysis using machine and deep learning models [Number: 2 Publisher: Multidisciplinary Digital Publishing Institute]. *Risks*, *6*(2), 38. https://doi.org/10.3390/risks6020038

Alam, T. M., Shaukat, K., Hameed, I. A., Luo, S., Sarwar, M. U., Shabbir, S., Li, J., & Khushi, M. (2020). An investigation of credit card default prediction in the imbalanced datasets. *IEEE Access*, *8*, 201173–201198. https://doi.org/10.1109/ACCESS.2020.3033784

Antonio Bahillo, J., Kremer, A., & kristensen, I. (2016). *The value in digitally transforming credit risk management — McKinsey*. Retrieved October 2, 2023, from https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/the-value-in-digitally-transforming-credit-risk-management

Bagnall, A., Lines, J., Bostrom, A., Large, J., & Keogh, E. (2017). The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, *31*(3), 606–660. https://doi.org/10.1007/s10618-016-0483-9

Berrar, D. (2018, January 1). Cross-validation. https://doi.org/10.1016/B978-0-12-809633-8.20349-X

Bostrom, A., & Bagnall, A. (2017). Binary shapelet transform for multiclass time series classification. In A. Hameurlain, J. Küng, R. Wagner, S. Madria, & T. Hara (Eds.), *Transactions on large-scale data- and knowledge-centered systems XXXII: Special issue on big data analytics and knowledge discovery* (pp. 24–46). Springer. https://doi.org/10.1007/978-3-662-55608-5_2

Branco, P., Torgo, L., & Ribeiro, R. (2015, May 13). A survey of predictive modelling under imbalanced distributions. https://doi.org/10.48550/arXiv.1505.01658

Brygała, M. (2022). Consumer bankruptcy prediction using balanced and imbalanced data [Publisher: MDPI AG]. *Risks*, *10*(24), 24. https://doi.org/10.3390/risks10020024

Bücker, M., Szepannek, G., Gosiewska, A., & Biecek, P. (2022). Transparency, auditability, and explainability of machine learning models in credit scoring [Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01605682.2021.1922098]. *Journal of the Operational Research Society*, *73*(1), 70–90. https://doi.org/10.1080/01605682.2021.1922098

Cabello, N., Naghizade, E., Qi, J., & Kulik, L. (2020). Fast and accurate time series classification through supervised interval search [ISSN: 2374-8486]. *2020 IEEE International Conference on Data Mining (ICDM)*, 948–953. https://doi.org/10.1109/ICDM50108.2020.00107

Cambridge dictionary. (2023, October 18). Cambridge dictionary. In *Cambridge dictionary*. Retrieved October 20, 2023, from https://dictionary.cambridge.org/dictionary/english/credit-risk

Cawley, G. C., & Talbot, N. L. C. (2009). On over-fitting in model selection and subsequent selection bias in performance evaluation.

CEIC. (2023a, June 1). EU household debt: % Of GDP. Retrieved October 22, 2023, from https://www.ceicdata.com/en/indicator/european-union/household-debt--of-nominal-gdp

CEIC. (2023b, June 1). Netherlands household debt. Retrieved October 22, 2023, from https://www.ceicdata.com/en/indicator/netherlands/household-debt

Chakraborty, B., & Yoshida, S. (2017). Proposal of a new similarity measure based on delay embedding for time series classification. In I. Rojas, H. Pomares, & O. Valenzuela (Eds.), *Advances in time series analysis and forecasting* (pp. 271–284). Springer International Publishing. https://doi.org/10.1007/978-3-319-55789-2_19

Chen, J., Wan, Y., Wang, X., & Xuan, Y. (2022). Learning-based shapelets discovery by feature selection for time series classification. *Applied Intelligence*, *52*(8), 9460–9475. https://doi.org/10.1007/s10489-021-03009-7

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Christ, M., Braun, N., Neuffer, J., & Kempa-Liehr, A. W. (2018). Time series FeatuRe extraction on basis of scalable hypothesis tests (tsfresh – a python package). *Neurocomputing*, *307*, 72–77. https://doi.org/10.1016/j.neucom.2018.03.067

Correa Bahnsen, A., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications*, *51*, 134–142. https://doi.org/10.1016/j.eswa.2015.12.030

