Development of Financial Distress Prediction Model for the Watchlist Classification of Wholesale Banking Clients at ING

by

Daniel Tianfu Chen

A thesis submitted to the
Faculty of Behavioural, Management and Social Sciences (BMS) and
in partial fulfilment of the requirements for the degree of

MSc in Industrial Engineering and Management

Faculty of Behavioural, Management and Social Sciences (BMS)

University of Twente

Enschede, Overijssel, The Netherlands

June 2023

Abstract

An Early Warning System (EWS) is a tool that enables the monitoring of the credit portfolio to identify clients in financial distress. ARIA is the EWS used by ING to monitor their Wholesale Banking (WB) clients using a variety of early warning triggers based on internal data, news articles, and market data. However, the current triggers are limited in their predictive capabilities as they are backwards-looking and are only derived from a single variable. A new Watchlist (WL) trigger aims to incorporate the information of all the current triggers into a single model that can predict whether a client should be on a watchlist based on their credit risk. The aim of this research focuses on exploring how such a WL trigger could be designed by answering the following main research question:

How could a WL trigger be designed that is able to effectively classify WB clients at ING on a watchlist based on their prospective credit risk?

This study introduces three metrics that measure the relationship between the triggers and the status of a client. A good WL trigger would have the following properties: it should be able to detect as many clients in distress as possible, raised triggers should indicate if there is a high probability that the client will be in financial distress, and the trigger should be able to detect financial distress as early as possible. These metrics can be measured by migration sensitivity, trigger precision, and time lag, respectively.

Using ML techniques, a financial distress prediction model is developed that determines when a WL trigger should be raised. This financial distress prediction uses internal triggers, external triggers, and internal client data as input to predict if a client will be in financial distress. The literature has different definitions for financial distress, and in this research, we define a financially distressed client as a client with a watchlist or default status. The financial distress prediction model tries to predict when a client will migrate from a regular status to a watchlist or default status, referred to as a negative migration.

The proposed model incorporates the historical data six months prior (the time window) to a negative migration to forecast if a negative migration occurs in the next month (the time gap). Furthermore, the model incorporates a target window of six months which adds flexibility to the model. This target window allows the model to make predictions before a negative migration. We are only interested in the early detection of financial distress, so we do not necessarily want to predict the exact moment of negative migration, which is made possible by the introduced target window.

Several supervised learning algorithms, including Linear Discriminant Analysis (LDA), Logistic Regression (LR), Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting Machine (GBM), Extreme Gradient Boosting (XGBoost), and Artificial Neural Network (ANN) are tested. Experimentation shows that the Random Forest model has the highest performance, with an F1 score of 0.197, a trigger precision of 0.127, and a migration sensitivity of 0.676. This means that if this model raises a trigger, there is a 12.7% probability that the client will have a negative migration, and among all negative migrations, the trigger occurred 67.6% of the time before the migration. Furthermore, the experiments show that extending the time and target window improves model performance. In addition, the timeliness of the model can be altered by configuring the model's target window and time gap.

More research is needed to investigate the optimal values for the time window, target window, and time gap. For this, more historical data needs to be collected, and the tradeoff between trigger precision, migration sensitivity and time lag needs to be investigated for future research. Furthermore, the financial distress prediction model could be improved by more extensive model tuning, considering other modelling approaches and collecting data from other sources.

Keywords: machine learning, early warning system, EWS, early warning trigger, credit risk, watchlist, default, financial distress prediction

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Twente to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Twente to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Daniel Tianfu Chen

ACKNOWLEDGEMENTS

I want to express my sincere gratitude to the following people, without whom this thesis would not have been possible:

First and foremost, I would like to express my appreciation to my thesis supervisors, Jörg Osterrieder, Marcos Machado, and Rui Santos, for their guidance, support, and valuable feedback throughout my research. I want to thank Jörg for connecting me with ING, which resulted in my graduation internship. In addition, I am thankful for the expertise and feedback you provided during my research. I am grateful to Marcos for the comprehensive feedback he consistently provided me on my thesis. It immensely helped me take my thesis to a higher level. To Rui, I would like to express my gratitude for his great enthusiasm for my project and for helping me shape my research.

Next, I would like to express my gratitude towards my colleagues and friends at ING. I am grateful to have worked with the ARIA squad, including Rui Santos, Mehmet Simsek, Peter Lichtenveldt, Christopher Pironti, Robin Zijp, Semida Andreicha, Krzysztof Mirek, Wioleta Ranik, Michał Kajstura, Piotr Treska, and Alessandra Amato. I want to thank them for making me feel part of the team, inspiring me with their work, and providing invaluable feedback. Besides, I want to give my special thanks to Robin for his mentorship. I thank him for helping me navigate the ING landscape, connecting me with many interesting people, and teaching me many things about the inner workings of a bank, making both me and my research more professional. In addition, I want to thank Christoper for supporting me with the model development of my study by introducing me to new concepts and validating my ideas. I really enjoyed our discussions about data science from which I have learned a lot.

Furthermore, I want to thank my data science chapter, including Cecilia Miao, Ali Hasmi, Andrei Rosu, Ferry Besamusca, Albert Yumol, Camille Rivero-Co, Christopher O'Lenskie, Keng Ng, and Ahmet Gok. They provided me with countless inspirations and feedback for my research. Also, I enjoyed spending time with you during the many company events and data science conferences. Next, I want to thank my previous external supervisor Anand Autar for his attentive monitoring of my progress and his commitment to ensuring I had access to all the necessary resources to complete my project successfully. Moreover, I thank Dirk Pronk, Marike Gelinck, and Diederik Sluijs for helping translate my technical insights into business needs. And, I am grateful to Joost Butter for reading my thesis and providing detailed feedback.

Lastly, I am grateful to my family for their support and encouragement throughout my research. Also, I want to thank my roommates, and friends from my study, fraternity Xaos, and the rowing association Euros for helping me keep my sanity during the long hours in the university library

by accompanying me with the many coffee breaks and offering their time to discuss my project.

Thank you all for your invaluable contributions to this thesis, and I hope you will enjoy reading it.

Contents

List of Figures

List of Tables

Abbreviations

ANN	Artificial Neural Network.
ARIA	Advanced Risk Integrated Application.
CSRC	China Securities Regulatory Commission.
DAG	Directed Acyclic Graph.
DT	Decision Tree.
EAD	Exposure At Default.
ECB	European Central Bank.
EWI	Early Warning Indicator.
EWS	Early Warning System.
FN	False Negative.
FP	False Positive.
GBM	Gradient Boosting Machine.
HMM	Hidden Markov Models.
LDA	Linear Discriminant Analysis.
LGD	Loss Given Default.
LR	Logistic Regression.
MCC	Matthews Correlation Coefficient.
MCDA	Multiple-Criteria Decision Analysis.
MDA	Multiple Discriminant Analysis.
ML	Machine Learning.
MLOps	Machine Learning Operations.
NLP	Natural Language Processing.
PD	Probability of Default.
RF	Random Forest.
ROS	Random Over-Sampling.
RUS	Random Under-Sampling.
RWA	Risk-Weighted Asset.
SHAP	SHapley Additive exPlanations.
ST	Special Treatment.
SVM	Support Vector Machine.
TN	True Negative.
TP	True Positive.
WB	Wholesale Banking.
WL	Watchlist.

XAI	Explainable AI.
XGBoost	Extreme Gradient Boosting.

1

Introduction

1.1. Data-driven Early Warning Systems

The early detection of increased credit risk is key for identifying clients in financial distress so that timely actions can be taken to avoid potential losses. An Early Warning System (EWS) enables the effective monitoring of the credit portfolio by providing Early Warning Indicators (EWI) and triggers to alert stakeholders such as risk and account managers when there are early signs of default. Successful implementations of EWS can substantially reduce the exposure of both retail and Wholesale Banking (WB) clients [1]. Another advantage is that EWS improves customer relationships since it enables proactively alerting and advising distressed clients to help them avoid possible bankruptcy. Finally, EWS is essential from a compliance perspective. The European Central Bank (ECB) provides guidance on the monitoring and management of loans that are unlikely to be paid back using an EWS. The guidance is non-binding, but banks should be able to explain themselves when they deviate from the guide [2].

The rapid increase of data calls for the use of data-driven EWS to incorporate big data in identifying credit risk for banks to stay competitive. Traditional early warning models were limited to the rudimentary analyses of accounting variables. Due to technological developments regarding the internet and computer processing power, opportunities arise to improve EWS by incorporating new data sources. For example, real-time financial data and relevant news articles posted online could enhance early warning capabilities [3].

Furthermore, Machine Learning (ML) has become a widely used tool to deal with these tremendous flows of data in EWS [4–6]. Currently, many banks rely on EWS consisting of ad hoc triggers derived from a single financial metric resulting in imprecise models with a high number of false positives [7]. The application of ML techniques can improve early warning signalling by providing automated decision-making and incorporating multiple variables to increase the performance of the early warning models. This research aims to investigate the applications of

ML models for the early detection of WB clients in distress at ING, a Dutch commercial bank [8].

1.2. PROBLEM BACKGROUND

ING is an international commercial bank established in the Netherlands with more than 51,000 employees offering retail and WB services in over 40 countries worldwide [8]. One of the WB services includes issuing loans to large clients such as big corporations, governments, and other financial institutions. These loans need to be continually monitored so that precautionary actions can be taken when the client's risk profile significantly changes.

The Advanced Risk Integrated Application (ARIA) is an EWS developed by ING that provides early warning triggers to monitor the credit risk of INGs WB clients. For this, ARIA utilizes internal and external data sources. The internal data consists of data from the internal systems and models of ING, such as the Probability of Default (PD) and the Risk-Weighted Assets (RWA). Furthermore, external data consists of data from sources outside the company, including stock prices, prices of derivatives, and online news articles. ARIA uses this data to calculate early warning triggers to signal potential increases in risk. For example, a trigger could be activated when the stock price of a company decreases by more than 10%. Additionally, ING developed ML models to identify risks by creating triggers that analyse news articles. ARIA extracts over 300.000 news articles daily from Google News and the Financial Times. These are then processed through the use of Natural Language Processing (NLP), specifically, topic models to identify if a company is involved with fraud, an acquisition, bankruptcy, or any other relevant activity.

If an early warning trigger is raised, a risk manager could decide if a client requires closer monitoring, or they can take predefined actions to mitigate the risks [9]. These actions could include changing the status client, implementing a forbearance strategy, writing off the exposures or taking legal actions.

1.3. PROBLEM STATEMENT

ING has policies that put WB clients with significant credit risk on a watchlist. The goal of the watchlist is to identify clients in distress at an early stage so that timely mitigation and forbearance actions can be taken before a default can happen. This way, potential losses and the probability that a client will default can be reduced. ING makes a distinction between a regular watchlist and a portfolio watchlist.

The regular watchlist is based on a manual assessment of a client with the support of monitoring tools. The regular watchlist procedure uses EWIs that support the decision-making of risk and account managers. These EWIs are predefined qualitative and quantitative indicators such as macro-economic variables, financial indicators and the client's behaviour. To give an illustration, a client can be watchlisted when affected by a negative Gross Domestic Product (GDP) growth or if its operations violate laws and regulations. When an account manager identifies a client with a significant risk increase, they need to fill out a form in order to give a client a watchlist status substantiated by reasons for the new classification. The reasons provided should be grounded based on the EWI defined in the watchlist policy. In contrast to the regular watchlist, a portfolio watchlist is a data-driven approach that automatically monitors and classifies clients on a watchlist. Currently, ING does not have a process for a watchlist portfolio to automatically monitor their WB clients.

The ARIA tool supports the watchlist classification by giving risk managers insight into their portfolio using early warning triggers. However, when it comes to decision-making for watchlists, the currently employed early warning triggers have certain limitations. Firstly, there is no clear relationship between the early warning triggers and the watchlist status of clients. Currently, it is not possible to monitor the effectiveness of early warning triggers for detecting clients with increased risks. Consequently, risk managers need to rely on their own experience and intuition to assess the impact of specific triggers on the status of a client. Secondly, the triggers are univariate, which means that they are only based on a single attribute. A disadvantage of univariate triggers is that they have a low hit ratio (precision) which limits their predictive power [7]. Thus, these triggers cannot accurately distinguish well-performing clients from clients in financial distress. Finally, the current triggers are primarily backwards-looking. This means that triggers reflect past changes, limiting the earliness of their insights. Instead, a trigger should aim to be forward-looking by utilizing patterns from the past to make predictions about the future so that signals can be raised as early as possible.

1.4. RESEARCH OBJECTIVES AND CONTRIBUTIONS

The primary goal of this study is to create a connection between early warning triggers and the impact on the credit status of clients. This study investigates what properties an early warning trigger should have and how these could be quantified using different metrics. Additionally, this research focuses on the development of a new trigger for the watchlist classification of clients in financial distress, referred to as the Watchlist (WL) trigger. For this, a case study has been conducted at ING, where the implementation of a financial distress prediction model using ML techniques has been investigated. Such a model combines the predictive power of multiple univariate early warning triggers to make credit risk forecasts in the future. Furthermore, the scope of this research is restricted to WB clients and credit risk. The data-driven WL trigger would have the following practical contributions compared to the current manual procedure:

- **Automation**: an ML model could automate the watchlist classification procedure, which saves the manual labour currently done.
- **Timeliness**: clients in distress could be detected earlier because the data can be analysed much quicker, and the analysis could be performed anytime when the model is run.
- **Consistency**: an algorithm ensures that the decisions it makes are consistent so that clients are always assessed in the same way.

- **Predictive performance**: an ML model can incorporate all the data available in its decision-making and identify new patterns that could improve the performance of the watchlist procedure.

Based on the aforementioned research goals, the following main research question has been defined:

How could a WL trigger be designed that is able to effectively classify WB clients at ING on a watchlist based on their prospective credit risk?

The main research question has been subdivided into the following subquestions:

1. How do the early warning triggers relate to the watchlist status of clients?
   (a) What properties does an effective early warning trigger have?
   (b) How can we measure these properties?
2. How can a financial distress prediction model be created for the watchlist classification of clients?
   (a) Which requirements does such a model have?
   (b) Which ML techniques can be used?
   (c) How can we evaluate the predictive power of this model?
   (d) How can the decision-making of the model be explained?
3. How can we implement a new WL trigger?
   (a) How does the proposed WL trigger compare to the existing ARIA triggers?
   (b) How can the WL trigger be implemented in the current processes of ARIA?

The thesis is structured as follows. Firstly, the literature related to ML in credit risk, financial distress prediction models, and EWSs are reviewed in Chapter 2. Secondly, the data collection process and data exploration analysis are discussed in Chapter 3, where the relationship between early warning triggers and the watchlist classifications is researched. Afterwards, different ML techniques that could be used for predicting financial distress are discussed in chapter 4. Subsequently, Chapter 5 introduces the experimental set-up of our financial distress prediciton model, and Chapter 6 discusses the results of the experiments. Finally, Chapter 7 discusses the conclusions, recommendations, and future research topics.

2

LITERATURE REVIEW

2.1. ML IN CREDIT RISK

Credit risk is the risk associated with a borrower not being able to meet their financial obligations to the bank resulting in a possible loss [10]. The assessment of credit risk is a key activity for banks to ensure profitability, compliance with banking regulations, and competitiveness [11]. ML is a suitable approach for forecasting credit risk because these models can deal with complex non-linear relationships [12], they can incorporate big data into the decision-making [4], and they can automate lending processes [13].

Applications of ML in credit risk could be roughly divided into the following three topics: fraud detection, credit scoring, and financial distress prediction [14]. Firstly, fraud detection models try to identify fraudulent behaviour patterns using anomaly detection methods and historical transaction data [15]. Secondly, banks use credit scoring models to assess the creditworthiness of their clients to determine if a client can be on board or should be written off. Supervised ML methods for credit scoring have been widely studied in the literature [11, 16, 17]. Finally, ML models are used to predict if a client will be in financial distress so that timely actions can be taken. Credit scoring and financial distress prediction allow for similar implementations of binary classification models, but they serve different purposes. The main difference is the costs associated with false positives in the model: false positives in credit rating models can result in lenders declining loans due to mispricing, while false positives in EWSs may only result in additional workload for staff but may not result in immediate action [18]. The latter application is the focus of this research, and related literature is described in more detail in the next section.

2.2. EARLY WARNING MONITORING OF FINANCIAL DISTRESS

An EWS is a system used to detect potential risk at an early stage so that preventive or mitigation actions can be taken before any problematic events can occur [19]. Such a system can display EWIs or triggers to alert stakeholders when early signs of risks are identified [1]. EWSs have been adopted in many financial use cases, such as the prediction of financial crises by central banks [20], the identification of likely-to-fail banks [21], and the credit risk monitoring of sovereign debts [22]. Next to these applications, EWSs are widely used to monitor the credit risk of borrowers, particularly by lenders and other stakeholders exposed to that credit risk [7]. A financial distress prediction model can be used as an EWI to predict the likelihood that a particular borrower will experience financial distress.

A literature review is conducted to analyse the application of financial distress prediction models for early warning monitoring. The scope is restricted to the monitoring of the credit risk of companies since this research only focuses on WB clients. The papers are compared by the financial distress prediction modelling approach, the independent variables, and the definition of financial distress used. Furthermore, we look at how the model is applied in an EWS and what type of clients are included. A summary of the literature review can be found in Appendix A.

In the literature, there is no general consensus on the definition of financial distress. As a result, the literature uses different dependent variables for training their models. For example, many distress prediction models use bankruptcy data to determine which clients are in distress [23]. However, in the case of early warning monitoring, most papers use a broader definition for financial distress as a financial distress model should be able to predict both the early and advanced stages of financial distress so that early actions can be taken to prevent high costs [24]. Likewise, Wang [25] argues that financial distress should have a more general definition that includes the states between economic failure, insolvency, and bankruptcy.

As a result, the literature related to early warning signals extends the definition of financial distress by formulating different criteria. Most of the literature developed a financial distress prediction model based on data from the China Securities Regulatory Commission (CSRC). CSRC defines so-called Special Treatment (ST) companies that made financial losses for two consecutive years [26]. Additionally, Tong and Tong [27] labelled distressed companies based on whether a company's cash flows decreased below a certain threshold, which would mean that borrowers would not be able to obtain new loans. Balasubramanian et al. [28] determined a company to be in financial distress when the accumulated losses over a year exceeded the entire net worth of a company. Ashraf et al. [24] based financial distress on multiple criteria related to book value, dividend declaration, failing to hold an annual general meeting, and bankruptcy.

Various financial distress prediction models are researched in the context of early warning monitoring. The majority of these models are based on supervised ML models including Decision Tree (DT), Logistic Regression (LR), Multiple Discriminant Analysis (MDA), Artificial Neural Network (ANN), Support Vector Machine (SVM), and ensemble methods like Random Forest (RF), Extreme Gradient Boosting (XGBoost) and CatBoost [4, 19, 25, 27–38]. Furthermore, there are a couple of implementations of traditional formulas like the Altman Z-score that use balance sheet-based financial ratios as input [24, 25]. In addition, Chen et al. [26] implements a

KMV model which measures credit risk based on an option pricing approach. Finally, some approaches involve domain experts to identify financial distress by using Multiple-Criteria Decision Analysis (MCDA) [39] or Fuzzy Logic [18] techniques.

Most financial distress prediction models focus on listed companies and use publicly available balance sheet data to create financial ratios as independent variables. In addition, studies have tried to improve the financial distress prediction models by adding more data, such as market data, non-financial indicators, and textual data. Market data includes stock returns, stock volatility, credit ratings, and indicators that compare the company's performance with the market [4, 24, 29]. Besides, several studies have come up with other non-financial indicators related to, for example, the age of the company, governance characteristics, and the ownership of shares [28, 37]. In addition, multiple studies apply NLP techniques to incorporate textual analysis into their models. Wang [25] used sentiment analysis scores based on public opinion combined with financial indicators to predict financial distress and Huang et al. [29] extracted data from annual reports in combination with sentiment analysis to improve their ML models.