Dastile, X., Celik, T., & Potsane, M. (2020). Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*, *91*, 106263. https://doi.org/10.1016/j.asoc.2020.106263

de Lange, P. E., Melsom, B., Vennerød, C. B., & Westgaard, S. (2022). Explainable AI for credit assessment in banks [Number: 12 Publisher: Multidisciplinary Digital Publishing Institute]. *Journal of Risk and Financial Management*, *15*(12), 556. https://doi.org/10.3390/jrfm15120556

Dempster, A., Petitjean, F., & Webb, G. I. (2020). ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, *34*(5), 1454–1495. https://doi.org/10.1007/s10618-020-00701-z

Deng, H., Runger, G., Tuv, E., & Vladimir, M. (2013). A time series forest for classification and feature extraction. *Information Sciences*, *239*, 142–153. https://doi.org/10.1016/j.ins.2013.02.030

Durand, D. (1941). Summary of findings. In *Risk elements in consumer instalment financing* (pp. 1–8). NBER. Retrieved October 22, 2023, from https://www.nber.org/books-and-chapters/risk-elements-consumer-instalment-financing/summary-findings

Eamonn, K. (2022, February 2). *Finding approximately repeated patterns in time series: The most useful, and yet most underutilized primitive in time series analytics*. https://u-paris.fr/diip/diip-seminars/

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: A review. *Data Mining and Knowledge Discovery*, *33*(4), 917–963. https://doi.org/10.1007/s10618-019-00619-1

Fernandez, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, *61*, 863–905. https://doi.org/10.1613/jair.1.11192

Flynn, M., Large, J., & Bagnall, T. (2019). The contract random interval spectral ensemble (c-RISE): The effect of contracting a classifier on accuracy. In H. Pérez García, L. Sánchez González, M. Castejón Limas, H. Quintián Pardo, & E. Corchado Rodríguez (Eds.), *Hybrid artificial intelligent systems* (pp. 381–392). Springer International Publishing. https://doi.org/10.1007/978-3-030-29859-3_33

Gao, J., Sun, W., & Sui, X. (2021). Research on default prediction for credit card users based on XGBoost-LSTM model (A. Farouk, Ed.). *Discrete Dynamics in Nature and Society*, *2021*, 1–13. https://doi.org/10.1155/2021/5080472

Grabocka, J., Schilling, N., Wistuba, M., & Schmidt-Thieme, L. (2014). Learning time-series shapelets. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 392–401. https://doi.org/10.1145/2623330.2623613

Gunnarsson, B. R., Vanden Broucke, S., Baesens, B., Óskarsdóttir, M., & Lemahieu, W. (2021). Deep learning for credit scoring: Do or don't? *European Journal of Operational Research*, *295*(1), 292–305. https://doi.org/10.1016/j.ejor.2021.03.006

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: A review [_eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-985X.1997.00078.x]. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, *160*(3), 523–541. https://doi.org/10.1111/j.1467-985X.1997.00078.x

Hills, J., Lines, J., Baranauskas, E., Mapp, J., & Bagnall, A. (2014). Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, *28*(4), 851–881. https://doi.org/10.1007/s10618-013-0322-1

Jastrzebska, A., Homenda, W., & Pedrycz, W. (2022). ARIMA feature-based approach to time series classification. In D. Groen, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, & P. M. A. Sloot (Eds.), *Computational science – ICCS 2022* (pp. 192–199). Springer International Publishing. https://doi.org/10.1007/978-3-031-08754-7_26

Kao, L.-J., Chiu, C.-C., & Chiu, F.-Y. (2012). A bayesian latent variable model with classification and regression tree approach for behavior and credit scoring. *Knowledge-Based Systems*, *36*, 245–252. https://doi.org/10.1016/j.knosys.2012.07.004

Karim, F., Majumdar, S., Darabi, H., & Harford, S. (2019). Multivariate LSTM-FCNs for time series classification. *Neural Networks*, *116*, 237–245. https://doi.org/10.1016/j.neunet.2019.04.014

Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, *34*(11), 2767–2787. https://doi.org/10.1016/j.jbankfin.2010.06.001

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1d convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, *151*, 107398. https://doi.org/10.1016/j.ymssp.2020.107398

Kvamme, H., Sellereite, N., Aas, K., & Sjursen, S. (2018). Predicting mortgage default using convolutional neural networks. *Expert Systems with Applications*, *102*, 207–217. https://doi.org/10.1016/j.eswa.2018.02.029

Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, *247*(1), 124–136. https://doi.org/10.1016/j.ejor.2015.05.030

Li, T., Kou, G., & Peng, Y. (2023). A new representation learning approach for credit data analysis. *Information Sciences*, *627*, 115–131. https://doi.org/10.1016/j.ins.2023.01.068

Liang, L., & Cai, X. (2020). Forecasting peer-to-peer platform default rate with LSTM neural network. *Electronic Commerce Research and Applications*, *43*, 100997. https://doi.org/10.1016/j.elerap.2020.100997

Löning, M., Bagnall, A., Ganesh, S., & Kazakov, V. (n.d.). Sktime: A unified interface for machine learning with time series.

Löning, M., Bagnall, A., Ganesh, S., Kazakov, V., Lines, J., & Király, F. J. (2019, September 17). Sktime: A unified interface for machine learning with time series. https://doi.org/10.48550/arXiv.1909.07872

Lubba, C. H., Sethi, S. S., Knaute, P., Schultz, S. R., Fulcher, B. D., & Jones, N. S. (2019). Catch22: CAnonical time-series CHaracteristics. *Data Mining and Knowledge Discovery*, *33*(6), 1821–1852. https://doi.org/10.1007/s10618-019-00647-x

Ma, X., & Lv, S. (2019). Financial credit risk prediction in internet finance driven by machine learning. *Neural Computing and Applications*, *31*(12), 8359–8367. https://doi.org/10.1007/s00521-018-3963-6

Mahbobi, M., Kimiagari, S., & Vasudevan, M. (2021). Credit risk classification: An integrated predictive accuracy algorithm using artificial and deep neural networks. *Annals of Operations Research*. https://doi.org/10.1007/s10479-021-04114-z

Mathai, N., Chen, Y., & Kirchmair, J. (2020). Validation strategies for target prediction methods. *Briefings in Bioinformatics*, *21*(3), 791–802. https://doi.org/10.1093/bib/bbz026

McKinsey. (2018, January 24). *PSD2: Taking advantage of open-banking disruption*. Retrieved October 22, 2023, from https://www.mckinsey.com/industries/financial-services/our-insights/psd2-taking-advantage-of-open-banking-disruption

Merćep, A., Mrčela, L., Birov, M., & Kostanjčar, Z. (2021). Deep neural networks for behavioral credit rating [Number: 1 Publisher: Multidisciplinary Digital Publishing Institute]. *Entropy*, *23*(1), 27. https://doi.org/10.3390/e23010027

Middlehurst, M., Large, J., & Bagnall, A. (2020). The canonical interval forest (CIF) classifier for time series classification. *2020 IEEE International Conference on Big Data (Big Data)*, 188–195. https://doi.org/10.1109/BigData50022.2020.9378424

Middlehurst, M., Large, J., Flynn, M., Lines, J., Bostrom, A., & Bagnall, A. (2021). HIVE-COTE 2.0: A new meta ensemble for time series classification. *Machine Learning*, *110*(11), 3211–3243. https://doi.org/10.1007/s10994-021-06057-9

Middlehurst, M., Schäfer, P., & Bagnall, A. (2023, April 25). Bake off redux: A review and experimental evaluation of recent time series classification algorithms. https://doi.org/10.48550/arXiv.2304.13029

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research [Publisher: Taylor & Francis, Ltd.]. *Journal of Management Information Systems*, *24*(3), 45–77. Retrieved November 20, 2023, from https://www.jstor.org/stable/40398896

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. https://doi.org/10.1145/2939672.2939778

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, *8*(4), e1249. https://doi.org/10.1002/widm.1249

Tan, C. W., Dempster, A., Bergmeir, C., & Webb, G. I. (2022). MultiRocket: Multiple pooling operators and transformations for fast and effective time series classification. *Data Mining and Knowledge Discovery*, *36*(5), 1623–1646. https://doi.org/10.1007/s10618-022-00844-1

Tan, Y., & Zhao, G. (2022). Multi-view representation learning with kolmogorov-smirnov to predict default based on imbalanced and complex dataset. *Information Sciences*, *596*, 380–394. https://doi.org/10.1016/j.ins.2022.03.022

Thomas, L. C. (2010). Consumer finance: Challenges for operational research [Publisher: Taylor & Francis _eprint: https://doi.org/10.1057/jors.2009.104]. *Journal of the Operational Research Society*, *61*(1), 41–52. https://doi.org/10.1057/jors.2009.104

Tripathi, D., Shukla, A. K., Reddy, B. R., Bopche, G. S., & Chandramohan, D. (2022). Credit scoring models using ensemble learning and classification approaches: A comprehensive survey. *Wireless Personal Communications*, *123*(1), 785–812. https://doi.org/10.1007/s11277-021-09158-9

van Thiel, D., & van Raaij, W. F. ((2019). Artificial intelligence credit risk prediction: An empirical study of analytical artificial intelligence tools for credit risk prediction in a digital era. *Journal of Risk Management in Financial Institutions*, *12*(3), 268–286.