2.3. Early Warning Systems

Although financial distress prediction models have been widely researched, discussion about the use of these models in the context of a EWS is limited. Most of the studies found do not mention EWS, and when they do, they mainly focus on establishing an EWS by only introducing a financial distress prediction model. However, some studies address other aspects of EWS. Firstly, Wen et al. [4] introduces a data architecture regarding the extraction and prepossessing of big data for an EWS. In addition, Koyuncugil and Ozgulbas [19] provides a data flow framework that considers roadmaps for clients that can help them improve their risk profile. Finally, the research from Kaluðer and Klepac [18] was the only study found that conducted research on an EWS implemented in an individual financial institution. They proposed a univariate analysis approach for early warning triggers and included early warning trigger data in a financial distress prediction model based on Fuzzy Logic.

In conclusion, the literature lacks research on implementing financial distress prediction models based on ML techniques applied to a financial institution's individual level. This study addresses the gap in the literature by conducting a case study at ING. Firstly, we build on the univariate analysis of early warning triggers introduced by Kaluðer and Klepac [18] by analysing the relationship between early warning triggers and the watchlist classification status of clients. Furthermore, this study introduces a new financial distress prediction model with a different dependent variable to predict financial distress. It also explores the use of other data sources for the independent variables. The current ML models focus mainly on the data of publicly listed companies, but there are no case studies that implement these models at an individual institution level utilising internal data.

3

Data

3.1. Data Collection

3.1.1. Independent Variables

The data used in this research was provided by ING, which means that not all the data can be disclosed due to confidentiality. Therefore, only general descriptions of variables are given, and no specific data of clients will be shared. ING has three data pipelines that extract and process the internal and external data used for ARIA. The first pipeline extracts internal data from the risk and financial systems within ING, which include, for example, the Probability of Default (PD). This data is uploaded monthly into ARIA to calculate the internal early warning triggers. The other two pipelines are used to extract the external data. One pipeline is used to extract and process news articles from the Financial Times [40] and Google News [41]. The other pipeline extracts data from Refinitiv [42] that provides market data of related financial products, such as the pricing of credit default swaps. These pipelines make it possible to calculate the external triggers in real-time.

Three tables from the ING databases have been extracted that consist of the historical data from May 2021 to September 2022. The first table has the clients' monthly data, including their status and related internal data like the PD. The second and third tables have the records for every time an internal or external trigger was flagged. These three tables were merged to create the final data set. The majority of the features are binary variables based on whether a trigger has been flagged (1) or not (0). Next to these features, the actual values of the underlying data of these triggers are used as independent variables to experiment with, which data results in higher model performance. Table 3.1 provides an overview of all the features and a general description of when the triggers are flagged. For instance, the PD is calculated by the internal financial models of ING. When the PD significantly changes compared to the previous month, then a trigger is raised by the ARIA system. Another example is Bankruptcy (BNK) trigger which

Table 3.1: Independent variables overview

Source	Symbol	Description	Type
Internal triggers	CVNT	Trigger related to the Convenant schedule	Binary
	LE	Flagged when the Limit Excess exceeds a certain threshold	Binary
	DPD	Flagged when the Days Past Due exceed a certain threshold	Binary
	RWA	Flagged when the Risk-Weighted Asset changes with a certain threshold	Binary
	EAD	Flagged when the Exposure at Default changes with a certain threshold	Binary
	IFRSS	Flagged when the International Financial Reporting (IFRS) status changes	Binary
	RUD	Flagged when Reviews Upcoming date is smaller than the Reporting Date	Binary
	ROD	Flagged when Reviews Upcoming date is larger than the Reporting Date	Binary
	ESRT	Trigger related to the ESR Transaction Outcome	Binary
	SS	Flagged when the sanction status changes	Binary
	LGD	Flagged when the Loss Given Default changes with a certain threshold	Binary
	RCF	Flagged when the change in the total Outstanding Amount of Revolving Credit Facilities	Binary
	IR	Flagged when the Internal Rating changes	Binary
	FBS	Flagged when the Forbearance Status changes	Binary
	PD	Flagged when the Probability of Default changes with a certain threshold	Binary
External triggers	BNK	Flagged when there are relevant news articles related to Bankruptcy	Binary
	MA	Flagged when there are relevant news articles related to Merger and Acquisition	Binary
	FRD	Flagged when there are relevant news articles related to Fraud	Binary
	ECC	Flagged when there are relevant news articles related to Environment and Climate Change	Binary
	SNC	Flagged when there are relevant news articles related to Sanctions	Binary
	HR	Flagged when there are relevant news articles related to Human Rights	Binary
	EQU	Flagged when Equity prices change with a certain threshold	Binary
Internal data	AVG PD	Average Probability of Default value of a client	Numeric
	TOTAL RWA	Total Risk-Weighted-Asset value of a client	Numeric
	AVG LGD	The average Loss Given Default of a client	Numeric
	AVG EAD	The average Exposure at Default of a client	Numeric
	TOTAL DPD	The accumulated Days Past Due Value	Numeric
	MAX DPD	The maximum Days Past Due value	Numeric
	TOTAL ALLOC LIMIT	The Total Allocated Limit to a client	Numeric
	MAX IFRS	The credit risk stage defined by the IFRS9 accounting standards	Categorical
	SEC	The sector of the client	Categorical

is raised when the NLP model detects news articles related to bankruptcy for a particular client.

3.1.2. DEPENDENT VARIABLES

As mentioned in section 2.2, many financial distress prediction models exist, but there is no general consensus on what defines a financially distressed client. In this research, we define financial distress as a situation where a client's risk profile significantly deteriorates so that the bank should take some action to ensure that a client can meet its financial obligation in the future. This definition includes clients that have defaulted and clients who, for example, might require extra monitoring or forbearance measures to avoid financial disruptions. If the model predicts that the client is in financial distress, the advice is given that this customer should be put on the watchlist.

For the watchlist classification, three different client credit statuses are considered. Firstly, there are clients with a regular status that paid their loans on time and have not been identified as carrying any significant risks. If the client's risk profile considerably deteriorates, then a risk manager could assign the client a watchlist status. Finally, if a client can no longer make his payment obligation to ING, then the client receives a default status. Legally, this is the case when the client did not make payments for ninety subsequent days.

A change in client status is referred to as a migration from one status to another. A negative migration is a change from a regular status to a watchlist or default status, and a positive migration is the other way around. Ideally, a client has always been watchlisted before it migrates to de-

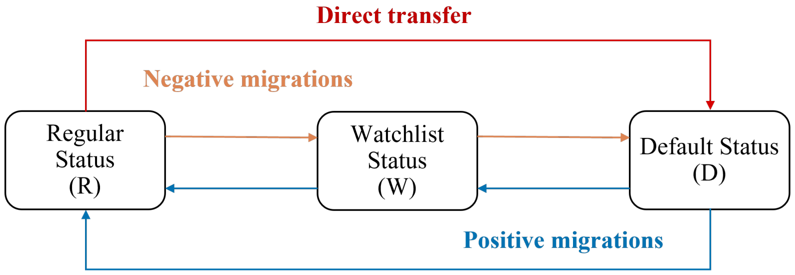


Figure 3.1: Client status migrations

Table 3.2: Migration frequency

From/To	R	W	D
R	813786	3788	2761
W	2435	24531	155
D	1662	24	20980

fault status, but in reality, there are still clients that default without ever receiving the watchlist label. In that case, we have a so-called direct transfer (Figure 3.1). Furthermore, being watchlisted does not necessarily lead to a migration to a default status because if the correct measures are taken, the loan status can be changed back to a regular one. In a few exceptional cases, it is also possible for a loan to be transferred from a default status to a watchlist status.

Two sources of data are considered for the data labelling. Firstly, there is the historical data of the clients that are watchlisted by the current manual procedures. Secondly, there is data on clients that have defaulted in the past, including instances that might have never been watchlisted beforehand. This research uses negative migrations as the independent variable for the financial distress prediction model.

3.2. EXPLORATORY ANALYSIS

3.2.1. WATCHLIST AND DEFAULT STATUS AND MIGRATION ANALYSIS

The first analysis looks at the number of migrations between June 2021 and November 2022. Table 3.2 presents the frequency for each migration combination between a regular, watchlist, and default status. Naturally, positive and negative migrations make up a small part of the total migrations as the client status remains unchanged most of the time. As a result, there is a significant class imbalance when considering negative migration as the independent variable. Moreover, the number of direct transfers is relatively high since only 5% of the negative default migrations came from a watchlist status. So, the watchlist procedure could be improved considerably by detecting these clients.

Besides, Figure 3.2 depicts a line graph with the total number of clients with a watchlist or default status. Generally, there are more watchlisted clients than clients in default, but interestingly, this gap has faded over time. This is mainly because the number of clients with a watchlist status has gradually decreased since June 2021. In addition, when looking at Figure 3.3, we see

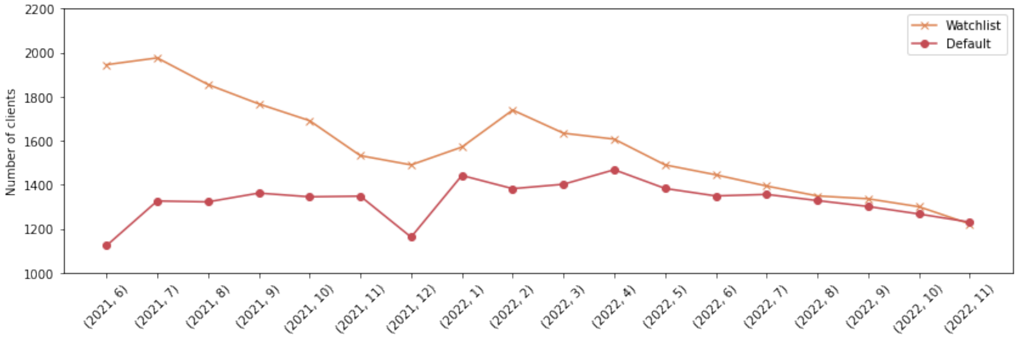


Figure 3.2: Watchlist and default time series

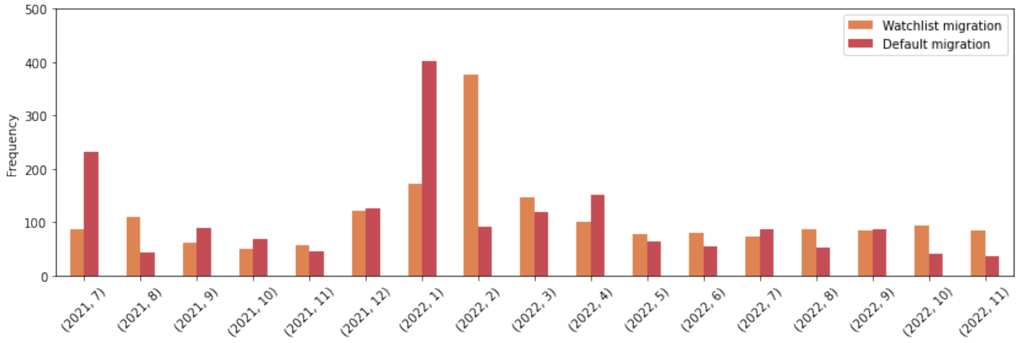


Figure 3.3: Negative migrations

that there has been a spike in watchlist migrations in February 2022. This was primarily due to the invasion of Ukraine, resulting in more than two hundred watchlist classifications of Russian clients.

Moreover, there was a sharp increase in defaults in January 2022. Presumably, this is due to missing default data, as there is a slight dip in default when looking at Figure 3.2. Also, when looking at the specific migration instances, many clients migrate from a default status for one month, and then the next month, they migrate back again. Therefore, for these clients, the status is changed to default resulting in a similar number of default migrations as in the other months (See Appendix B).

Finally, the reasons for classification are analysed to gain more insight into the current watchlist classification procedure. As mentioned in section 1.2, account managers must provide predefined reasons for the classification. These reasons are counted and plotted in Figure 3.4. This figure shows that most watchlisted classifications were due to bad news, poor industry outlook, or a material decline in profitability. This suggests that specific features may hold greater significance in identifying watchlisted clients. For instance, considering that many clients end up on the watchlist due to negative news, we would expect that external triggers based on online news articles would prove effective in detecting such clients.

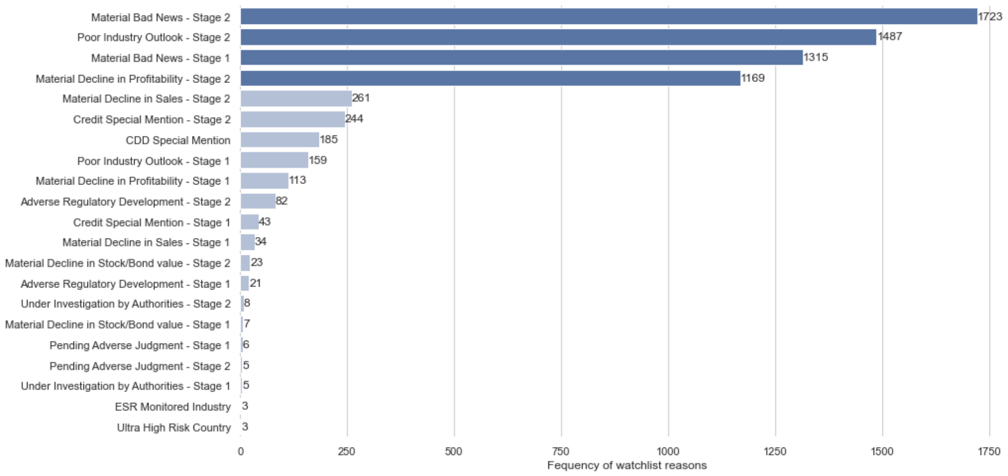


Figure 3.4: Frequency of watchlist reasons

3.2.2. EARLY WARNING TRIGGERS

The statistical relationships between the ARIA triggers and the negative migrations are measured to analyse to what extent the triggers individually would be able to predict financial distress. As a result, the performance of the newly proposed WL trigger can be compared with the already-used ARIA triggers. In addition, Babel et al. [1], and Kaluđer and Klepac [18] proposed several metrics to assess the quality of early warning triggers for detecting nonperforming loans (See Appendix C.1). Inspired by these metrics, the following statistics were used to evaluate the relationship between the ARIA triggers and negative migrations.

- **Migration sensitivity**: the fraction of negative migrations which had a certain trigger raised within six months before.
- **Trigger precision**: the fraction of raised triggers for which there were negative migrations within the next six months.
- **Time lag**: the number of months between a trigger event and a negative migration.

While many metrics could be used to evaluate the performance of early warning triggers, we primarily focus on these three metrics because they describe the properties the WL trigger should have related to its predictive performance and earliness. Appendix C.2 describes how the early warning triggers are calculated and other metrics that could be used to evaluate the triggers. The coming sections describe the results and interpretation of the metrics calculated for the period between January 2021 and September 2022. In addition, time windows of six months are used to determine if a trigger or negative migrations occurred. A six-month time window is used to ensure that there are enough time windows within the selected period to calculate the metrics. When more historical data is available, more research could be done to see how the metrics change for different time window sizes.

MIGRATION SENSITIVITY

The migration sensitivity indicates how prevalent a trigger is among the watchlisted clients. For example, if a trigger has a sensitivity of 50%, then half of the watchlisted clients had that trigger raised within a range of six months in advance. Figure 3.5, displays a graph of the sensitivity for each trigger with their corresponding 99% confidence intervals. This graph shows that RWA occurs the most often among financially distressed clients, while the external triggers (EQU, BNK, FRD, SNC, HR, ECC) occur infrequently. The WL trigger should aim for a high sensitivity because it should be able to detect as many negative migrations as possible. However, increased sensitivity could be achieved by always raising the trigger, which would make the trigger redundant. Therefore, trigger precision should be taken into account to make sure that a raised trigger is actually meaningful.

TRIGGER PRECISION

Trigger precision indicates how well a trigger can separate financially distressed clients from regular clients. A high precision would mean that if a client has a given trigger raised, there is a high probability that the client will end up with a watchlist or default status. Ideally, a trigger

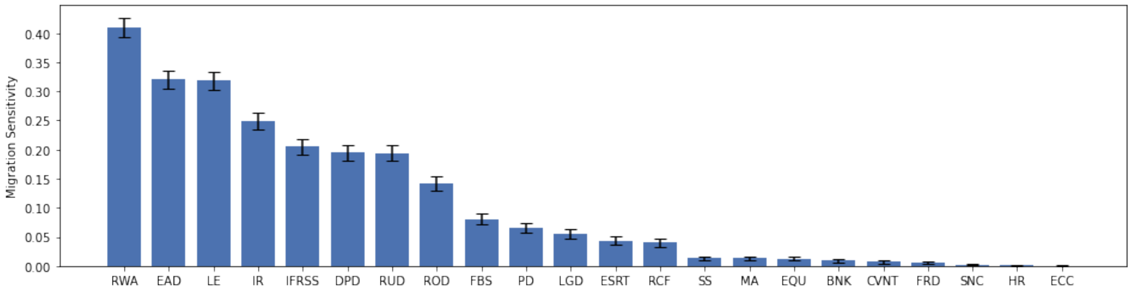


Figure 3.5: Migration sensitivity with 99% confidence interval

would have a high sensitivity and precision to have a high predictive performance on detecting distressed clients. Figure 3.6 provides a bar graph of the precision with the corresponding 99% confidence intervals. The PD has a relatively high precision of 20%, which means that when observing this trigger, there is a 20% probability that a client will migrate to a watchlist or default status in the next six months.

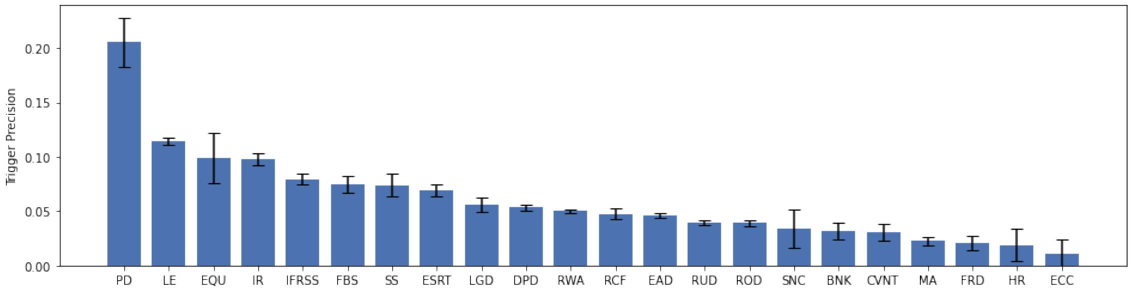
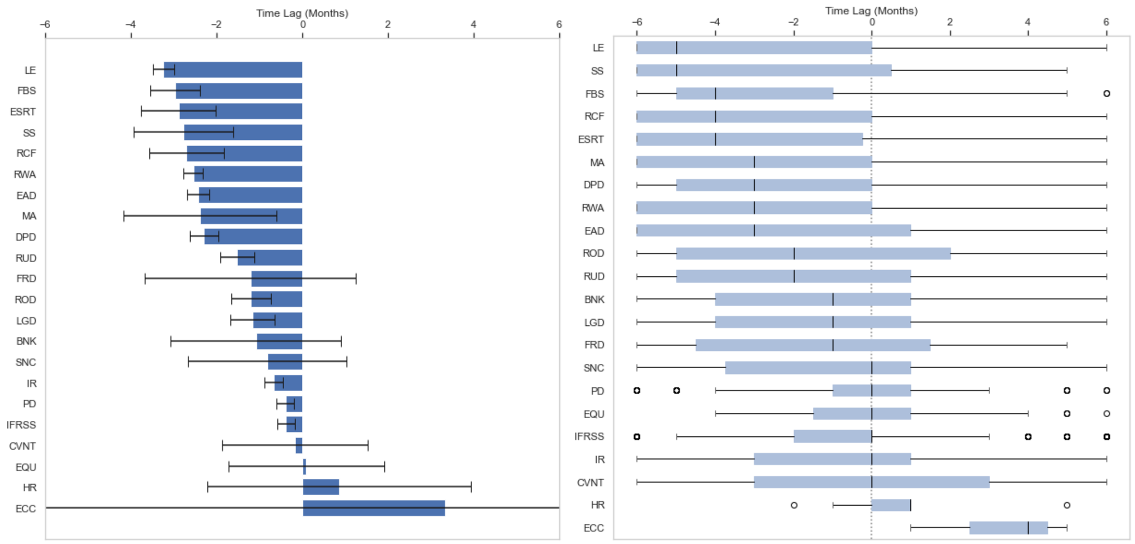


Figure 3.6: Trigger precision with 99% confidence interval

TIME LAG

The sensitivity and precision give insight into the predictive performance, but it does not shed light on the timeliness of the triggers. Therefore, the distribution of the number of triggers raised before a watchlist classification is analysed to see how early a trigger can detect a client in distress. Appendix C.4 provides an overview of the time lag distributions for each trigger. Figure 3.7 shows a box plot and the average time lag with a 99% confidence interval of the distributions. However, these graphs do not paint the complete picture as it does not consider the dispersion and skewness of the time lag data. From the charts, we can conclude that most triggers are early, except for the FRD, BNK, SNC, CVNT, EQU, HR, and ECC. However, these triggers have large confidence intervals due to a small sample size. More significant conclusions could be made about their earliness when more data is available.

Conclusively, we aim to design a WL trigger which has high predictive performance at detecting negative migrations while at the same time being able to make these predictions as early as possible. The migration sensitivity and trigger precision measure the predictive performance, and the average time lag measures the earliness. Moreover, when designing a WL trigger, the importance of these metrics compared to each other need to be considered. For example, aiming for a higher precision might come at the cost of lower sensitivity and time lag and vice versa.



(a) Average time lag with 99% confidence interval

(b) Time lag box plot

Figure 3.7: Time lag analysis

4

METHODOLOGY

4.1. SUPERVISED LEARNING

Chapter 2 shows that the majority of the found literature uses a supervised learning approach for their financial distress prediction model. Supervised learning is a ML paradigm where the labels we try to predict are assumed to be known [43]. This research focuses on the classification models commonly used for the early detection of financial distress, which includes LDA, LR, SVM, DT, ensemble learning methods, and ANN.

4.1.1. LINEAR DISCRIMINANT ANALYSIS

LDA is a ML technique used for classification. The method works by identifying a linear combination of the independent variables that are able to separate the different classes. The goal is to project the data onto this linear combination in such a way that the separability between the classes is maximized while, at the same time, the variation within the classes is minimized [44]. Altman's Z-score function (See Equation 4.1) is a popular implementation for financial distress prediction using LDA where he predicts bankruptcy using five financial ratios: working capital / total assets (A), retained earnings / total assets (B), earnings before interest and taxes (EBIT) / total assets (C), market value of equity / total liabilities (D), and sales / total assets (E) [45].

$$\text{Z-score} = 1.2A + 1.4B + 3.3C + 0.6D + 1.0E \tag{4.1}$$

4.1.2. LOGISTIC REGRESSION

LR is a statistical method used for binary classification [46]. The logistic function (Equation 4.2) maps a linear combination of the independent variables and weight parameters into a value between zero and one, representing the probability of the binary outcome. These weights are obtained through Maximum Likelihood Estimation, which can be solved by optimization algorithms such as Gradient Descent or Newton's method. Numerous studies have researched the use of logistic regression models for financial distress prediction [28–31, 34].

$$\Pr(\mathbf{y} = \mathbf{1}|\mathbf{X}; \boldsymbol{\beta}) = \frac{1}{1 + \exp(-\mathbf{X}\boldsymbol{\beta})} \tag{4.2}$$

4.1.3. SUPPORT VECTOR MACHINE

SVM is a ML model developed by Boser et al. [47] which can be used for both classification and regression problems. SVM tries to find a hyperplane that is able the maximize the margin between the closest data points, which are also known as the support vectors. Equation 4.3 shows the optimization problem that needs to be solved where the optimization function ensures that the margin is maximized, and the constraint ensures that the data points are correctly classified. In addition, this optimization can be improved by allowing a soft margin which permits misclassifications close to the margin in order to reduce overfitting when training the model. Furthermore, kernel functions make it possible to map the data points into higher dimensions so that it is possible to find margins that can capture non-linear relationships between the independent and dependent variables.

$$\min_{\mathbf{w},b} \quad \frac{1}{2}||\mathbf{w}||^2 \tag{4.3}$$

$$\text{subject to} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \ldots, n \tag{4.4}$$

4.1.4. DECISION TREE

The DT is a ML algorithm for classification that recursively splits the data to separate the data into different classes. The data is divided by conditions based on the feature that best separates the data. This property can be measured using different purity measures such as entropy, Gini impurity or chi-square tests [48]. Pruning techniques or setting a maximum depth can be used to avoid overfitting the model [49]. Decision trees have been widely used for financial distress prediction due to their high interpretability [19, 27, 34, 50].

4.1.5. ENSEMBLE LEARNING

Ensemble learning is a ML technique that combines multiple models to improve the predictive performance. Studies have shown that ensemble learning models can perform better than individual models by combining the strengths of multiple models and by reducing overfitting [51]. Bagging and boosting are two standard methods for constructing ensemble models.

Bagging involves training multiple models on random sample subsets of the training data [52]. Then, the models' predictions are combined into a single prediction by, for example, taking the average or a majority vote. RF is a ML model that combines several DTs using the bagging method [53]. RF constructs multiple trees based on a random subset of the features to reduce

Table 4.1: Confusion matrix

		Predicted	
		1	0
Actual	1	True Positive (TP)	False Positive (FP)
	0	False Negative (FN)	True Negative (TN)

the correlation between the trees, resulting in less overfitting. After the training, the prediction of the different trees is aggregated by taking a majority vote in the case of a classification problem.

In contrast to bagging, boosting involves training a sequence of weak models where the training samples are weighted. A weak learner is a model that performs a little better than random guessing. By combining many weak learners, a strong learner can be constructed with a better predictive performance [54]. After each time a weak learner is constructed, new weights are assigned to the training samples so that previously misclassified instances are emphasized when training the new weak learner. When all the weak learners are constructed, the predictions are aggregated by giving more weight to more accurate learners. There exist many ML models based on the boosting technique, such as AdaBoost, CatBoost, GBM, LightGBM, and XGBoost.

4.1.6. ARTIFICAL NEURAL NETWORKS

ANN is ML model that consists of a network of neurons separated by multiple layers [55]. Each neuron takes the neurons in the previous layer as a linear input which is then transformed using an activation function. The weights of the linear functions are obtained through a process called backpropagation, where the weights and biases are iteratively adjusted backwards through each layer. During this process, the model minimizes the difference between the model output and the actual output using a loss function. This optimization problem can be solved using the Gradient Descent algorithm.

4.2. EVALUATION METRICS

Several evaluation metrics are considered to measure the predictive performance of the financial distress prediction models. Table 4.1 provides an overview of the confusion matrix, which is used to compare the predicted values of a model with the actual values. Based on this table, the evaluation metrics are calculated.

Accuracy is a standard performance metric which shows the fraction of correct classifications compared to the total number of classifications made (Equation 4.5). Accuracy is one of the most commonly used metrics to evaluate financial distress prediction models [24, 28, 31, 35]. While accuracy is quite an intuitive metric, it can be misleading when there is a class imbalance. In our case, only 0.8% of the target data are negative migrations, which means that if the model would always guess that there are no negative migrations, the model would still have a high accuracy of 99.2%.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.5}$$

AUC is another popular metric for evaluating binary classification models for detecting financial distress [29, 56, 57]. The AUC is determined by plotting the Receiver Operating Characteristic (ROC) curve and then calculating the area beneath the curve [58]. The ROC curve is constructed by plotting the True Positive Rate (TPR) (Equation 4.6), also known as the sensitivity or recall, against the False Positive Rate (FPR) (Equation 4.7) for different threshold values. Although AUC is less sensitive to the class imbalance problem than accuracy, it can still be too optimistic when the majority class is much bigger than the minority class.

$$TPR = \frac{TP}{TP + FP} \tag{4.6}$$

$$FPR = \frac{FP}{FP + FN} \tag{4.7}$$

A couple of studies use the F1 score [29, 59, 60] or the Matthews Correlation Coefficient (MCC) [61, 62] to evaluate their models to deal with the class imbalance issue. The F1 score deals with class imbalance by taking the harmonic mean of the precision and recall (Equation 4.8). The MCC metric can be interpreted as the correlation between the predicted and observed classifications (See Equation 4.9)[63].

$$F1\,score = \frac{2TP}{2TP + FP + FN} \tag{4.8}$$

$$MCC = \frac{TP \times FN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.9}$$

Finally, the sensitivity or TPR (Equation 4.6), and precision (Equation 4.10) are commonly used to supplement the aforementioned metrics as they provide more insight into the performance of the model in regard to the positive class [29, 32, 36].

$$Precision = \frac{TP}{TP + FP} \tag{4.10}$$

4.3. Model Explainability

Explainable AI (XAI) refers to the practice of making the decision-making processes of ML models transparent and understandable to human beings [64]. XAI is a crucial part of financial distress prediction modelling. Firstly, risk managers should be able to communicate to their clients the reasons why they have been put on the watchlist, as this will have consequences on their creditworthiness and future relationship with the bank. Secondly, model explainability helps identify the causes for watchlist classification so that according actions can be taken. Finally, it can be used to validate the predictions of the models, which helps establish trust in the decision-making of the models.

SHAP values were introduced by Shapley [65], who studied the contribution of players to the outcome of a cooperative game using game theory. In the case of ML, the features represent the players and the outcome of the output of the model. The SHAP values are determined by calculating the contribution of each feature [66]. This is done by creating all possible subsets of the features and calculating the difference in output between the subset that includes the feature and the combination that does not include the feature. The difference in output is then weighted by the number of possible combinations that include the feature. Finally, the contributions of each feature are summed across all possible combinations to calculate the SHAP value for that feature which represents the change in the model's output when that feature is included in the model.

5

Experimental Set-up

5.1. Model Approach

This research aims to develop a watchlist classification model based on a financial distress prediction modelling approach that incorporates the historical data of early warning triggers. Based on the analysis of the problem background and the exploratory data analysis, requirements have been formulated that define the design of our financial distress prediction model. These requirements can be summarized as follows:

- **Negative migrations**: In this research, we define a client in financial distress as a client with a watchlist or default status. Since we focus on watchlist classification, we are mainly interested in predicting negative migrations. Positive migrations are left out of the scope, but extending the model by including these types of migrations could be interesting for future research.
- **Sequentiality**: we try to predict future events (migrations) based on historical data. This means that features must be designed accordingly, and the evaluation process must consider the sequential order of events.
- **Earliness**: the aim of the financial distress prediction model is not necessarily to identify the exact moment of a migration. If the model can detect financial distress earlier than the negative migration would happen, the prediction is even more valuable because actions can be taken further in advance.
- **Error costs**: a tradeoff between the costs associated with false positives and false negatives needs to be considered. The cost of a false positive is much lower than that of a false negative since a false positive only leads to redundant monitoring of clients, whereas a false negative leads to defaults that could be prevented.

Figure 5.1 provides an example of a fictional client that migrated to a watchlist status. In this example, the client obtained four triggers (BNK, DPD, and DPD) in a time window of twelve months before the watchlist classification. Based on this historical data, we try to predict whether a client will migrate to a watchlist or default status in the next month. In this case, the time gap is one month, which means we only try to predict the migrations one month in the future. Moreover, we add some flexibility to the model by considering a target window. We do not necessarily want to predict the exact moment of the migrations because if the model detects financial distress earlier than when a migration occurred, the prediction should still be considered correct. Therefore, the five months before migration are also encoded as financial distress. In summary, we define the following parameters for the financial distress prediction model:

- **Time window:** the historical period used as input for the financial distress prediction model.
- **Time gap:** The period between the time window and migration that determines how far we want to predict in the future
- **Target window**: the period before a migration where we try to detect financial distress.

The following sections discuss the preprocessing of the data, the model development, and the model evaluation process of the financial distress prediction model. Figure 5.2 provides a flowchart which summarises the steps taken for each of these three phases.

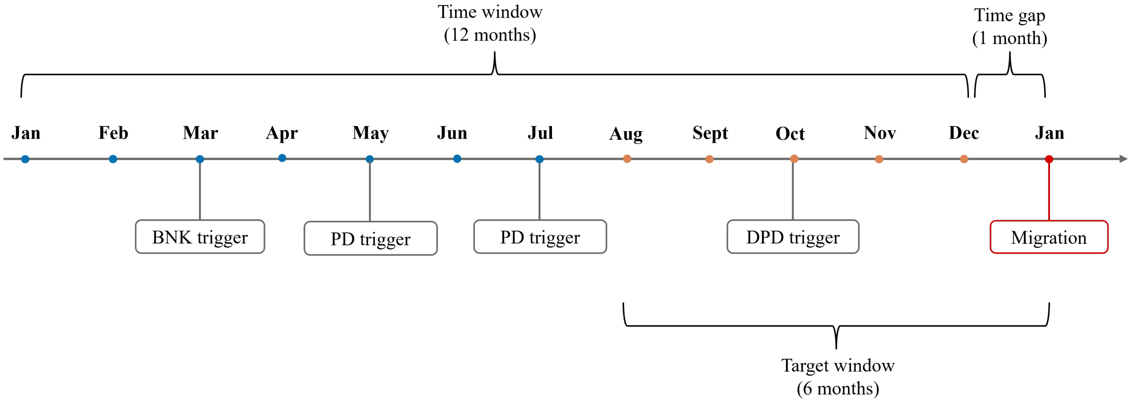


Figure 5.1: Example of early warning trigger history of a client in financial distress

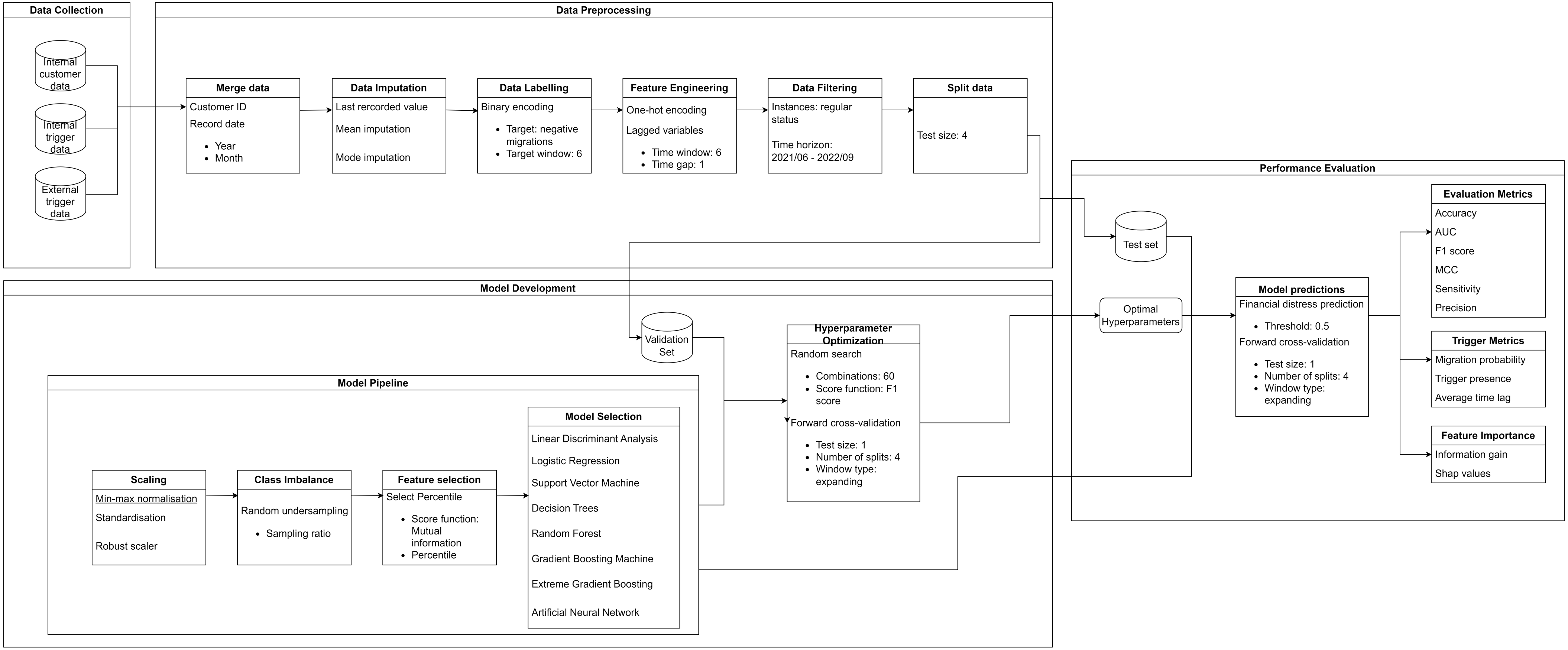


Figure 5.2: Flow chart of the financial distress prediction model

5.2. Data Preprocessing

Data preprocessing is the process related to collecting, cleaning, and manipulating raw data so that it can be processed by the ML models. Our financial distress predictions model involves several preprocessing steps, such as merging data from multiple sources, imputing missing values, creating new features through feature engineering, labelling data for classification tasks, and filtering data to remove irrelevant or redundant information.

5.2.1. Merge Data

Three data sets were used, including the internal client data and the recordings of the internal and external triggers. The internal client data consists of details about outstanding loans of WB clients and the associated risk measurements such as the PD and risk ratings. In addition, this table includes the status of a client, such as a default or a watchlist status. However, many values were missing in certain months from the data extract obtained from the pipeline, so the status of the clients was eventually extracted manually from a different source. The trigger data includes information about the types of triggers raised at a given moment. A binary table was created from this data to determine if a trigger was raised in a particular month, with 1 denoting a raised trigger and 0 denoting no trigger. After that, the transformed table is merged with the internal data table using the customer id and reporting date as a unique key.

5.2.2. Data Imputation

Next, data imputation was used to fill in the missing values for the dependent variables. In the case of missing data for features based on triggers, it is assumed that no trigger was raised, so for these instances, all the values are imputed as 0. For the features based on the internal data, missing values were filled using the last recorded value. If there were no recorded values for a client, then the overall mean or mode was used for the numerical or categorical variables, respectively.

5.2.3. Feature Engineering

The categorical features related to the IFRS stage and sector of a client were transformed using one-hot encoding. One-hot encoding converts categorical data into a binary representation where each category is represented by a binary column with a value of 1 indicating the presence of that category and 0 indicating its absence.

Afterwards, lagged features are created for each feature that can change over time. Lagged features are a set of variables derived from previous values of sequential data that can be used as predictors to model the behaviour of the data in the future. The construction of the lagged features depends on the time lag and time window. For the base model, a time window of six months and a time lag of one month is used. In the literature, multiple studies use lagged features for default or financial distress prediction models. These studies use different time windows varying from multiple months [67] to a couple of years [68, 69]. Since we have limited data, we need to consider a tradeoff between the time window size and the number of training samples. The number of samples decreases when increasing the time window because more historical data is needed to construct the lagged features. A time window of six months was chosen to include a reasonable amount of historical data for each prediction while keeping enough data to train the models. Also, the effect of different window sizes is investigated in the experimentation part of this research. The time gap was set to one month because the trigger data is always available one month in advance.

Lastly, the original features are dropped when the lagged variables are created since they include foreknowledge about the month we try to predict. Table 5.1 provides an example of how the lagged features are created for the scenario described previously in Figure 5.1. One-hot encoding, in combination with creating lagged features considerably, explodes the number of features. Initially, there were thirty features, and after feature engineering, this number increased to 198 features in total. For this reason, feature selection is used to reduce the number of features (See Section 5.3.3).

5.2.4. DATA LABELLING

Subsequently, the binary dependent variable for predicting distress is created based on the status of a client. Firstly, the migrations between each month are derived from the client's status. We defined financial distress as a negative migration from a regular to a default (R,D) or watchlist (R,W). As a result, the negative migrations are encoded as 1, while all other migrations are denoted by 0.

Next, the target window needs to be considered because predicting financial distress before a migration is also considered a correct prediction. Therefore, the months prior to a negative migration, dependent on the size of the target window, are encoded as 1. Table 5.2 provides an example of how the data is labelled based on a target window of three.