Wang, H. (2021). Credit risk management of consumer finance based on big data [Publisher: Hindawi]. *Mobile Information Systems*, *2021*, e8189255. https://doi.org/10.1155/2021/8189255

Wang, X., Wirth, A., & Wang, L. (2007). Structure-based statistical features and multivariate time series clustering. *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, 351–360. https://doi.org/10.1109/ICDM.2007.103

Wiginton, J. C. (1980). A note on the comparison of logit and discriminant models of consumer credit behavior [Publisher: Cambridge University Press]. *The Journal of Financial and Quantitative Analysis*, *15*(3), 757–770. https://doi.org/10.2307/2330408

Yan, Q., & Cao, Y. (2020). Optimizing shapelets quality measure for imbalanced time series classification. *Applied Intelligence*, *50*(2), 519–536. https://doi.org/10.1007/s10489-019-01535-z

Ye, L., & Keogh, E. (2009). Time series shapelets: A new primitive for data mining. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 947–956. https://doi.org/10.1145/1557019.1557122

Yu, L., Wang, S., & Lai, K. (2008). Credit risk assessment with a multistage neural network ensemble learning approach. *Expert Systems with Applications*, *34*(2), 1434–1444. https://doi.org/10.1016/j.eswa.2007.01.009

Zhang, X., & Yu, L. (2024). Consumer credit risk assessment: A review from the state-of-the-art classification algorithms, data traits, and learning methods. *Expert Systems with Applications*, *237*, 121484. https://doi.org/10.1016/j.eswa.2023.121484

Appendix A

Studies in credit risk

Table A.1: Summary studies on consumer credit risk models. SVM = Support Vector Machine, NN = Neural Network(aggregated term), LR = Logistics regression, KNN = K-Nearest Neighbor

Paper	**ML model**	**Data**	**Evaluation metrics**	**Explainability**	**Validation method**	**Hybr/ ensemble**
Yu et al. (2008)	NN's, SVM, ELM's	Application data on retail customers	Confusion matrix, AUC	None	train-test split	Ensemble
Liang and Cai (2020)	NN	Time series of loan default rate	mean absolute error	None	moving and rolling cross-validation	Ensemble
Ma and Lv (2019)	MLIA, LR	application information of retail customers	AUC	None	train-test split	None
Brygała (2022)	LR	application information of retail customer	Confusion matrix, Total effectiveness	Significance	train-test split	None

Paper	ML model	Datatype	Evaluation metrics	Explainability	Validation method	Hybr/ ensemble
Merćep et al. (2021)	LR, SVM, RF, GB, NN	aggregated balance and transaction features	AUC, H-measure Brier score	-	cross-validation	None
H. Wang (2021)	SVM, ADA Boost, RF	Credit card data, data internet crawlers	Accuracy, precision, recall, F1	None	train-test split	Ensemble
de Lange et al. (2022)	Light-GBM, LR	balance and behavioral data	AUC, PR-curve, Approximated costs of imperfect credit scoring models	SHAP-value on explanatory variables	train-test split	Ensemble
Bücker et al. (2022)	GBM-model	inAnonimized credit information	AUC,	Importance, SHAP-value	train-test split	Ensemble
Khandani et al. (2010)	CART model	credit card data	Confusion matrix, AUC	claim interpretable	train-test split	Hybrid
Gao et al. (2021)	LSTM-XGBOOST	Monthly transaction and billing information	Confusion matrix, AUC	Feature importance	train-test split	Ensemble
Kvamme et al. (2018)	CNN, RF, Ensemble	End-of-day balance data on all accounts customer	Confusion matrix, AUC, Brier score, H-measure	None	train-test split	-