5.2.5. DATA FILTERING

Finally, data points were filtered out of the data set. Firstly, all migrations that do not transfer from a regular status are left out because we are only interested in predicting if a negative migration will occur if a client does not have a watchlist or default status yet.

Moreover, only the data between January 2021 and September 2022 was included because only during this period was there enough trigger data available for the prediction model (See Appendix D). Throughout the life of the ARIA application, triggers are redesigned, and new ones

Table 5.1: Example of lagged variables

Trigger/lag	Lag 12	Lag 11	Lag 10	Lag 9	Lag 8	Lag 7	Lag 6	Lag 5	Lag 4	Lag 3	Lag 2	Lag 1
PD	0	0	0	0	1	0	1	0	0	0	0	0
BNK	0	0	1	0	0	0	0	0	0	0	0	0
DPD	0	0	0	0	0	0	0	0	0	1	0	0
LGD	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...

are introduced. As a result, some triggers do not have much historical information. For that reason, the triggers that do not have a track record of at least a year are also excluded from the data set. Finally, considerations were made to exclude the watchlist migrations of Russian clients due to being an outlier (See Section 3.2.1). Still, it was decided to keep these instances in the data set as they did not significantly change the performance of the models. Consequently, the data set consists of 548.210 monthly data points that make up 38.316 different clients in total.

5.3. MODEL DEVELOPMENT

The model development phase focuses on creating the model with the highest performance in predicting financial distress. This involves dealing with the class imbalance, normalising the values to a similar scale, selecting the most relevant features, choosing the most suitable ML algorithms, and finding the best-performing hyperparameters for these models.

5.3.1. CLASS IMBALANCE: RANDOM UNDER-SAMPLING

There is a class imbalance in the data set because, the majority of the time, the status of a client remains unchanged. As a result, there are only a few negative migrations, which comprise about 0.8% of the data, compared to the overall number of migrations from a regular status. Therefore, Random Under-Sampling (RUS) was used to balance the class distribution by randomly selecting a subset of instances from the majority class. A disadvantage of RUS is that valuable information might get lost when only using a subset of the data, which is not the case for other sampling techniques such as Random Over-Sampling (ROS) and SMOTE [32]. A significant advantage, however, is that RUS significantly decreases the size of the data set, resulting in reduced computational time for the training of the models. When applying RUS, the ratio between the minority and majority classes must be set. In this research, this ratio is one of the hyperparameters validated for different values to find the optimal ratio resulting in the best performance.

5.3.2. SCALING: MIN-MAX NORMALISATION

Data scaling is the process of rescaling numerical data to a standard range to improve its comparability and reduce the impact of different units or scales. Three different data normalisation techniques were considered:

- **Min-Max Normalisation**: scales the data to a fixed range of zero to one by subtracting the minimum value and dividing by the range.

Table 5.2: Example label encoding (R: regular status, W: watchlist status, D: default status)

	2021/12	2022/01	2022/02	2022/03	2022/04	2022/05	2022/06	2022/07	2022/08	2022/09
Client status	R	R	R	R	R	W	W	W	D	D
Migration (from, to)	n.a.	(R, R)	(R, R)	(R, R)	(R, R)	(R, W)	(W, W)	(W, W)	(W, D)	(D, D)
Label encoding	n.a.	0	0	1	1	1	0	0	0	0

$$\frac{x - x_{min}}{x_{max} - x_{min}} \tag{5.1}$$

- **Standardisation**: scales the data to have zero mean and unit variance, resulting in a standardised distribution with values centred around zero.

$$z = \frac{x - \mu}{\sigma} \tag{5.2}$$

- **Robust Scaler**: scales data by removing the median and scaling it according to the interquartile range, making it robust to the presence of outliers.

$$\frac{x - q_{50}(x)}{q_{75}(x) - q_{25}(x)} \tag{5.3}$$

These techniques were evaluated using the validation set to see which approach resulted in the best performance. Overall, Min-Max Normalisation and Standardisation outperformed the Robust Scaler, but the two methods had no significant performance difference. Min-Max Normalisation was used for the remainder of the research because the values are more interpretable as all the features based on the triggers are already a value between zero and one.

5.3.3. FEATURE SELECTION: MUTUAL INFORMATION

Feature selection is the process of selecting a subset of relevant dependent variables from a more extensive set of available features. This can enhance predictive performance by reducing overfitting and decreasing the computational time for model training. However, the number of features explodes when creating the lagged features because the total number of features is the length of the time window multiplied by the number of original features; so, a time window size of 6 results in a total of 240 independent variables.

The feature selection procedure used in our model is based on mutual information. Mutual information is a measure of the amount of information shared between two random variables [70]. It measures the reduction in uncertainty of one variable when the other variable by calculating the difference in Shannon Entropy (See Equation 5.4 and 5.5).

$$I(X;Y) = H(X) - H(X|Y) \tag{5.4}$$

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{5.5}$$

Feature selection based on mutual information was chosen because of the following reasons [71]:

- It can handle both continuous and discrete data. This makes it very suitable for our data set, consisting of categorical and numerical data types.
- It can capture any statistical dependency between features and the target variable, including nonlinear relationships. Other techniques, such as the F-test and correlation, are limited to measuring linear dependencies.
- It is computationally efficient and can be easily calculated for large datasets.

The top percentage of features with the highest mutual information is included in the financial distress prediction model. This optimal percentage of features is one of the parameters found for each model based on hyperparameter optimization.

5.3.4. MODEL SELECTION: SUPERVISED LEARNING

Due to time constraints, only a selection of supervised learning models used in the literature is tested in this research. These include LDA, LR, DT, SVM, RF, GBM, XGBoost, and ANN. In addition, a dummy model is tested that serves as a baseline comparison with the models above. This model is based on a random guess strategy, and it is used as a benchmark to determine if the more advanced models are better at predicting financial distress than random guessing.

Regarding the ANN models, two novel architectures with two and three layers were considered. These models use the rectified linear unit (ReLu) as activation functions, and the final activation function is a Sigmoid function which turns the prediction into a binary output. Besides, the dropout technique is used to reduce overfitting by ignoring some of the neurons during the training of the model. Finally, the number of neurons for each layer, the dropout rate, and the learning rate are determined using hyperparameter optimization. The ANN can be extensively fine-tuned, but due to limited time, only two ANN architectures were considered. For future research, it could be interesting to also explore including other techniques like weight regularization and early stopping to improve model performance. Also, more complex architecture like recurrent ANN or transformers could be considered.

5.3.5. HYPERPARAMETER OPTIMIZATION: RANDOM SEARCH

Hyperparameter optimization is the process of systematically searching for the best combination of hyperparameters to maximize the performance of a machine learning model. For this, ransom search was used because it is more computationally efficient than grid search [72]. Grid search exhaustively searches over all possible parameter combinations, while random search only explores a fixed number of random combinations. For each model, we explore 60 parameter combinations because, with 60 combinations, there is approximately 95% confidence that the found parameters are in the top 5% of best-performing combinations [73]. The optimal parameter combinations are selected based on their F1 score. The F1 score was used because it is a reliable metric for evaluating models with a class imbalance (See Section 5.4). The parameter spaces explored for each ML model can be found in Appendix F.1.

5.3.6. MODEL VALIDATION: EXPANDING-WINDOW FORWARD CROSS-VALIDATION

The approach for validating and evaluating the model needs to be considered to ensure that there is no overfitting. First, the data is split into a validation and test set. The validation set is used during the model development phase to select the best model configuration, while the test set provides an unbiased estimate of the model's performance on unseen data.

Secondly, since we are dealing with sequential events, the chronology of the data needs to be maintained to ensure temporal dependence between the data points when measuring model [74]. Therefore, an expanding-window forward cross-validation approach is used for model validation and evaluation [75]. Cross-validation is a technique that evaluates a model multiple times by separating the data into several folds where each fold is used once for testing, and the remaining data is used for training.

Cross-validation is used because it provides a more accurate estimation of the model performance than the hold-out approach, where you only use a single training and test set [76]. To ensure temporal dependence, a time series approach for cross-validation is used where the data instances are grouped in folds which represent the different months. These folds are then separated sequentially into a training and validation/test set. The negative migrations are roughly equally distributed over different months, so we do not need to address the class imbalance when creating the folds.

Finally, the training set is expanded for each cross-validation split by including the previous validation/test set. Both for the model validation and evaluation, four iterations with a test size of one fold are used to calculate the performance of the models. As a result, the metrics are calculated four times which are then averaged. Table 5.3 provides an overview of how the data sets are divided.

Table 5.3: Expanding-window forward cross-validation overview

2021/06	**...**	**2021/10**	**2021/11**	**2021/12**	**2022/01**	**2022/02**	**2022/03**	**2022/04**	**2022/05**	**2022/06**
Training			Validation							
Training				Validation						
Training					Validation					
Training						Validation				
Training							Test			
Training								Test		
Training									Test	
Training										Test

5.4. PERFORMANCE EVALUATION

The model's performance is evaluated based on two perspectives: as a ML model and as an early warning trigger. The performance of the model can be measured by metrics commonly used to evaluate ML models (See Section 4.2). These include:

- Accuracy
- Area Under the Curve (AUC)
- F1 score
- Matthews Correlation Coefficient (MCC)
- Sensitivity
- Precision

Besides, the performance can be compared with the other ARIA early warning triggers. For this, we use the metrics introduced in section 3.2.2 to see if the WL trigger could identify more distressed clients at an early stage. These include:

- Migration Sensitivity
- Trigger Precision
- Time Lag

5.4.1. MODEL EXPLAINABILITY: SHAP VALUES

This research focuses on the use of SHapley Additive exPlanations (SHAP) values to explain the watchlist classification. SHAP values make it possible to analyse the importance of features comprehensively by showing the contribution of each future on the performance of a model. In addition, SHAP is a model-agnostic approach which means that it can be used in combination with any ML model. Moreover, the use of SHAP values has already been extensively studied in the credit risk domain [77–80].

6

Results and Discussion

6.1. Experiments

The experimentation part focuses on testing the performance of the financial distress prediction models. This includes a comparison of the performance of each classification model and the influence of different configurations. The following research questions are defined for the following experiments:

- **Experiment 1 (E1)**: which supervised learning model best predicts negative migrations?
- **Experiment 2 (E2)**: how well can the model predict negative migrations compared to default and watchlist migrations alone?
- **Experiment 3 (E3)**: what influence does the time window size have on the model performance?
- **Experiment 4 (E4)**: what influence does the target window size have on the model performance?
- **Experiment 5 (E5)**: what influence does the time gap have on the model performance?

Firstly, the supervised learning models commonly used for financial distress prediction models are evaluated. These models include LDA, LR, DT, SVM, RF, GBM, XGBoost, and ANN. The models are compared using a consistent configuration, with a time window of six months, a time gap of one month, and a target window of six months. This configuration was chosen to ensure that enough data was available for training the models and that the models could be trained within a reasonable time.

Additionally, different target variables are considered. The base financial distress prediction model predicts negative migrations that consist of both watchlist and default migrations. To gain additional insight into the model's performance, the model is configured to predict watchlist and default migrations separately to determine which migrations are more easily predicted.

Furthermore, the relationship between the performance of the models and the time window is analysed. By increasing the time window, more historical information is used to predict the migrations, which could improve the prediction performance. However, this research is limited to the historical data available, which means that increasing the time window reduces the number of training samples. Consequently, we expect to find a tipping point in model performance due to a tradeoff between the time window and the size of the training set.

Besides, the relationship between the target window and the average time lag of models is investigated. The target window adds flexibility to the financial distress prediction model by considering the model's timeliness. Therefore, we expect the models to perform better when the target window increases. Also, a larger time window size could result in a higher average time lag because the model has the incentive to make predictions earlier.

Finally, the influence of the size of the time gap on the model performance is analysed. We expect the model performance to decrease for a larger time lag because it is harder to predict further into the future. Therefore, a tradeoff needs to be made between a model's earliness and prediction performance.

6.2. EXPERIMENTAL RESULTS

6.3. RESULTS: MODEL PERFORMANCE (E1)

For each experiment configuration, model tuning is used to find the optimal hyperparameters. Appendix F provides an overview of the optimal hyperparameters found for each model configuration. Figure 6.1 illustrates the highest F1 score found for each ML model based on their optimal hyperparameters. These F1 scores do not provide the actual performance of the models since this needs to be measured with unseen data. But due to time constraints, the experiments are only conducted with the best-performing model using the validation set to avoid bias when evaluating the performance. In this case, RF has the best performance. Therefore, the RF model is used to compare the performance of the different migration approaches, time windows and target windows.

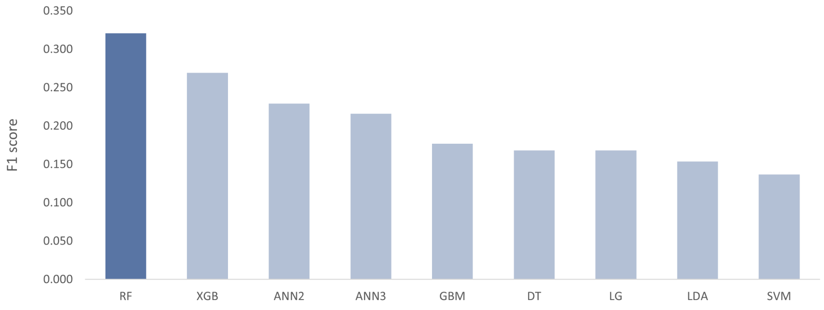


Figure 6.1: Model validation F1 score

In addition, mutual information was used to select the most relevant features for the model. Appendix E shows the calculated mutual information values for each lag feature with the target variable. The mutual information indicates that the features based on the internal customer data have the strongest relationship with negative migrations, while the early warning triggers have a relatively weak relationship. Consequently, the models will consist mainly of features based on the internal data after feature selection. From this, we can conclude that it is better to include the data on which the triggers are based instead of the triggers themselves. However, a limitation of mutual information is that it does not consider the interaction effects between features on the predictive performance. Therefore, the SHAP values are also considered in the next section to determine the overall impact of features on model performance.

Ten different classifiers were evaluated using six evaluation metrics, shown in Table 6.1. Among the models considered, the RF model performed the best with an F1 score of 0.197, an AUC of 0.893, and precision and sensitivity of 0.119 and 0.582, respectively. These results indicate that the RF model can distinguish between positive and negative instances and can correctly identify a large proportion of true positives while minimizing false positives. Other ensemble models that performed reasonably well include Gradient Boosting Machine (GBM) and XG-Boost, which had F1 scores of 0.096 and 0.161, respectively. However, these models had lower precision and sensitivity than the RF model resulting in lower overall performance.

The Decision Tree (DT), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), and Logistic Regression (LG) models had lower F1 scores, accuracies, and AUC values compared to the top models. Artificial Neural Networks (ANNs) with two and three hidden layers had low F1 scores of 0.050 and 0.055, respectively, indicating poor performance. This is probably due to overfitting, as they performed relatively well on the validation set. Overall, every classifier performed better than the dummy model, meaning they did better than random.

6.4. RESULTS: MIGRATION APPROACH (E2)

The approach focused on predicting negative migrations had the highest F1 score of 0.197 compared to the default migration and watchlist migration models with an F1 score of 0.151 and

Table 6.1: Evaluation metrics for each classifier

Model	F1 score	Accuracy	AUC	Precision	Recall	Correlation
DT	0.073	0.911	0.688	0.044	0.231	0.069
LDA	0.069	0.866	0.646	0.040	0.295	0.068
LG	0.098	0.936	0.646	0.063	0.228	0.093
SVM	0.077	0.927	0.648	0.049	0.203	0.070
GBM	0.096	0.935	0.737	0.063	0.230	0.092
RF	**0.197**	0.930	**0.893**	**0.119**	**0.582**	**0.240**
XGB	0.161	0.914	0.879	0.095	0.561	0.203
ANN2	0.050	**0.963**	0.673	0.042	0.064	0.033
ANN3	0.055	0.961	0.667	0.043	0.082	0.040
DUMMY	0.029	0.499	0.500	0.015	0.488	-0.003

0.120, respectively (See Table 6.2). Interestingly, default migration as the target variable resulted in a higher sensitivity than the negative migration approach, which means the model has fewer FNs. Moreover, the default migrations seem easier to predict than watchlist migrations because they score higher for every evaluation metric. This may be because the watchlist classifications are subject to expert opinion, which could lead to inconsistencies in the data.

6.5. RESULTS: TIME WINDOW (E3)

Table 6.3 provides the evaluation metrics against the different time windows. As expected, the model performance increases for larger time windows. However, the increase in performance diminishes, which is most likely caused by the reduced number of samples available for training the model. By extrapolating the results, we suspect the model could perform much better with even more historical data. Therefore, it would be interesting to investigate the optimal time window size when more data is available.

6.6. RESULTS: TARGET WINDOW (E4)

A target window of one month would mean that the model tries to predict the exact month that a negative migration would occur. Table 6.4 shows that such a model would perform poorly at predicting financial distress. Because of that, the target window was introduced because it adds flexibility to the model by also allowing earlier predictions. This is in accordance with our goal because we aim to detect financial distress as early as possible. The results show that

Table 6.2: RF model performance for different target variables

Prediction approach	F1-score	Accuracy	AUC	Precision	Sensitivity	Correlation
Negative migration	**0.197**	0.930	0.893	**0.119**	0.582	0.240
Default migration	0.151	**0.949**	**0.955**	0.083	**0.827**	**0.251**
Watchlist migration	0.120	0.911	0.897	0.067	0.646	0.187

Table 6.3: RF model performance for different time window sizes

Time window	F1 score	Accuracy	AUC	Precision	Sensitivity	Correlation
2	0.184	0.927	0.882	0.112	0.555	0.224
4	0.179	0.929	0.882	0.109	0.523	0.214
6	0.197	0.930	0.893	0.119	0.582	0.240
8	0.220	0.934	0.906	0.135	0.628	0.268
10	**0.234**	**0.935**	**0.919**	**0.142**	**0.671**	**0.288**

Table 6.4: RF model performance for different target window sizes

Target window	F1 score	Accuracy	AUC	Precision	Sensitivity	Correlation
1	0.037	**0.959**	0.694	0.020	0.197	0.051
2	0.074	0.934	0.769	0.041	0.342	0.100
4	0.156	0.927	0.857	0.092	0.514	0.194
6	**0.197**	0.930	**0.893**	**0.119**	**0.582**	**0.240**

the model performance significantly increases by extending the target window. Due to data limitations, only a target window until six months was investigated, but when more migration data is available, it would be interesting to research the optimal target window.

Besides, these model performance metrics can underestimate the functional performance of the models as a trigger. This is because these metrics reflect how well the model can predict all the negative migrations within the target window. However, in our case, the predictions already suffice when it can detect a negative migration once as early as possible. Consequently, the actual sensitivity could be much higher, which is discussed in Section 6.8.

6.7. RESULTS: TIME GAP (E5)

We hypothesised that the time gap would have a negative impact on the model performance, but this does not seem to be the case when looking at the results in Table 6.5. Although there is only a slight difference in F1 scores, there is a small upward trend in performance when the time gap is increased. This could mean that variables with a larger time lag are better at predicting financial distress. We believe this is due to the fact that the six-month time window is not optimal yet. Commonly, the PD, EAD, LGD, and RWA are only reevaluated once a year. Therefore, at least a time window of one year would be needed to incorporate patterns related to these variables into the model. Unfortunately, due to limited data, it is not possible to test this relationship. Therefore, for future research, it could be interesting to test the relationship between the time gap and model performance with a time window of at least twelve months.

6.8. WATCHLIST TRIGGER METRICS

The financial distress prediction model aims to create a new WL trigger that should raise a red flag when a client should be on the watchlist. The performance of this new proposed WL trigger can be measured using the trigger precision, migration sensitivity, and time lag metrics introduced in Section 3.2.2. Figure 6.2 plots the trigger precision against the migration sensitivity. For this, the test set of four months between June 2022 and September 2022 was used.

Most of the supervised learning models tested are not able to detect negative migrations more accurately than the ARIA triggers except for the XGBoost and RF models. These models considerably outperform the other triggers based on both trigger precision and migration sensitivity. RF has a trigger precision of 12.7% which means that of the 9474 times the trigger was raised, 1202 times a negative migration occurred within the next six months. The migration sensitivity

Table 6.5: RF model performance for different time gap sizes

Time gap	F1-score	Accuracy	AUC	Precision	Sensitivity	Correlation
1	0.197	0.930	0.893	0.119	0.582	0.240
2	0.201	0.931	0.894	0.122	0.586	0.244
3	0.212	**0.937**	0.900	0.131	0.572	0.251
4	**0.214**	0.930	**0.905**	**0.132**	**0.628**	**0.262**

was 67.6% which means that of the 581 negative migrations that occurred, 393 of these migrations had the RF trigger raised within the six months prior.

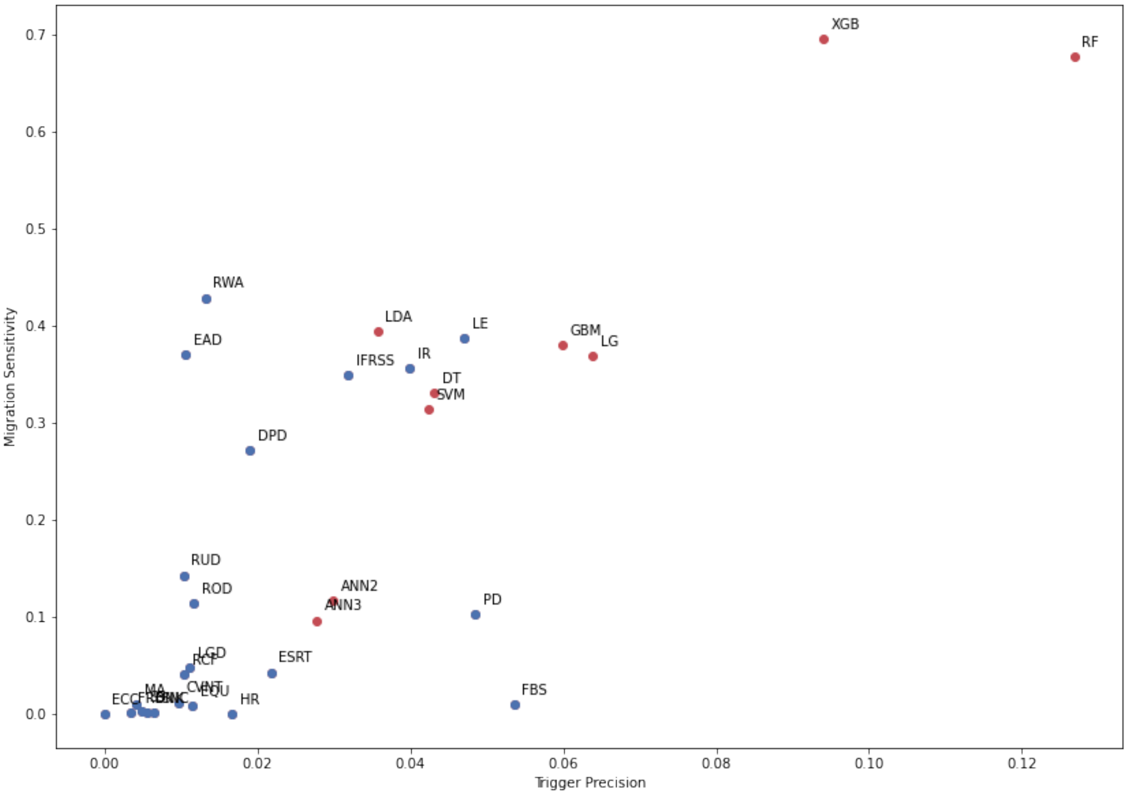


Figure 6.2: Trigger precision against migration sensitivity for the ARIA triggers (blue) and the WL triggers (red)

The trigger precision is similar to the precision calculated for evaluating ML models in the previous section (slight variation due to the randomness caused by RUS). This is due to the fact that the outcome window used for calculating the trigger precision is the same as the target window used for the financial distress prediction model. The migration sensitivity, however, is much higher than the sensitivity calculated for the ML models. When evaluating the triggers, it does not matter whether the trigger was raised for each individual six months prior to a negative migration because only one raised trigger is sufficient to detect the migration. For that reason, the sensitivity used for the ML models underestimates the detection of negative migrations compared to the migration sensitivity.

Additionally, a threshold could be set to influence migration sensitivity and trigger precision. The threshold is a decision boundary between the positive and negative classes by determining if the predicted probabilities are below or above the threshold. The model becomes more conservative when the threshold increases, meaning fewer positive instances will be predicted. Generally, this results in a higher precision but lower sensitivity.

Figure 6.3 depicts the trigger precision and migration sensitivity for different thresholds. This graph clearly shows the tradeoff between the metrics, except for a threshold higher than 0.7 because then both the trigger precision and migration sensitivity decrease. The threshold can

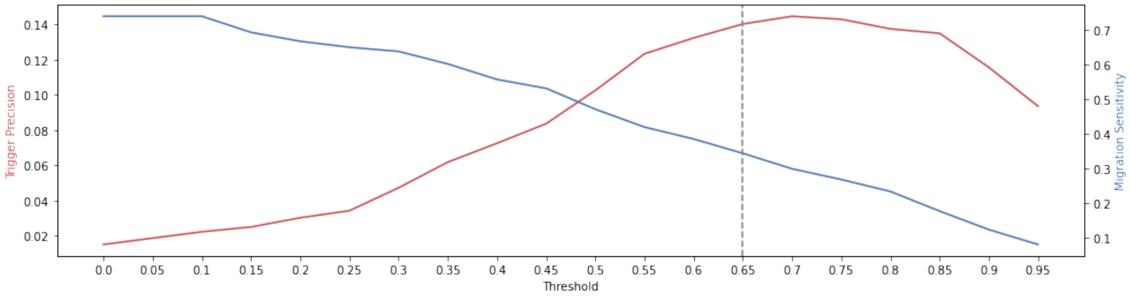


Figure 6.3: Trigger precision and migration sensitivity for different threshold values

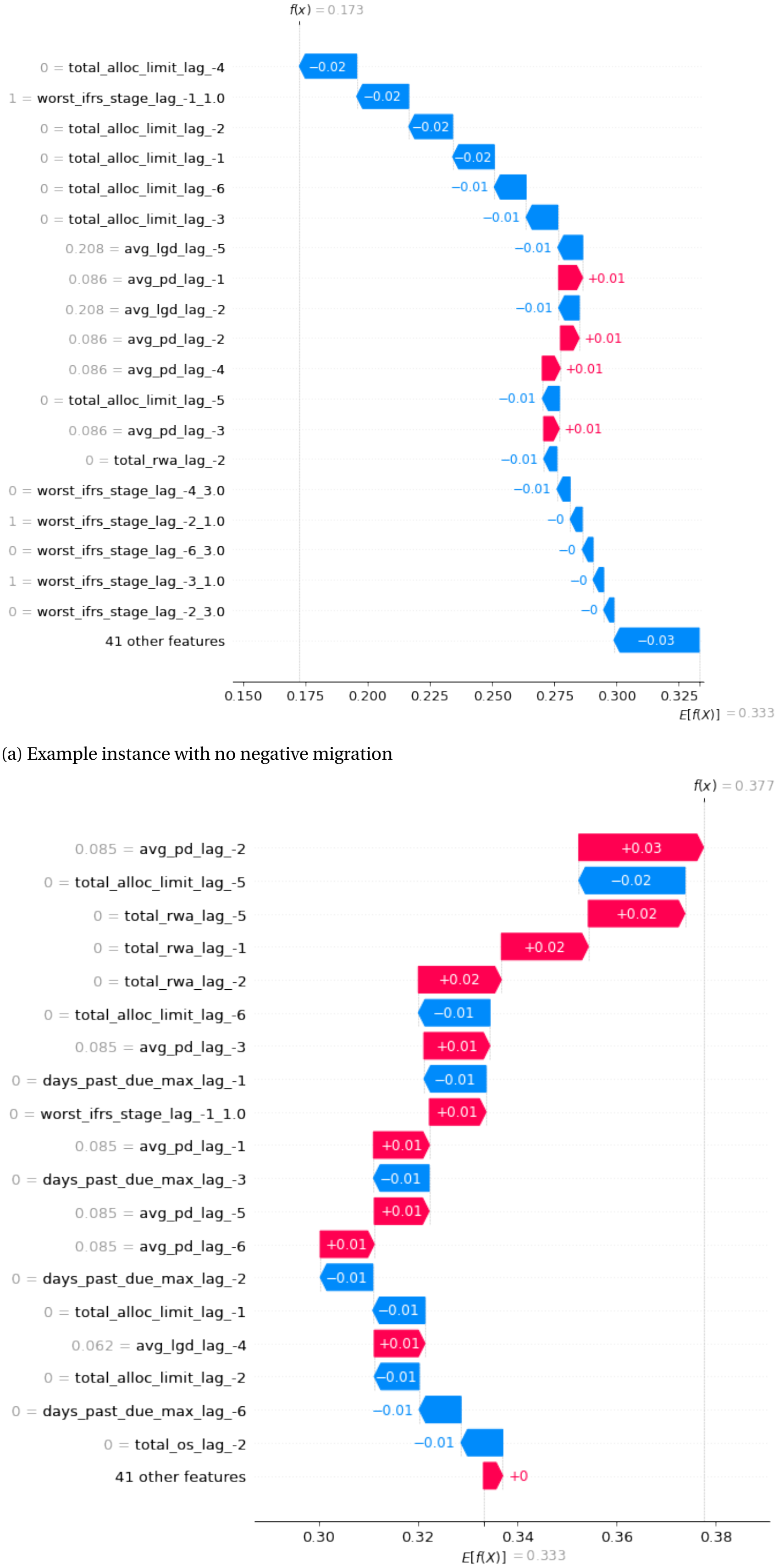
be configured depending on the costs associated with FP and FN. A FP would lead to a redundant workload for risk managers while FN would lead to clients in financial distress that are undetected.

The time lag is the final property of the WL trigger that needs to be considered. Unfortunately, due to limited historical data, the test set only consists of four months, which is insufficient data to measure the average time lag of the proposed triggers reliably. In section 3.2.2, at least a period of twelve months was used to see how early or late the triggers were at detecting negative migrations. Therefore, when more historical data is available, more research could be done on the earliness of the triggers. The time lag of the triggers can be altered by changing the time gap and target window of the models. The desired balance between time lag and model accuracy could be set by tuning these parameters.

6.9. SHAP VALUES

Figure 6.4 provides the SHAP values of a sample with and without a predicted negative migration in a waterfall chart. These graphs give insight into how the individual predictions came about by showing which features had a negative or positive contribution to the final prediction. The first example shows that most features had a negative contribution on the watchlist classification of which the total allocated limit and the IFRS stage had the biggest overall impact. As a result, the model predicted that this client would not migrate to the watchlist. In contrast, the second example demonstrates distinct features that positively contribute to a watchlist classification including the average PD and the total RWA. Consequently, the model predicts that this client will receive a watchlist status.

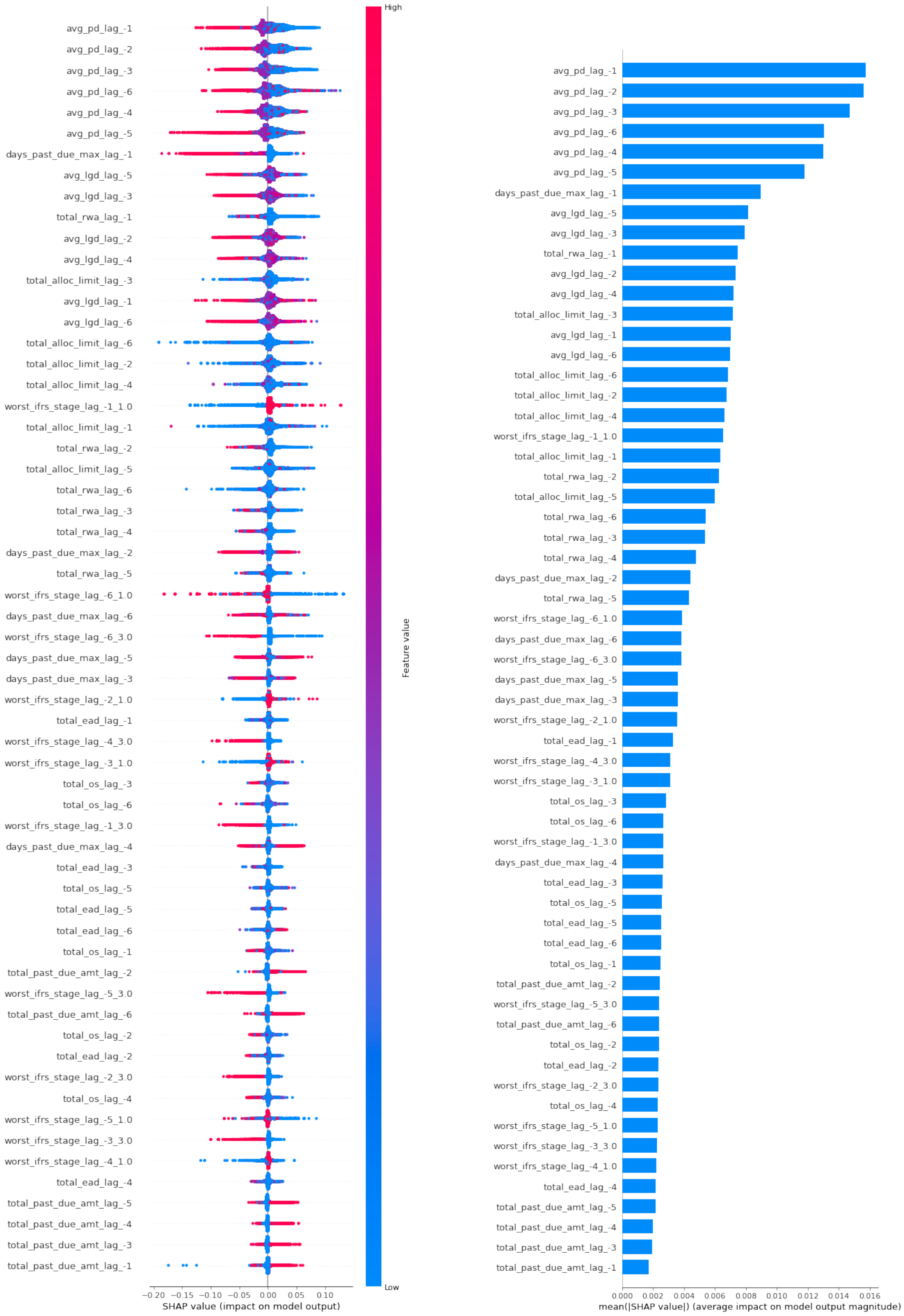
Next to analysing the individual predictions, the aggregated impact of the features can be analysed by looking at the average absolute SHAP value shown in Figure 6.5. This value provides insight into the overall impact or importance of each feature in influencing the model's predictions. The graph shows that the lag features related to PD, DPD, and LGD have the highest impact on the output of the model. Such insights provide valuable insights into the relative importance of these lag features in the model's decision-making process while taking into account the different interaction effects between the independent variables.



(a) Example instance with no negative migration

(b) Example instance with negative migration

Figure 6.4: SHAP waterfall plots



(a) SHAP beeswarm plot

(b) Average SHAP bar plot

Figure 6.5: SHAP value analysis

7 Conclusion

7.1. Conclusions

In conclusion, this research investigates how a WL trigger can be designed to effectively classify WB clients on a watchlist based on their prospective credit risk. The main research question is answered based on the following subquestions:

How do the early warning triggers relate to the watchlist status of clients?

Firstly, the relationship between the ARIA triggers was researched by defining their desired properties and how these could be measured. The desired properties include the following: a WL trigger should be able to detect as many clients in distress as possible, a raised WL trigger should indicate that there is a high probability that a client will have a negative migration, and the trigger should be able to detect financial distress as early as possible. These properties can be measured by migration sensitivity, trigger precision, and time lag, respectively.

How can a financial distress prediction model create for the watchlist classification of clients?

Secondly, a financial distress prediction model incorporating historical triggers and internal customer data has been developed to predict if a negative migration occurs in the next month. For this, the following supervised learning models were tested: LDA, LR, DT, SVM, RF, GBM, XGBoost, and ANN. According to the experiments, the RF performed best with an F1 score of 0.197. Besides, experimentation showed that increasing the time window and target window significantly improves the model's predictive performance. Also, the timeliness of the models can be extended by increasing the time lag, which interestingly did not significantly decrease the performance of the models. Furthermore, the SHAP values have proven to be an insightful way of explaining the decision-making of the models by showing how each feature contributes to making the predictions. Also, the mutual information and SHAP values show that the internal customer data has the highest impact on the performance while the impact of the ARIA triggers is relatively small. Conclusively, the predictive performance is better when using the underlying data rather than the ARIA triggers alone as input for the models.

How can we implement a new WL trigger?

The new WL trigger can be derived from the predictions of the introduced financial distress prediction model, and the effectiveness of the WL trigger can be compared with the ARIA triggers using the aforementioned trigger metrics. Both RF and XGBoost were able to significantly outperform the ARIA triggers at detecting financial distress. Unfortunately, due to limited historical data, we can not make any conclusions about the timeliness of the triggers.

This study contributes to the current body of research by bridging the gap between theory and practice. The literature lacks research on how EWS and financial distress prediction models are implemented at financial institutions. This study provides a comprehensive case study at ING on the use of early warning triggers and their relationship with the watchlist and default status of clients. As a result, several statistical analyses are introduced that could be used to measure the effectiveness of individual triggers in detecting clients in financial distress. Besides, the proposed financial distress prediction model differs from the models presented in the literature as it incorporates different features, such as early warning triggers and internal customer data, while other studies mainly focus on publicly available data. In addition, we extend the definition of financial distress, which focuses both on clients with a default or a watchlist status. The literature does not provide a financial distress prediction model that tries to predict watchlist classifications. Also, the concept of a target window to take into account the earliness property of the distress prediction model has not been used in the found literature related to the financial distress prediction model for early warning detection. Finally, the research provides new insights into how a financial distress prediction model could be implemented as a WL trigger and what properties such a trigger should have.

Moreover, the practical contribution relates to increasing the predictive performance of clients in financial distress. This research shows that many clients in default did not transfer to the watchlist first. By introducing a new WL trigger, the information collected by ARIA can be aggregated to improve the earliness and predictive performance of the whole system. As a result, the introduced WL trigger makes it possible to make inferences about the future, which risk managers can use to monitor their credit portfolios and support their decision-making. Moreover, this research provides insight into the effectiveness of early warning triggers by introducing three metrics. These insights could be used to design new triggers in the future, and they can help communicate the effectiveness of triggers to stakeholders.

7.2. FUTURE RESEARCH AND RECOMMENDATIONS

Finally, limitations and possible topics for future research are discussed. Firstly, this thesis examines three metrics to measure the effectiveness of early warning triggers. However, due to limited time, it was not possible to research how vital migration sensitivity, trigger precision, and time lag are compared to each other. It would be interesting for future research to estimate the costs linked with these properties or seek expert advice to attain the best balance between these metrics.

In addition, the analysis of the model performance is only limited to calculating the estimates of the metrics but it does not consider the uncertainty of these measurements. For future research, it could be interesting to construct confidence intervals using the bootstrap sampling technique to measure this uncertainty so that statistically significant conclusions can be made.

Besides, the proposed model is limited to only predicting negative migrations. For future research, it could be interesting to extend the model to detect other types of migrations, such as positive migrations or migrations between watchlist and default.

Furthermore, more research could be done to improve the financial distress prediction models. The experimentation part shows that extending the time window significantly increases the predictive performance of the models. Therefore, when more historical data is available, more research could be done on finding the optimal time window. Also, more data sources could be added as input for the models. For instance, external data like macroeconomic and market variables or internal data related to financial accounting data of clients could be incorporated.

Finally, the proposed model could be improved by trying other ML techniques and doing more extensive model tuning. For example, different techniques to deal with class imbalance and feature selection could be tested. Additionally, the predictive performance of other ML models such as LightGBM, CatBoost, K-Nearest Neighbour, Naive Bayes, Graphical Models, or Hidden Markov Models (HMM) could be researched. Also, early time series classification could be an attractive model to explore as it can incorporate the tradeoff between predictive performance and timeliness into the cost function.