Paper	ML model	Datatype	Evaluation metrics	Explainability	Validation method	Hybr/ ensemble
Correa Bahnsen et al. (2016)	LR, DT, RF	Aggregated and raw features	total savings	None	Train-test split	Ensemble
Mahbobi et al. (2021)	SMOTE, NN, SVM, KNN	Behavioral and billing features	Confusion matrix, Brier score, cross-entropy loss	relative feature importance	train-test split	None
Addo et al. (2018)	SMOTE, NN, SVM, KNN	Enterprise dataset	AUC, RMSE	None	train-test split	Ensemble
Kao et al. (2012)	CART, LR, BP, SVM, MARS	application data and income features	Confusion matrix	Feature importance	train-test split	-
Li et al. (2023)	ADA Boost, RF, NN, distance based	Data on enterprises	Accuracy, AUC	Visualization of classes	3fold cross-validation	-
van Thiel and van Raaij (2019)	RF, NN	mortgage application data	Confusion matrix, AUC	None	Train-test split	Ensemble
Literature review papers on credit risk models						
Dastile et al. (2020)	overview multiple models	no data used	PCC, AUC, G-mean, F-measure	Rationalize outcomes	Train-test split	Ensemble
Zhang and Yu (2024)	overview multiple models	no data used	Confusion matrix, G-mean, F-measure	None	cross- validation	Both

Paper	ML model	Datatype	Evaluation metrics	Explainability	Validation method	Hybr/ ensemble
Lessmann et al. (2015)	Overview single and ensemble classifiers	Credit card data	AUC, PCC, BS, H, KS	None	train-test split	Both
Abellán and Castellano (2017)	Combining different ensemble methods with different classifiers	6 credit card datasets	accuracy, AUC	None	5 fold cross validation	Ensemble
Tripathi et al. (2022)	various hybrid and ensemble approaches	5 credit card datasets	Accuracy	None	10 fold cross validation	Hybrid

Table A.2: Studies in Time Series Classification

Paper	TSC Taxonomy	Context
Deng et al. (2013)	Interval	This study introduces a tree-ensemble method for time series classification
Lubba et al. (2019)	feature	This study introduces "catch22," a method that identifies a compact set of 22 time-series features with strong classification performance
Cabello et al. (2020)	interval	This study introduces the Supervised Time Series Forest (STSF) as a novel method for time series classification
Middlehurst et al. (2020)	interval	This study introduces the Canonical Interval Forest (CIF), a novel classifier that combines the strengths of the Time Series Forest (TSF) and the 'catch22' feature set.

Paper	TSC Taxonomy	Context
Flynn et al. (2019)	interval	This study introduces an enhanced version of the Random Interval Spectral Ensemble (RISE) called c-RISE
Eamonn (2022)	shapelet	This study introduces time series shapelets as a solution to the limitations of the nearest neighbor algorithm for time series classification.
Bagnall et al. (2017)	all	In this study, the authors evaluate 18 recently proposed time series classification algorithms on an expanded dataset of 85 time series datasets, conducting 100 resampling experiments for each
Middlehurst et al. (2023)	all	This study revisits the study of Bagnall et al. (2017) that compares the analysis of Time Series Classification (TSC) algorithms, originally conducted on 85 datasets from the University of California, Riverside (UCR) archive, and assesses the progress in various algorithm categories.
Karim et al. (2019)	deep learning	This study extends the capabilities of univariate time series classification models, such as the Long Short Term Memory Fully Convolutional Network (LSTM-FCN) and Attention LSTM-FCN (ALSTM-FCN), to address multivariate time series classification.
Dempster et al. (2020)	deep learning	this study demonstrates that simple linear classifiers with random convolutional kernels achieve state-of-the-art accuracy while significantly reducing computational costs.
Kiranyaz et al. (2021)	deep learning	This study offers a comprehensive review of 1D Convolutional Neural Networks (CNNs), highlighting their architecture, principles, and recent engineering applications.

Appendix B

Feature selection process model Y

B.1 Original features created from balance data

Classified

B.2 Feature selection pipeline

The model developers of model Y have chosen to remove features that have 90 percent missing values, have constant values with variance 0, and drop features with high correlation (Pearson correlation = 1). Figure B.1 shows the total feature selection. Then the model developers chose to drop features based on the following reasons:

1. **Feature importance.** Features with exactly zero feature importance on the IT data are eliminated.
2. **PSI-based feature selection.** Population stability index (PSI) is a measure of how much the distribution of a single feature has shifted. Here, we compute PSI values using CV on the IT data.
3. **Correlation-based feature selection.** Highly correlated features can inhibit the model performance and make interpretation of the final feature set difficult. Therefore, we explicitly removed features that were highly correlated to other features using the Pearson correlation coefficient.