Appendix H and Appendix I provide more detailed recommendations for future research and deploying the WL trigger.

References

[1] B. Babel, G. Kaltenbrunner, S. Kinnebrock, L. Pancaldi., K. Richter, S. Schneider, First-mover matters: building credit monitoring for competitive advantage, Technical Report, 2012. URL: https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/first-mover-matters-building-credit-monitoring-for-competitive-advantage.

[2] ECB, Guidance to banks on non-performing loans, Technical Report, 2017. URL: https://www.bankingsupervision.europa.eu/banking/priorities/npl/html/guidanceonnpls.en.html.

[3] S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia, D. C. Anastasiu, in: 2019 IEEE fifth international conference on big data computing service and applications (BigDataService), IEEE, pp. 205–208.

[4] C. Wen, J. Yang, L. Gan, Y. Pan. Big data driven internet of things for credit evaluation and early warning in finance. Future Generation Computer Systems 124 (2021) 295–307.

[5] Y. Liang, D. Quan, F. Wang, X. Jia, M. Li, T. Li. Financial big data analysis and early warning platform: a case study. IEEE Access 8 (2020) 36515–36526.

[6] B. Fang, P. Zhang, in: Big data concepts, theories, and applications, Springer, 2016, pp. 391–412.

[7] A. R., G. R., D. S., M. M., R. N., H. R., Banks Must Act on their Early Warning Systems or Risk ROE Downturn, Technical Report, 2022. URL: https://www.pwc.co.uk/banking-capital-markets/assets/documents/future-of-ews.pdf.

[8] ING, Ing profile, https://www.ing.com/Newsroom/Media-kit/Profile-fast-facts/Profile.htm, 2023.

[9] E. B. Authority. Guidelines on loan origination and monitoring (2019).

[10] BIS, Principles for the Management of Credit Risk, Technical Report, 1999. URL: https://www.bis.org/publ/bcbs54.htm.

[11] S. Shi, R. Tse, W. Luo, S. D’Addona, G. Pau. Machine learning-driven credit risk: a systemic review. Neural Computing and Applications (2022) 1–13.

[12] M. Bazarbash, Fintech in financial inclusion: machine learning applications in assessing credit risk, International Monetary Fund, 2019.

[13] S. Aziz, M. Dowling, in: Disrupting finance, Palgrave Pivot, Cham, 2019, pp. 33–50.

[14] S. Bhatore, L. Mohan, Y. R. Reddy. Machine learning techniques for credit risk evaluation: a systematic literature review. Journal of Banking and Financial Technology 4 (2020) 111–138.

[15] W. Hilal, S. A. Gadsden, J. Yawney. Financial fraud:: A review of anomaly detection techniques and recent advances (2022).

[16] A. Markov, Z. Seleznyova, V. Lapshin. Credit scoring methods: Latest trends and points to consider. The Journal of Finance and Data Science (2022).

[17] M. R. Kumar, V. K. Gunjan. Review of machine learning models for credit scoring analysis. Ingeniería Solidaria 16 (2020).

[18] I. Kaluđer, G. Klepac, in: Central European Conference on Information and Intelligent Systems, Faculty of Organization and Informatics Varazdin, p. 250.

[19] A. S. Koyuncugil, N. Ozgulbas. Financial early warning system model and data mining application for risk detection. Expert systems with Applications 39 (2012) 6238–6253.

[20] L. Alessi, A. Antunes, J. Babeckỳ, S. Baltussen, M. Behn, D. Bonfim, O. Bush, C. Detken, J. Frost, R. Guimaraes, et al. Comparing different early warning systems: Results from a horse race competition among members of the macro-prudential research network (2015).

[21] M. Pompella, A. Dicanio. Ratings based inference and credit risk: Detecting likely-to-fail banks with the pc-mahalanobis method. Economic modelling 67 (2017) 34–44.

[22] A. Petropoulos, V. Siakoulis, E. Stavroulakis. Towards an early warning system for sovereign defaults leveraging on machine learning methodologies. Intelligent Systems in Accounting, Finance and Management 29 (2022) 118–129.

[23] W.-Y. Lin, Y.-H. Hu, C.-F. Tsai. Machine learning in financial crisis prediction: a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42 (2011) 421–436.

[24] S. Ashraf, E. GS Félix, Z. Serrasqueiro. Do traditional financial distress prediction models predict the early warning signs of financial distress? Journal of Risk and Financial Management 12 (2019) 55.

[25] W. Wang. Early warning and monitoring analysis of financial accounting indicators of listed companies based on big data. Scientific Programming 2022 (2022).

[26] X. Chen, X. Wang, D. D. Wu. Credit risk measurement and early warning of smes: An empirical study of listed smes in china. Decision support systems 49 (2010) 301–310.

[27] L. Tong, G. Tong. A novel financial risk early warning strategy based on decision tree algorithm. Scientific Programming 2022 (2022).

[28] S. A. Balasubramanian, G. Radhakrishna, P. Sridevi, T. Natarajan. Modeling corporate financial distress using financial and non-financial variables: The case of indian listed companies. International Journal of Law and Management 61 (2019) 457–484.

[29] B. Huang, X. Yao, Y. Luo, J. Li. Improving financial distress prediction using textual sentiment of annual reports. Annals of Operations Research (2022) 1–28.

[30] X.-F. Hui, J. Sun, in: International Conference on Modeling Decisions for Artificial Intelligence, Springer, pp. 274–282.

[31] K. Xu, Q. Zhao, X. Bao. Study on early warning of enterprise financial distress—based on partial least-squares logistic regression. Acta Oeconomica 65 (2015) 3–16.

[32] W. Liu, H. Fan, M. Xia, C. Pang. Predicting and interpreting financial distress using a weighted boosted tree-based tree. Engineering Applications of Artificial Intelligence 116 (2022) 105466.

[33] D. Wu, X. Ma, D. L. Olson. Financial distress prediction using integrated z-score and multilayer perceptron neural networks. Decision Support Systems (2022) 113814.

[34] J. Zhong, Z. Wang. Artificial intelligence techniques for financial distress prediction. AIMS Mathematics 7 (2022) 20891–20908.

[35] X. Hu. Design and application of a financial distress early warning model based on data reasoning and pattern recognition. Advances in Multimedia 2022 (2022).

[36] Y. Zou, C. Gao, H. Gao. Business failure prediction based on a cost-sensitive extreme gradient boosting machine. IEEE Access 10 (2022) 42623–42639.

[37] M. M. Ma'aji, N. A. H. Abdullah, K. L.-H. Khaw. Financial distress among smes in malaysia: An early warning signal. International Journal of Business & Society 20 (2019).

[38] Y. Cao. Aggregating multiple classification results using choquet integral for financial distress early warning. Expert Systems with Applications 39 (2012) 1830–1836.

[39] J. Sun, H. Li. Financial distress early warning based on group decision making. Computers & Operations Research 36 (2009) 885–906.

[40] Financial times, https://www.ft.com/, 2023.

[41] Google news, https://news.google.com/, 2023.

[42] Refinitiv, https://www.refinitiv.com/, 2023.

[43] C. M. Bishop, N. M. Nasrabadi, Pattern recognition and machine learning, volume 4, Springer, 2006.

[44] P. Xanthopoulos, P. M. Pardalos, T. B. Trafalis, P. Xanthopoulos, P. M. Pardalos, T. B. Trafalis. Linear discriminant analysis. Robust data mining (2013) 27–33.

[45] E. I. Altman, in: Handbook of research methods and applications in empirical finance, Edward Elgar Publishing, 2013.

[46] J. S. Cramer. The origins of logistic regression (2002).

[47] B. E. Boser, I. M. Guyon, V. N. Vapnik, in: Proceedings of the fifth annual workshop on Computational learning theory, pp. 144–152.

[48] T. Thomas, A. P. Vijayaraghavan, S. Emmanuel, Machine learning approaches in cyber security analytics, Springer, 2020.

[49] J. Mingers. An empirical comparison of pruning methods for decision tree induction. Machine learning 4 (1989) 227–243.

[50] Q. Zheng, J. Yanhui, in: 2007 IEEE International Conference on Service Operations and Logistics, and Informatics, IEEE, pp. 1–6.

[51] D. Opitz, R. Maclin. Popular ensemble methods: An empirical study. Journal of artificial intelligence research 11 (1999) 169–198.

[52] L. Breiman. Bagging predictors. Machine learning 24 (1996) 123–140.

[53] G. Biau, E. Scornet. A random forest guided tour. Test 25 (2016) 197–227.

[54] R. E. Schapire. The boosting approach to machine learning: An overview. Nonlinear estimation and classification (2003) 149–171.

[55] D. Svozil, V. Kvasnicka, J. Pospichal. Introduction to multi-layer feed-forward neural networks. Chemometrics and intelligent laboratory systems 39 (1997) 43–62.

[56] B. P. V. Ninh, T. Do Thanh, D. V. Hong. Financial distress and bankruptcy prediction: An appropriate model for listed firms in vietnam. Economic Systems 42 (2018) 616–624.

[57] M. Sheikhi, M. F. Shams, Z. Sheikhi. Financial distress prediction using distress score as a predictor. International Journal of Business and Management 7 (2012) 169.

[58] T. Fawcett. An introduction to roc analysis. Pattern recognition letters 27 (2006) 861–874.

[59] M. Sreedharan, A. M. Khedr, M. El Bannany. A multi-layer perceptron approach to financial distress prediction with genetic algorithm. Automatic Control and Computer Sciences 54 (2020) 475–482.

[60] B. Ribeiro, C. Silva, A. Vieira, A. Gaspar-Cunha, J. C. das Neves, in: The 2010 International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 1–7.

[61] P. Hájek, V. Olej, in: Engineering Applications of Neural Networks: 14th International Conference, EANN 2013, Halkidiki, Greece, September 13-16, 2013 Proceedings, Part II 14, Springer, pp. 1–10.

[62] M. Pavlicko, M. Durica, J. Mazanec. Ensemble model of the financial distress prediction in visegrad group countries. Mathematics 9 (2021) 1886.

[63] D. Chicco, G. Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC genomics 21 (2020) 1–13.

[64] P. J. Phillips, C. A. Hahn, P. C. Fontana, D. A. Broniatowski, M. A. Przybocki. Four principles of explainable artificial intelligence. Gaithersburg, Maryland (2020) 18.

[65] L. S. Shapley, Notes on the N-person Game, Rand Corporation, 1951.

[66] S. M. Lundberg, S.-I. Lee. A unified approach to interpreting model predictions. Advances in neural information processing systems 30 (2017).

[67] H. Kim, H. Cho, D. Ryu. Predicting corporate defaults using machine learning with geometric-lag variables. Investment Analysts Journal 50 (2021) 161–175.

[68] D. Yan, G. Chi, K. K. Lai. Financial distress prediction and feature selection in multiple periods by lassoing unconstrained distributed lag non-linear models. Mathematics 8 (2020) 1275.

[69] V. Klepáč, D. Hampel, et al. Prediction of bankruptcy with svm classifiers among retail business companies in eu. Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis 64 (2016) 627–634.

[70] A. Kraskov, H. Stögbauer, P. Grassberger. Estimating mutual information. Physical review E 69 (2004) 066138.

[71] B. C. Ross. Mutual information between discrete and continuous data sets. PloS one 9 (2014) e87357.

[72] J. Bergstra, Y. Bengio. Random search for hyper-parameter optimization. Journal of machine learning research 13 (2012).

[73] S. Weiran, Hyper Parameter Tuning with Randomised Grid Search - Towards Data Science, 2021. URL: https://towardsdatascience.com/hyper-parameter-tuning-with-randomised-grid-search-54f865d27926.

[74] C. Bergmeir, J. M. Benítez. On the use of cross-validation for time series predictor evaluation. Information Sciences 191 (2012) 192–213.

[75] M. Schnaubelt, A comparison of machine learning model validation schemes for non-stationary time series data, Technical Report, FAU Discussion Papers in Economics, 2019.

[76] R. Kohavi, et al., in: Ijcai, volume 14, Montreal, Canada, pp. 1137–1145.

[77] A. Gramegna, P. Giudici. Shap and lime: an evaluation of discriminative power in credit risk. Frontiers in Artificial Intelligence 4 (2021) 752558.

[78] B. H. Misheva, J. Osterrieder, A. Hirsa, O. Kulkarni, S. F. Lin. Explainable ai in credit risk management. arXiv preprint arXiv:2103.00949 (2021).

[79] W. Wang, C. Lesner, A. Ran, M. Rukonic, J. Xue, E. Shiu, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pp. 13396–13401.

[80] Z. Zhang, C. Wu, S. Qu, X. Chen. An explainable artificial intelligence approach for financial distress prediction. Information Processing & Management 59 (2022) 102988.

[81] W. Yi. Z-score model on financial crisis early-warning of listed real estate companies in china: a financial engineering perspective. Systems Engineering Procedia 3 (2012) 153–157.

[82] M. M. Ma'aji, N. A. H. Abdullah, K. L.-H. Khaw. Predicting financial distress among smes in malaysia. European Scientific Journal, ESJ 14 (2018) 91–102.

[83] C. Goutte, E. Gaussier, in: Advances in Information Retrieval: 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005. Proceedings 27, Springer, pp. 345–359.

[84] F. M. Dekking, C. Kraaikamp, H. P. Lopuhaä, L. E. Meester, A Modern Introduction to Probability and Statistics: Understanding why and how, volume 488, Springer, 2005.

[85] S. Bates, T. Hastie, R. Tibshirani. Cross-validation: what does it estimate and how well does it do it? arXiv preprint arXiv:2104.00673 (2021).

[86] R. W. Johnson. An introduction to the bootstrap. Teaching statistics 23 (2001) 49–54.

[87] M. Hanif, F. Sami, M. Hyder, M. I. Ch. Hidden markov model for time series prediction. Journal of Asian Scientific Research 7 (2017) 196–205.

[88] O. Blümke. A structural hidden markov model for forecasting scenario probabilities for portfolio loan loss provisions. Knowledge-Based Systems 249 (2022) 108934.

[89] W.-K. Ching, H.-Y. Leung, Z. Wu, H. Jiang. Modeling default risk via a hidden markov model of multiple sequences. Frontiers of Computer Science in China 4 (2010) 187–195.

[90] H. Kamath, N. F. Jahan. Using hidden markov model to monitor possible loan defaults in banks (2020).

[91] U. Mori, A. Mendiburu, I. M. Miranda, J. A. Lozano. Early classification of time series using multi-objective optimization techniques. Information Sciences 492 (2019) 204–218.

[92] A. Gupta, H. P. Gupta, B. Biswas, T. Dutta. Approaches and applications of early classification of time series: A review. IEEE Transactions on Artificial Intelligence 1 (2020) 47–61.

[93] Z. Xing, J. Pei, P. S. Yu. Early classification on time series. Knowledge and information systems 31 (2012) 105–127.

[94] K. Jaskie, C. Elkan, A. Spanias, in: 2019 53rd Asilomar Conference on Signals, Systems, and Computers, IEEE, pp. 2007–2011.

[95] M. Treveil, N. Omont, C. Stenac, K. Lefevre, D. Phan, J. Zentici, A. Lavoillotte, M. Miyazaki, L. Heidmann, Introducing MLOps, O'Reilly Media, 2020.

[96] Google, MLOps: Continuous delivery and automation pipelines in machine learning, https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning, 2023.

A

LITERATURE REVIEW

A.1. LITERATURE REVIEW SUMMARY

Table A.1: Literature review: early warning monitoring of financial distress

Study	Model	Independent variables	Dependent variable	Application in EWS	Client type
[27]	DT	Financial ratios	Cash flow below a critical value	Establishes a EWS by implementing a financial distress prediction model	Listed companies
[25]	SVM	Financial ratios, and online public opinion text	ST company	Does not mention EWS but discusses the use of big data for early warning monitoring and develops a financial distress prediction model that puts forward early warning signals	Listed companies in China
[19]	DT	Financial ratios	Financial performance	Provides literature review on financial EWS and develops a data flow framework of an EWS that includes a financial distress prediction model to determine risk profiles.	SMEs listed in Turkey
[29]	LR, ANN, SVM, RF, and XGBoost	Financial ratios, textual sentiment of annual reports, market variables (firm's equity compared to total market value, excess return, stock volatility)	ST company	Mentions the role of financial distress prediction models in EWS and established an EWS based on a financial distress prediction model.	Listed companies in China

[30]	SVM, LR, MDA, and ANN	Financial ratios	ST company	Does not mention EWS but develops a financial distress prediction model for early warning signalling	Listed SMEs in China
[31]	Partial least-squares LR	Financial ratios	ST company	Establishes a EWS by implementing a financial distress prediction model	Listed companies in China
[32]	Weighted XG-Boost	Financial ratios	ST company	Does not mention EWS but develops a financial distress prediction model for early warning signalling	Listed companies in China
[26]	KMV model	Financial ratios	ST company	Does not mention EWS but develops a financial distress prediction model for early warning signalling	Listed SMEs in China
[33]	Hybrid Z-score and ANN model	Financial ratios	ST company	Does not mention EWS but develops a financial distress prediction model for early warning signalling	Listed companies
[34]	LR, DT, ANN, RF, SVM, XGBoost, and CatBoost	Financial ratios	ST company	Establishes a EWS by implementing a financial distress prediction model	Listed companies

[35]	ANN	Financial ratios	ST company	Does not mention EWS but develops a financial distress prediction model for early warning signalling	Listed companies in China

[24]	Z-score, O-score, Hazard model, LR, and D-score	Financial ratios and market variables related to volatility and returns	Multiple criteria: • Less than 50% quotation of book value for consecutive 3 years • Failure of dividend/bonus declaration from continuous 5 years • Failed to conduct AGM for consecutive 3 years • Failed to pay the yearly listing fee for 2 years. • Delisted/ Suspended/ Liquidation/ Winding up/ Bankruptcy	Does not mention EWS but tests several traditional financial distress prediction models for early warning signalling by extending the definition of financial distress by including companies that show early signs of distress	Listed companies in Pakistan

[4]	RF	Financial ratios and other indicators like credit ratings and relationship strength and duration	Financial credit risk based on expert judgement	Designs a data architecture for an early warning model in credit risk to extract and pre-process big data. Analyse credit risk based on a qualitative and quantitative approach	Listed SMEs in the Internet of Things financial sector
[81]	Z-score	Financial ratios	ST company	Does not mention EWS but but tests the suitability of the Z-score model for financial early warning	Listed real estate companies in China
[36]	XGBoost	Financial ratios	ST company	Establishes a EWS by implementing a financial distress prediction model	Listed companies in China

[82]	MDA, and LR	Financial ratios, governance indicators regarding characteristics of shareholders and board members, and non-financial indicators related to the age of the company	Failed SMEs based on the following criteria: • Annual sales turnover which did not exceed RM25 million following the National SME Development Council's definition of SMEs and • The companies were classified under the winding up by Court Order or by creditors' request	Does not mention EWS but develops a financial distress prediction model for early warning signalling	Listed SMEs in Malaysia
[28]	LR	Financial ratios, and non-financial indicators (age, promoters holdings pledged, and institutional holdings)	Companies that, at the end of any financial year, have accumulated losses equal to or exceeding their entire net worth.	Does not mention EWS but develops a financial distress prediction model for early warning signalling	Listed companies in India

[39]	Group decision-making framework (MCDA)	Qualitative attributes: • Investment risk • Market information • Management and control • Consciousness of debt risk • Corporate governance • Financial ability	ST company	Establishes a EWS by implementing a financial distress prediction model based on a group-decision making approach	Case study of a single listed Chinese company
[38]	Combination model	Financial ratios	ST company	Does not mention EWS but develops a financial distress prediction model for early warning signalling	Listed companies in China

[18]	Fuzzy Logic	Early warning triggers data sources: • Internal data • Group data • Financial statements • Macroeconomic and industry analysis • Credit Bureau • Capital markets • Government databases • Media • Payment transactions	Domain expert knowledge	Establishes an EWS by developing a methodology that uses domain expert knowledge to provide early warning signals. It also mentions early triggers and introduces performance indicators to measure their effectiveness.	Clients of a financial institution
This research	LDA, LR, SVM, RF, GBM, XGBoost, and ANN	Early warning triggers and internal client data (See Section 3.1.1)	Client status migrations (See Section 3.1.2)	Provides a case study of EWS at ING by analysing the relationship between early warning triggers and financial distress, and develops a financial distress prediction model based on ML models and early warning triggers for watchlist classification of WB clients	WB clients at ING

B

Client Status Time Series

As discussed in Section 3.2.1, when assuming the default status of clients in December 2022 to be the same as the month before, then the time series of the defaults becomes smooth (Figure B.1).