4. **Recursive feature selection with cross-validation.** This approach is performed as the last step of the feature selection pipeline and is meant to remove features that do not significantly contribute to the overall model performance (selected number of features: 340). However, by all stakeholders, the model developers selected a lower number of features since it provided comparable performance and ensured a far lower burden on implementation and monitoring.

B.2.1 Top 10 features Model Y

Table B.1: Top 10 most important features ranked with mean absolute SHAP value.

Feature	**SHAP-value**
Feature 1	0.254827
Feature 2	0.225686
Feature 3	0.167179
Feature 4	0.12847
Feature 5	0.119022
Feature 6	0.10675
Feature 7	0.101947
Feature 8	0.0971233
Feature 9	0.0900051
feature 10	0.0848077

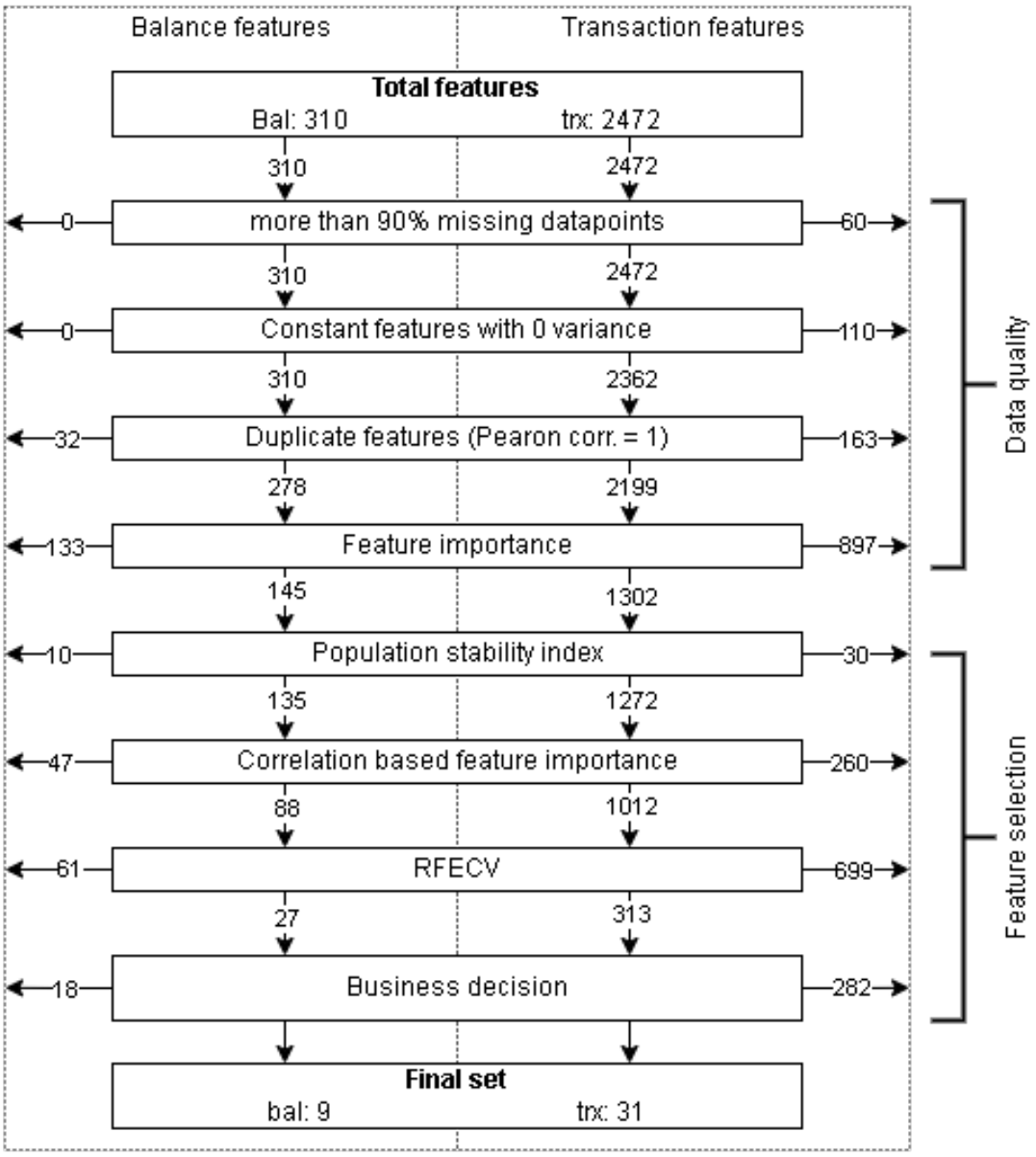


Figure B.1: Overview feature selection model Y

Appendix C

Experimental setup

C.1 summary statistics

Table C.1: Summary statistics of monthly balance data

	mean	std	min	0.25	0.5	0.75	max
Time series 1	XXX	XXX	XXX	XXX	XXX	XXX	XXX
Time series 2	XXX	XXX	XXX	XXX	XXX	XXX	XXX
Time series 3	XXX	XXX	XXX	XXX	XXX	XXX	XXX
Time series 4	XXX	XXX	XXX	XXX	XXX	XXX	XXX
Time series 5	XXX	XXX	XXX	XXX	XXX	XXX	XXX
Time series 6	XXX	XXX	XXX	XXX	XXX	XXX	XXX
Time series 7	XXX	XXX	XXX	XXX	XXX	XXX	XXX

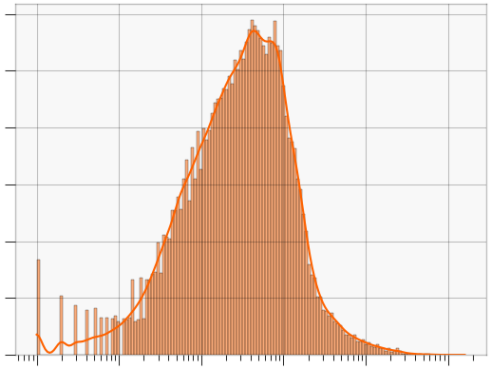
C.2 Distribution Shapes

Distribution shapes of the multivariate monthly dataset are given in Figure C.1

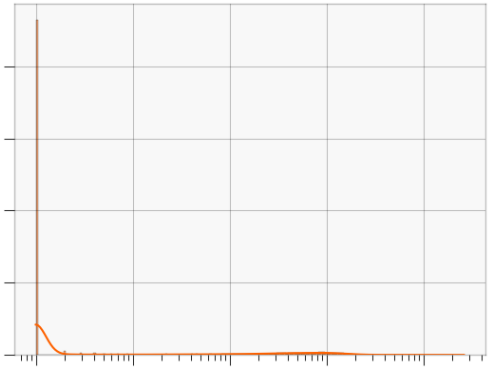
C.3 Hyperparameters models

Table C.2: Hyperparameters of used models

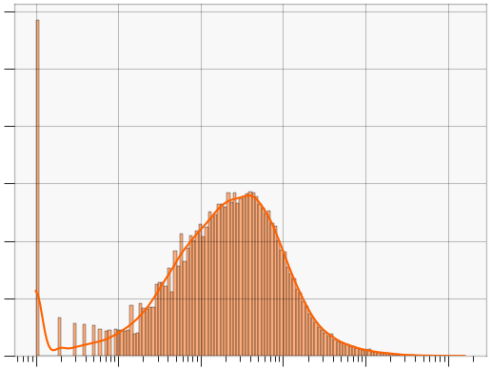
Model Name	**Hyperparameters**
CIF	• $n_{estimators} = 200$ • $n_{intervals} = \sqrt{\text{series\_length}} \times \sqrt{\text{n\_dims}}$ • att_subsample_size = 8 • min_interval = 7 • max_interval = 365 • base_estimator = $DecisionTreeClassifier$
SHAPELET	• n_shapelet_samples = 1000 • max_shapelets = $n_{classes}/n_{n\_shapelet\_samples}$ • max_shapelet_length = None • estimator = Random Forest • transform_limit_in_minutes = 0 • time_limit_in_minutes = 0
ROCKET	• num_kernels = 8,000
LSTM	• n_epochs=2000 • batch_size=128 • dropout=0.8 • kernel_sizes=(8, 5, 3) • filter_sizes=(128, 256, 128) • lstm_size=8 • attention=False