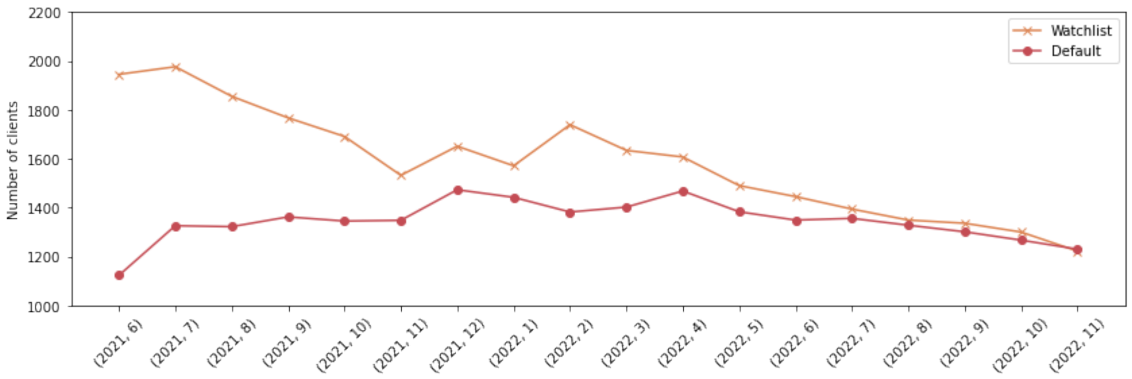


Figure B.1: Watchlist and default time series without 2021-12 outlier

C

Early Warning Trigger Metrics

C.1. Literature

Table C.1 provides an overview of the early warning trigger metrics found in the literature.

Table C.1: Early Warning Triggers in the Literature

Study	Early Warning Trigger Metrics
[1]	**Hit ratio**: proportion of flagged customers that are transferred to the watchlist **Direct transfers**: number of nonperforming clients that were not on the watchlist before **Selectivity**: ratio of flagged nonperforming clients compared to flagged performing clients **Regression**: statistical significance in a univariate or multivariate context **Time**: average time before default when a trigger identifies a nonperforming customer for the first time
[18]	**Weight of evidence**: measures the statistical relationship between binary target and independent variables **Information value**: predictive power of a set of independent variables on a binary target variable **Time to default**: average time between trigger and default **Workload**: number of clients for which a trigger has been raised
This research	**Migration sensitivity**: the fraction of negative migrations which had a certain trigger raised six months before. **Trigger pecision**: the fraction of raised triggers for which there were negative migrations in the next six months. **Time lag**: the number of months between a trigger event and a negative migration

C.2. CALCULATION METHOD

In this appendix, two approaches are discussed for which we could calculate the sensitivity and precision of the early warning triggers. Firstly, studies in the literature calculate their early warning metrics by defining an outcome window in which it is determined if a client went into default. Then, the period (trigger window) before the outcome window is used to determine if a given trigger occurred. Figure C.1a gives an example where both the trigger and outcome window is six months. A confusion matrix can be constructed by determining the frequency of clients who had a negative migration and whether they had a specific trigger raised (See Table C.2).

Table C.2: Early warning trigger confusion matrix

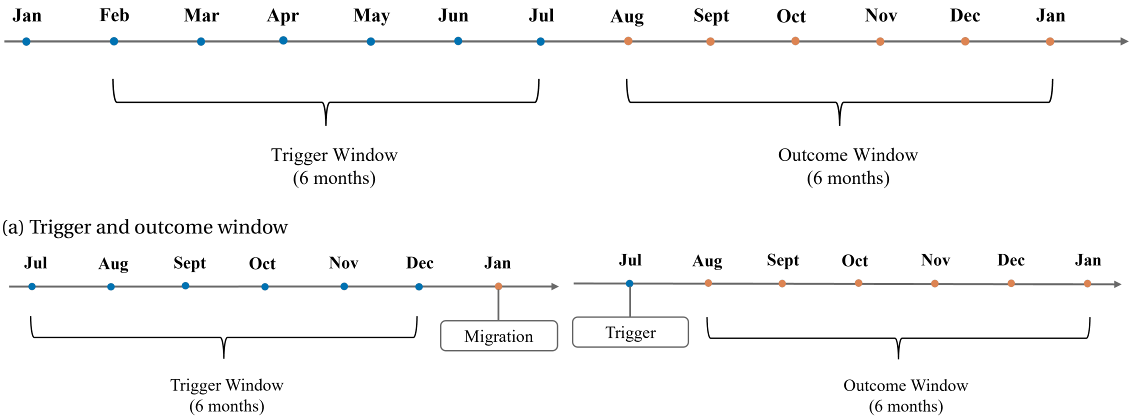
	Negative migration	**No negative migration**
Trigger raised	True Positive (TP)	False Negative (FN)
Trigger not raised	False Positive (FP)	True Negative (TN)

Based on this confusion matrix, regular evaluation metrics can be calculated to measure the performance of ML models. In our case, the sensitivity and precision are calculated as follows, and the results of these calculations can be found in the next section:

- Sensitivity: $\hat{p}_s = \dfrac{TP}{TP + FN}$
- Precision: $\hat{p}_1 = \dfrac{TP}{TP + FP}$

The second approach calculates the metrics by using the moment a trigger or a migration occurs as a reference point. In case of sensitivity, we can select all the instances where a migration occurred. Then, we look six months in advance to see whether a trigger occurred before the migration (Figure C.1b). By dividing the number of times a trigger occurred before a migration by the total number of migrations, we can determine how many migrations a particular trigger could identify. In the case of precision, we select all the raised triggers and then determine if migration occurred within the six months after the trigger (Figure C.1c). Then, we divide the number of times a migration occurred by the total number of times the trigger was raised. As a result, we can determine how likely migration is to happen in the next six months when we observe a trigger. The metrics calculated by this approach are found in Section 3.2.2.

The second approach is used to compare the performance between the ARIA triggers and the proposed WL trigger because it can be interpreted more easily, and it requires a smaller period to calculate the metrics.



(a) Trigger and outcome window

(b) Trigger window before a negative migration

(c) Outcome window after a raised trigger

Figure C.1: Calculation methods

C.3. RESULTS: TRIGGER AND TARGET WINDOW APPROACH

C.3.1. SENSITIVITY

Figure C.2 shows the sensitivity of the triggers based on the first approach using the trigger and outcome window. A few triggers have a higher sensitivity compared with the second approach (See Figure 3.5). In this case, the RWA, EAD, and IFRSS triggers occur frequently among the watchlisted clients. This is probably due to the outcome window, which makes it more likely that a combination of a specific trigger with a negative migration occurs. But overall, there are no significant differences in the ranking when considering the confidence intervals.

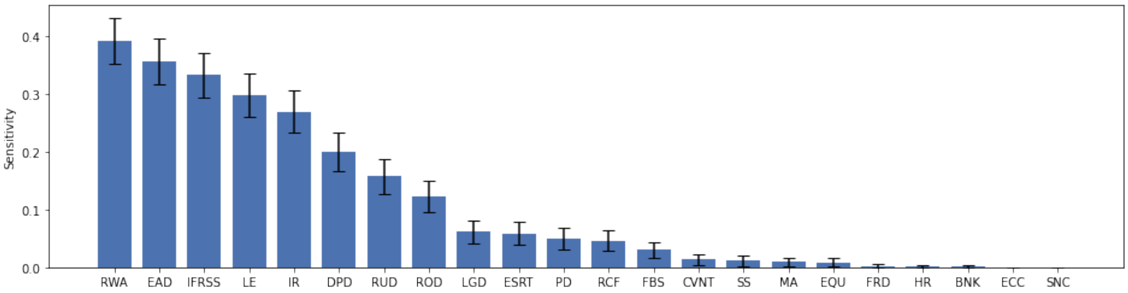


Figure C.2: Sensitivity with 99% confidence interval

C.3.2. PRECISION

Figure C.3 shows the precision for each trigger based on the first approach. The precision differs significantly from the precision calculated in Figure 3.6. In this case, the FBS has the highest precision instead of the PD. This has probably to do with the timeliness of the triggers and the different time horizons used for the approaches. PD has a relatively small average time lag while the average time lag of FBS is rather large (See Figure C.1. Consequently, PD might be more likely to occur a few months before a negative migration, while FBS is better at detecting a negative migration in the next half year. Therefore, the first approach could be used to measure short-term predictability, while the second approach is better at measuring the long-term predictability of the triggers.

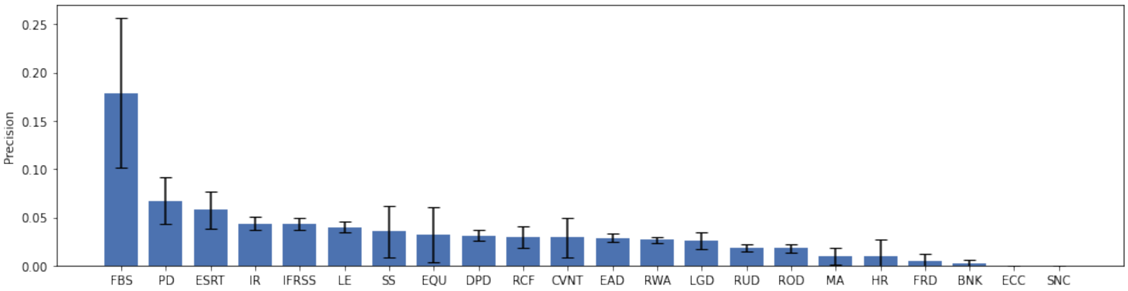


Figure C.3: Precision with 99% confidence interval

C.3.3. STATISTICAL SIGNIFICANCE

Besides, the precision can be analysed based on its difference with the fraction of clients that did not have the particular trigger raised but still ended up on the watchlist. A fraction of clients get on the watchlist no matter what trigger has been raised before. Therefore by looking at the difference, we can determine to what extent the trigger is able to separate both classes. Figure C.4 shows this difference with the corresponding 99% confidence intervals. If the confidence interval does not overlap with the zero line, we can conclude that the trigger can significantly separate financially distressed clients from regular clients. This is the case for most triggers except for the EQU, FBS, CVNT, LGD, RCF, SNC and HR.

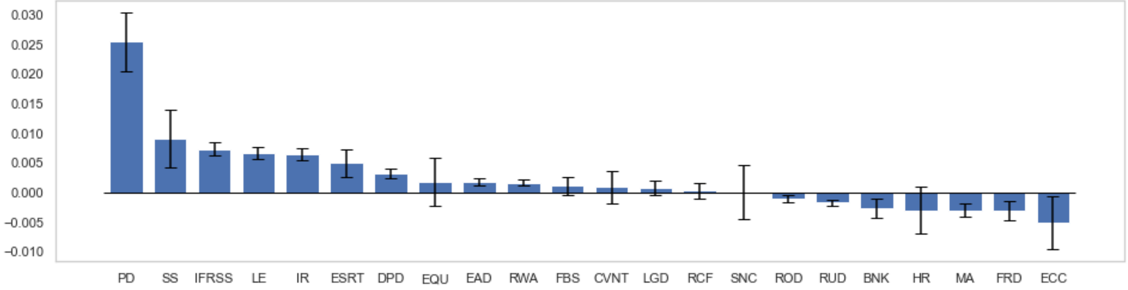


Figure C.4: Two sample proportion difference with 99% confidence interval

C.3.4. CORRELATION

The Pearson correlation measures the linear dependence between two variables. If the two variables are binary, then the correlation equals the Matthews Correlation Coefficient (MCC), which is a metric that could be used to evaluate the performance of a binary classification [63]. Therefore, MCC indicates how well an individual trigger would perform on the watchlist classification problem. Figure C.5, shows the results of the MCC calculations. According to Figure C.5, LE, PD, IFRSS, and IR would have the highest predictive performance. In addition, the RUD, ROD, and MA triggers exhibit a modest but inverse correlation with the watchlist classifications. This is the case for the RUD, ROD, ESRT, SS, RCF, FBS, LE, EAD, RWA, and DPD triggers.

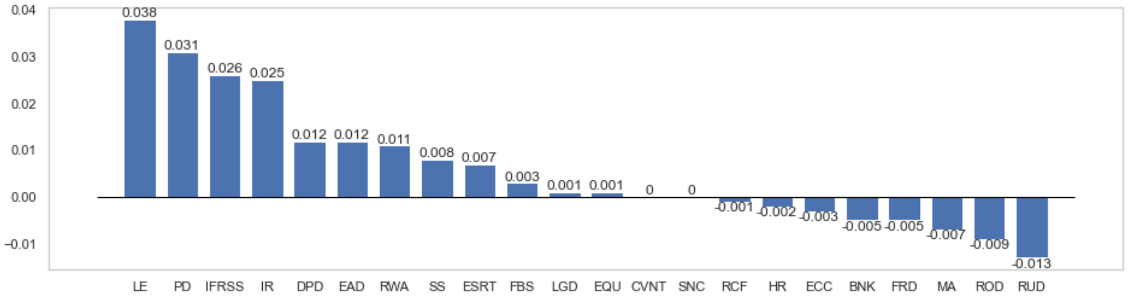


Figure C.5: Correlation per trigger

C.4. TIME LAG DISTRIBUTION

Figure C.6 provides the distribution of when triggers were raised compared to the month a migration occurred. Preferably, the distributions are skewed to the left meaning that triggers are able to detect financial distress at an early stage.

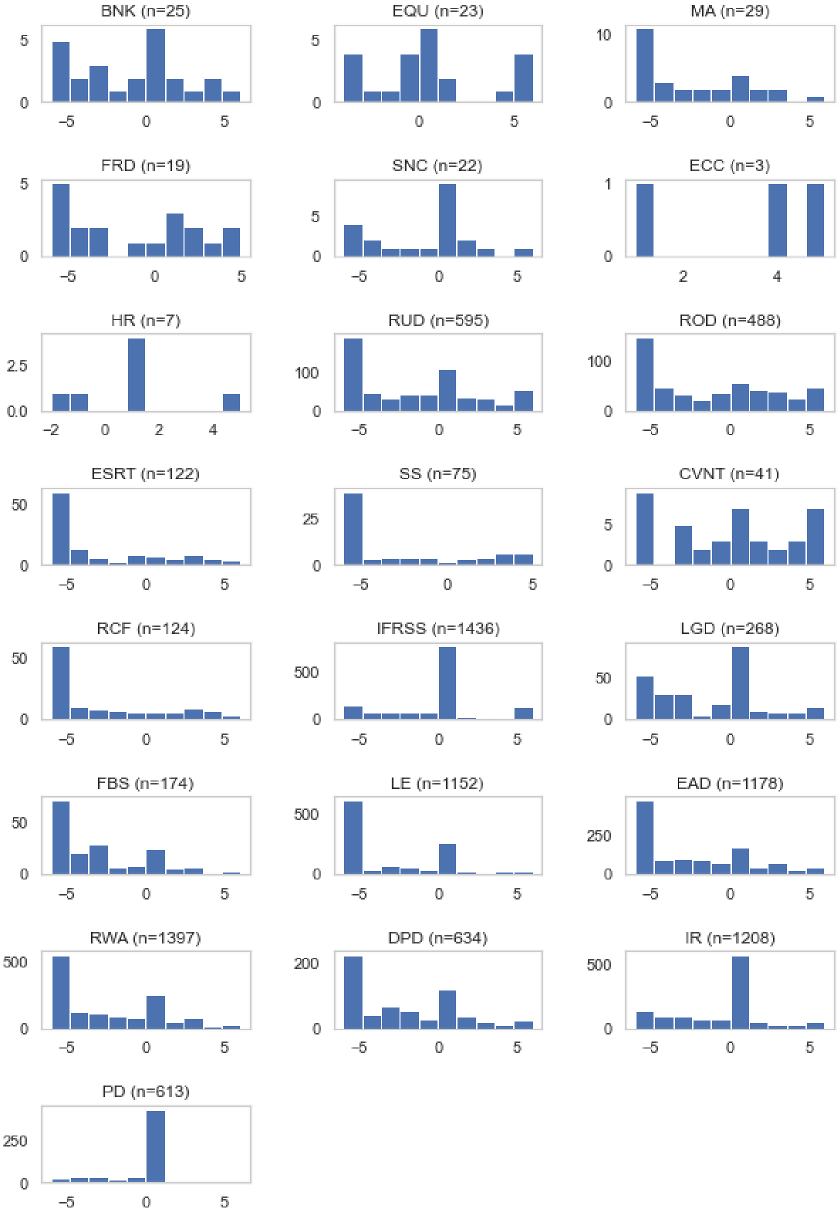


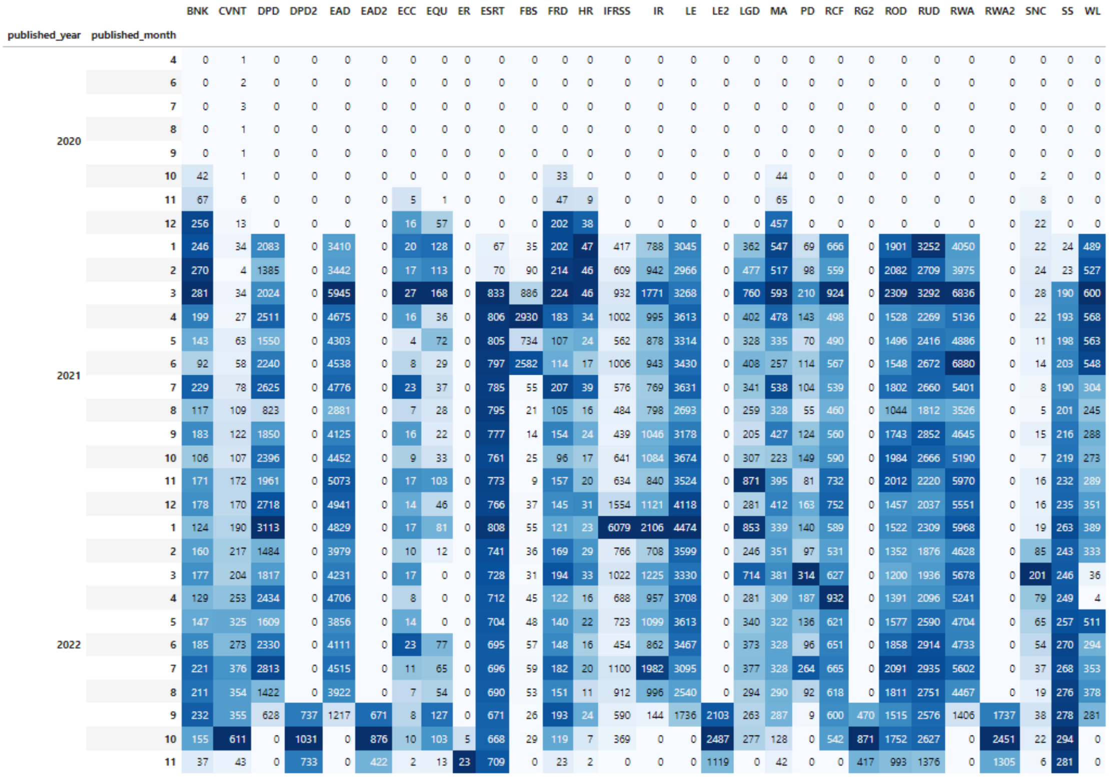
Figure C.6: Time lag distributions

D

Early Warning Trigger Distribution

D.1. TRIGGER FREQUENCY

Figure D.1 provides a heatmap of the number of triggers raised in a given month. In September 2022, a couple of triggers, which only showed red (severe) alerts, were redesigned to new triggers that provide both red and amber (less severe) warnings. As a result, many of these triggers are not raised after September, while others start occurring in that month.