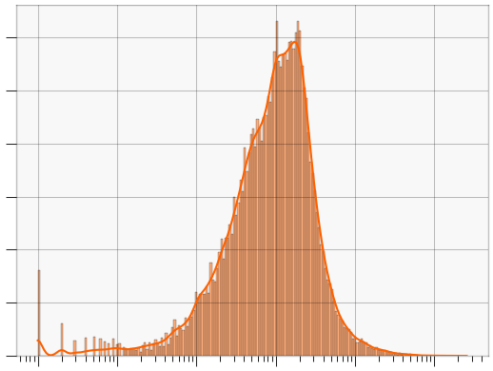
(a) Time series 7



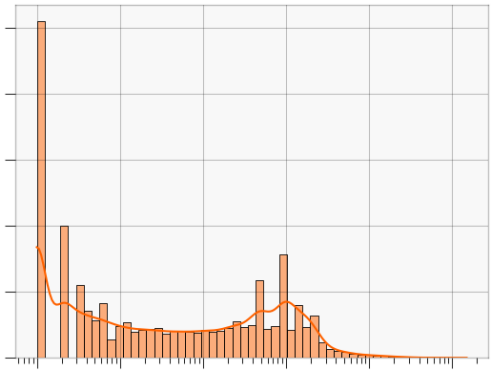
(b) Time series 5



(c) Time series 6



(d) Time series 4



(e) Time series 3

Figure C.1: Distribution shapes of the multivariate monthly dataset.

Appendix D

Catch-22 features

Catch-22 features[1]

[1]Sourced from: https://time-series-features.gitbook.io/catch22-features/

Feature Description	Category
DN_HistogramMode_10: 10-bin histogram mode	Distribution Shape
DN_HistogramMode_5: 5-bin histogram mode	Distribution Shape
CO_f1ecac: First autocorrelation coefficient	Linear Autocorrelation
CO_FirstMin_ac: First minimum in the autocorrelation function	Linear Autocorrelation
CO_HistogramAMI_even_2_5: Automutual information with even bins, delay 2	Nonlinear Autocorrelation
CO_trev_1_num: Time reversal asymmetry statistic	Nonlinear Autocorrelation
MD_hrv_classic_pnn40: Proportion of NN intervals differing by more than 40 ms	Incremental Differences
SB_BinaryStats_mean_longstretch1: Mean length of longest homogeneous segments	Symbolic
SB_TransitionMatrix_3ac_sumdiagcov: Sum of diagonal covariance in transition matrix	Symbolic
PD_PeriodicityWang_th0_01: Periodicity measure by Wang et al.	Linear Autocorrelation
CO_Embed2_Dist_tau_d_expfit_meandiff: Mean difference in embedded point distances	Other
IN_AutoMutualInfoStats_40_gaussian_fmmi: Automutual information with Gaussian window, lag 40	Nonlinear Autocorrelation
FC_LocalSimple_mean1_tauresrat: Ratio of mean period of local maxima to minima	Incremental Differences
FC_LocalSimple_mean3_stderr: Standard error over mean of local simple features	Simple Forecasting
DN_OutlierInclude_p_001_mdrmd: Outlier measure for positive outliers	Extreme Event Timing
DN_OutlierInclude_n_001_mdrmd: Outlier measure for negative outliers	Extreme Event Timing
SP_Summaries_welch_rect_area_5_1: Area under Welch's spectral density estimate	Linear Autocorrelation
SP_Summaries_welch_rect_centroid: Centroid of Welch's spectral density estimate	Linear Autocorrelation
SB_BinaryStats_diff_longstretch0: Difference in longest homogeneous segments of zeros	Symbolic
SB_MotifThree_quantile_hh: Entropy of successive pairs in symbolized series	Symbolic
SC_FluctAnal_2_rsrangefit_50_1_logi_prop_r1: Fluctuation analysis (low-scale scaling)	Self-Affine Scaling
SC_FluctAnal_2_dfa_50_1_2_logi_prop_r1: Detrended fluctuation analysis (low-scale scaling)	Self-Affine Scaling

Table D.1: Description of Catch22 Features with Categories

Glossary

Catch-22 In the context of time series analysis, Catch-22 refers to a curated set of 22 highly informative characteristics. These features are selected for their ability to succinctly capture the essential properties of time series data, facilitating various analytical tasks such as classification or pattern recognition (Lubba et al., 2019). 14, 22, 43, 45, 54

XGBoost An advanced implementation of gradient boosting algorithms, XGBoost (eXtreme Gradient Boosting) is an ensemble learning technique that enhances predictive accuracy by aggregating the outputs of several simpler models. This method leverages gradient boosting frameworks at scale and is recognized for its efficiency, flexibility, and portability (T. Chen & Guestrin, 2016). 9, 10, 14–17, 19, 21, 23, 29, 39, 40, 48–51