published_year	published_month	BNK	CVNT	DPD	DPD2	EAD	EAD2	ECC	EQU	ER	ESRT	FBS	FRD	HR	IFRSS	IR	LE	LE2	LGD	MA	PD	RCF	RG2	ROD	RUD	RWA	RWA2	SNC	SS	WL
2020	4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	10	42	1	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0	44	0	0	0	0	0	0	0	2	0	0
	11	67	6	0	0	0	0	5	1	0	0	0	47	9	0	0	0	0	0	65	0	0	0	0	0	0	0	8	0	0
	12	256	13	0	0	0	0	16	57	0	0	0	202	38	0	0	0	0	0	457	0	0	0	0	0	0	0	22	0	0
2021	1	246	34	2083	0	3410	0	20	128	0	67	35	202	47	417	788	3045	0	362	547	69	666	0	1901	3252	4050	0	22	24	489
	2	270	4	1385	0	3442	0	17	113	0	70	90	214	46	609	942	2966	0	477	517	98	559	0	2082	2709	3975	0	24	23	527
	3	281	34	2024	0	5945	0	27	168	0	833	886	224	46	932	1771	3268	0	760	593	210	924	0	2309	3292	6836	0	28	190	600
	4	199	27	2511	0	4675	0	16	36	0	806	2930	183	34	1002	995	3613	0	402	478	143	498	0	1528	2269	5136	0	22	193	568
	5	143	63	1550	0	4303	0	4	72	0	805	734	107	24	562	878	3314	0	328	335	70	490	0	1496	2416	4886	0	11	198	563
	6	92	58	2240	0	4538	0	8	29	0	797	2582	114	17	1006	943	3430	0	408	257	114	567	0	1548	2672	6880	0	14	203	548
	7	229	78	2625	0	4776	0	23	37	0	785	55	207	39	576	769	3631	0	341	538	104	539	0	1802	2660	5401	0	8	190	304
	8	117	109	823	0	2881	0	7	28	0	795	21	105	16	484	798	2693	0	259	328	55	460	0	1044	1812	3526	0	5	201	245
	9	183	122	1850	0	4125	0	16	22	0	777	14	154	24	439	1046	3178	0	205	427	124	560	0	1743	2852	4645	0	15	216	288
	10	106	107	2396	0	4452	0	9	33	0	761	25	96	17	641	1084	3674	0	307	223	149	590	0	1984	2666	5190	0	7	219	273
	11	171	172	1961	0	5073	0	17	103	0	773	9	157	20	634	840	3524	0	871	395	81	732	0	2012	2220	5970	0	16	232	289
	12	178	170	2718	0	4941	0	14	46	0	766	37	145	31	1554	1121	4118	0	281	412	163	752	0	1457	2037	5551	0	16	235	351
2022	1	124	190	3113	0	4829	0	17	81	0	808	55	121	23	6079	2106	4474	0	853	339	140	589	0	1522	2309	5968	0	19	263	389
	2	160	217	1484	0	3979	0	10	12	0	741	36	169	29	766	708	3599	0	246	351	97	531	0	1352	1876	4628	0	85	243	333
	3	177	204	1817	0	4231	0	17	0	0	728	31	194	33	1022	1225	3330	0	714	381	314	627	0	1200	1936	5678	0	201	246	36
	4	129	253	2434	0	4706	0	8	0	0	712	45	122	16	688	957	3708	0	281	309	187	932	0	1391	2096	5241	0	79	249	4
	5	147	325	1609	0	3856	0	14	0	0	704	48	140	22	723	1099	3613	0	340	322	136	621	0	1577	2590	4704	0	65	257	511
	6	185	273	2330	0	4111	0	23	77	0	695	57	148	16	454	862	3467	0	373	328	96	651	0	1858	2914	4733	0	54	270	294
	7	221	376	2813	0	4515	0	11	65	0	696	59	182	20	1100	1982	3095	0	377	328	264	665	0	2091	2935	5602	0	37	268	353
	8	211	354	1422	0	3922	0	7	54	0	690	53	151	11	912	996	2540	0	294	290	92	618	0	1811	2751	4467	0	19	276	378
	9	232	355	628	737	1217	671	8	127	0	671	26	193	24	590	144	1736	2103	263	287	9	600	470	1515	2576	1406	1737	38	278	281
	10	155	611	0	1031	0	876	10	103	5	668	29	119	7	369	0	0	2487	277	128	0	542	871	1752	2627	0	2451	22	294	0
	11	37	43	0	733	0	422	2	13	23	709	0	23	2	0	0	0	1119	0	42	0	0	417	993	1376	0	1305	6	281	0

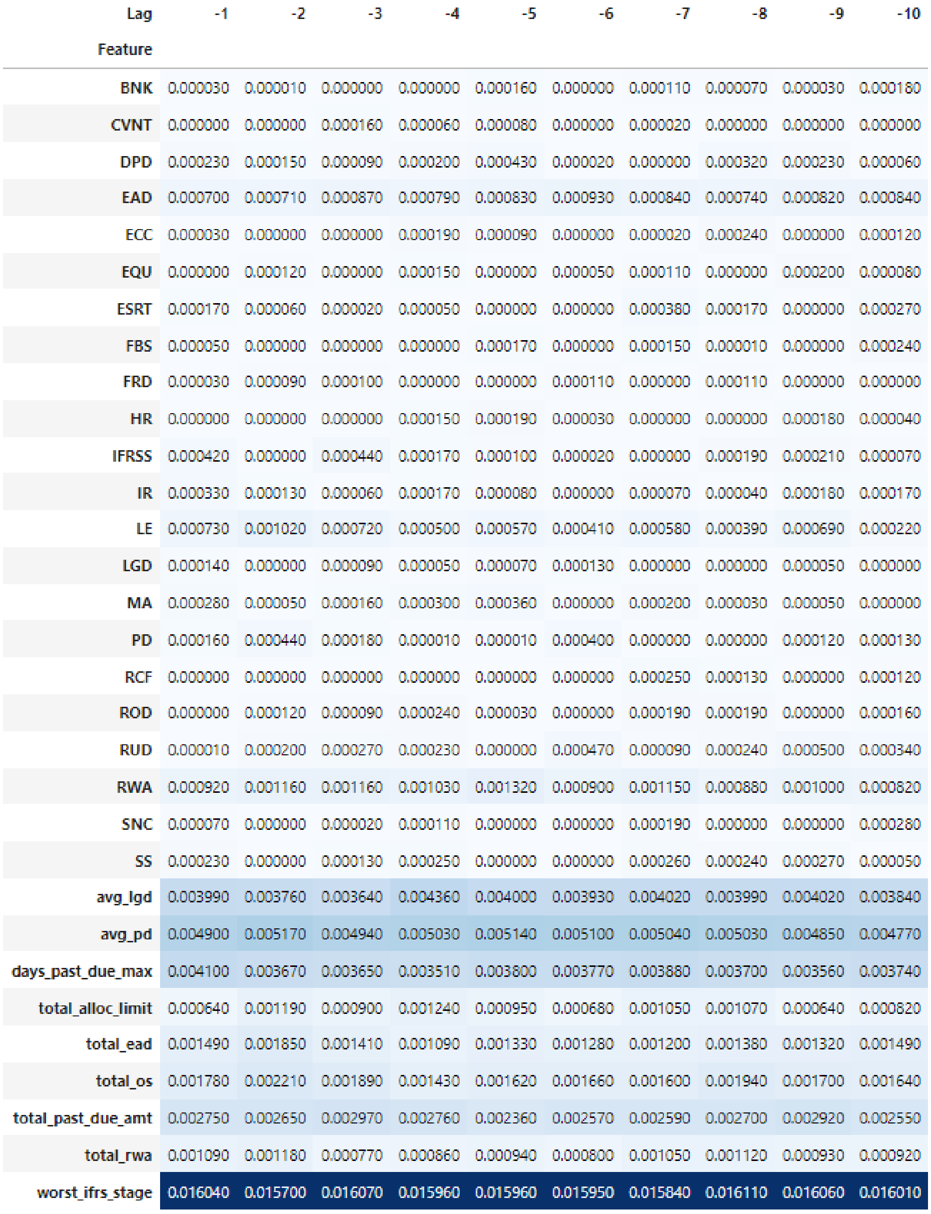
Figure D.1: Trigger frequency distribution

E

MUTUAL INFORMATION

E.1. MUTUAL INFORMATION HEATMAP

Figure E.2 provides a heatmap of the mutual information calculated for each feature. From this figure, we can conclude that the internal client data has the strongest relationship with the target variable, whereas the relationship with the early warning triggers is quite modest.



Lag	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
Feature										
BNK	0.000030	0.000010	0.000000	0.000000	0.000160	0.000000	0.000110	0.000070	0.000030	0.000180
CVNT	0.000000	0.000000	0.000160	0.000060	0.000080	0.000000	0.000020	0.000000	0.000000	0.000000
DPD	0.000230	0.000150	0.000090	0.000200	0.000430	0.000020	0.000000	0.000320	0.000230	0.000060
EAD	0.000700	0.000710	0.000870	0.000790	0.000830	0.000930	0.000840	0.000740	0.000820	0.000840
ECC	0.000030	0.000000	0.000000	0.000190	0.000090	0.000000	0.000020	0.000240	0.000000	0.000120
EQU	0.000000	0.000120	0.000000	0.000150	0.000000	0.000050	0.000110	0.000000	0.000200	0.000080
ESRT	0.000170	0.000060	0.000020	0.000050	0.000000	0.000000	0.000380	0.000170	0.000000	0.000270
FBS	0.000050	0.000000	0.000000	0.000000	0.000170	0.000000	0.000150	0.000010	0.000000	0.000240
FRD	0.000030	0.000090	0.000100	0.000000	0.000000	0.000110	0.000000	0.000110	0.000000	0.000000
HR	0.000000	0.000000	0.000000	0.000150	0.000190	0.000030	0.000000	0.000000	0.000180	0.000040
IFRSS	0.000420	0.000000	0.000440	0.000170	0.000100	0.000020	0.000000	0.000190	0.000210	0.000070
IR	0.000330	0.000130	0.000060	0.000170	0.000080	0.000000	0.000070	0.000040	0.000180	0.000170
LE	0.000730	0.001020	0.000720	0.000500	0.000570	0.000410	0.000580	0.000390	0.000690	0.000220
LGD	0.000140	0.000000	0.000090	0.000050	0.000070	0.000130	0.000000	0.000000	0.000050	0.000000
MA	0.000280	0.000050	0.000160	0.000300	0.000360	0.000000	0.000200	0.000030	0.000050	0.000000
PD	0.000160	0.000440	0.000180	0.000010	0.000010	0.000400	0.000000	0.000000	0.000120	0.000130
RCF	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000250	0.000130	0.000000	0.000120
ROD	0.000000	0.000120	0.000090	0.000240	0.000030	0.000000	0.000190	0.000190	0.000000	0.000160
RUD	0.000010	0.000200	0.000270	0.000230	0.000000	0.000470	0.000090	0.000240	0.000500	0.000340
RWA	0.000920	0.001160	0.001160	0.001030	0.001320	0.000900	0.001150	0.000880	0.001000	0.000820
SNC	0.000070	0.000000	0.000020	0.000110	0.000000	0.000000	0.000190	0.000000	0.000000	0.000280
SS	0.000230	0.000000	0.000130	0.000250	0.000000	0.000000	0.000260	0.000240	0.000270	0.000050
avg_lgd	0.003990	0.003760	0.003640	0.004360	0.004000	0.003930	0.004020	0.003990	0.004020	0.003840
avg_pd	0.004900	0.005170	0.004940	0.005030	0.005140	0.005100	0.005040	0.005030	0.004850	0.004770
days_past_due_max	0.004100	0.003670	0.003650	0.003510	0.003800	0.003770	0.003880	0.003700	0.003560	0.003740
total_alloc_limit	0.000640	0.001190	0.000900	0.001240	0.000950	0.000680	0.001050	0.001070	0.000640	0.000820
total_ead	0.001490	0.001850	0.001410	0.001090	0.001330	0.001280	0.001200	0.001380	0.001320	0.001490
total_os	0.001780	0.002210	0.001890	0.001430	0.001620	0.001660	0.001600	0.001940	0.001700	0.001640
total_past_due_amt	0.002750	0.002650	0.002970	0.002760	0.002360	0.002570	0.002590	0.002700	0.002920	0.002550
total_rwa	0.001090	0.001180	0.000770	0.000860	0.000940	0.000800	0.001050	0.001120	0.000930	0.000920
worst_ifrs_stage	0.016040	0.015700	0.016070	0.015960	0.015960	0.015950	0.015840	0.016110	0.016060	0.016010

Figure E.1: Mutual information between independent and dependent variables

E.2. MUTUAL INFORMATION LINE GRAPHS

Figure E.2 plots the mutual information against the time lag for each variable. This indicates for which time lag the relationship between the independent and dependent variables is the strongest.

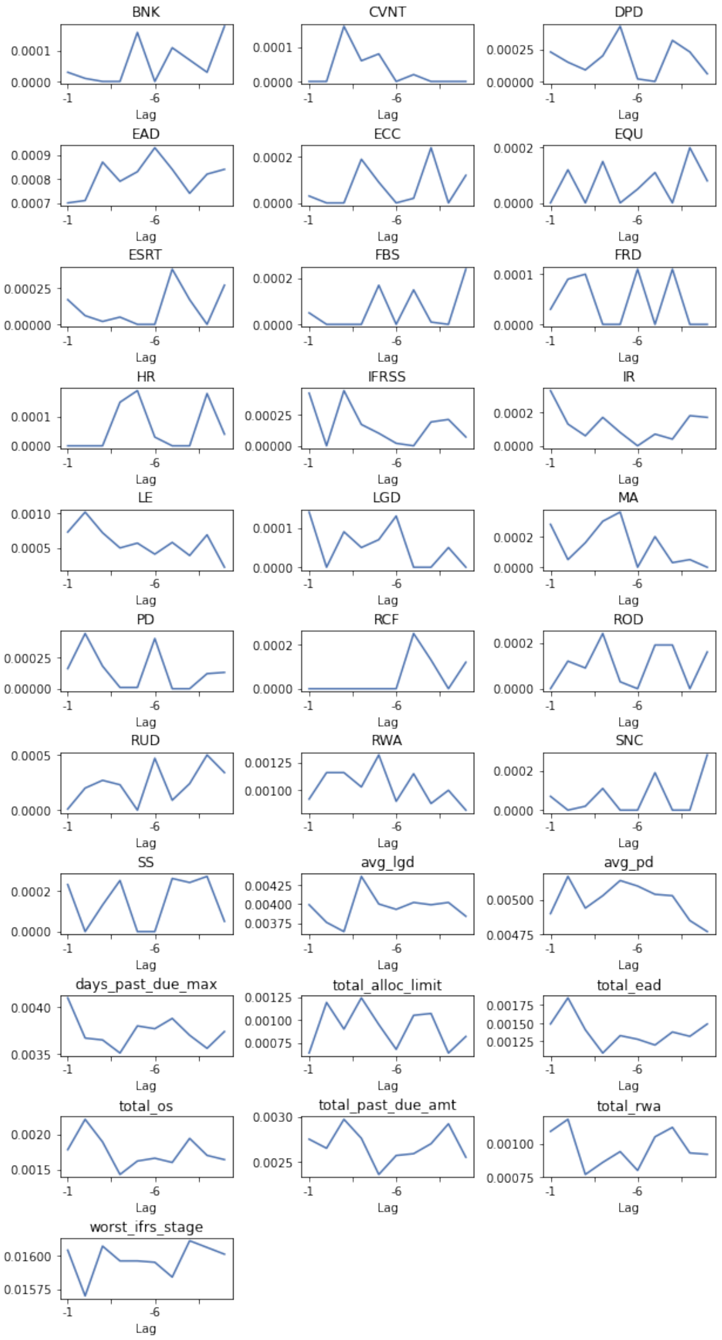


Figure E.2: Mutual information of each independent variable against the time lag

F

Optimal Hyperparameters

This appendix provides an overview of the hyperparameter space used and the optimal hyperparameters found for each model configuration.

F.1. Hyperparameter Space

Table F.1: Hyperparameter space

Model	Parameter space
DT	percentile : [5, 10, 15, 20, 30]
	sampling_strategy : [0.5, 0.6, 0.7, 0.9, 1]
	max_features: [“auto”, “sqrt”, “log2”]
	ccp_alpha: [0.1, .01, .001]
	max_depth : [5, 6, 7, 8, 9]
	criterion : [“gini”, “entropy”]
LDA	percentile : [5, 10, 15, 20, 30]
	sampling_strategy : [0.5, 0.6, 0.7, 0.9, 1]
	solver : [“svd”, “lsqr”, “eigen”]
GMB	percentile : [5, 10, 15, 20, 30]
	sampling_strategy : [0.5, 0.6, 0.7, 0.9, 1]
	criterion: [friedman_mse]
	loss: [“log_loss”,exponential]
	max_features: [log2,sqrt]
	learning_rate: [0.01,0.05,0.1,1,0.5]
	max_depth: [3,4,5]
	min_samples_leaf: [4,5,6]
	subsample: [0.6,0.7,0.8]

	n_estimators: [5,10,15,20]
LG	percentile : [5, 10, 15, 20, 30]
	sampling_strategy : [0.5, 0.6, 0.7, 0.9, 1]
	solver : [“lbfgs”, “liblinear”]
	penalty : [“none”, “l1”, “l2”, “elasticnet”]
	C : list(np.logspace(-4, 4, 50))
SVM	percentile : [5, 10, 15, 20, 30]
	sampling_strategy : [0.5, 0.6, 0.7, 0.9, 1]
	C: [0.1,1, 10, 50]
	gamma: [1,0.1,0.01,0.001]
	kernel: [“rbf”, “poly”, “sigmoid”]
RF	percentile : [5, 10, 15, 20, 30]
	sampling_strategy : [0.5, 0.6, 0.7, 0.9, 1]
	max_depth: [3,5,10, None]
	n_estimators: [10,50,100,150]
	max_features: [1,3,5,7,’sqrt’,’log2’]
	min_samples_leaf: [1,2,3]
	min_samples_split: [1,2,3]

XGB	percentile : [5, 10, 15, 20, 30]
	sampling_strategy : [0.5, 0.6, 0.7, 0.9, 1]
	subsample: [0.5, 0.75, 1]
	colsample_bytree: [0.3, 0.5, 0.8]
	max_depth: [2, 6, 12]
	min_child_weight: [1,5,15]
	learning_rate: [0.1, 0.2, 0.5]
	scale_pos_weight: [1, 3, 5]
ANN2	percentile : [5, 10, 15, 20, 30]
	sampling_strategy : [0.5, 0.6, 0.7, 0.9, 1]
	learn_rate : [0.001, 0.01, 0.1]
	dropout_rate : [0.2, 0.5, 0.8]
	neurons1 : [32, 64, 128]
	neurons2 : [16, 32, 64]
	batch_size : [16, 32, 64]
	epochs : [50, 100, 150, 300]
ANN3	percentile : [5, 10, 15, 20, 30]
	sampling_strategy : [0.5, 0.6, 0.7, 0.9, 1]
	learn_rate : [0.001, 0.01, 0.1]
	dropout_rate : [0.2, 0.5, 0.8]
	neurons1: [32, 64, 128]
	neurons2: [16, 32, 64]
	neurons3: [8, 16, 32]
	batch_size : [16, 32, 64]
	epochs : [50, 100, 150, 300]

F.2. OPTIMAL PARAMETERS MODEL PERFORMANCE

Table F.2: Optimal hyperparameters for each ML model

Model	Optimal hyperparameters	F1 score
DT	sampling_strategy: 0.6 max_features: sqrt max_depth: 8 criterion: gini ccp_alpha: 0.001 percentile: 5	0.168
LDA	sampling_strategy: 0.9 solver: lsqr percentile: 30	0.154
GBM	sampling_strategy: 0.7 subsample: 0.6 n_estimators: 15 min_samples_leaf: 5 max_features: log2 max_depth: 5 loss: log_loss learning_rate: 0.1 criterion: friedman_mse percentile: 30	0.177
LG	sampling_strategy: 0.7 solver: liblinear penalty: l1 C: 24.420530945486497 percentile: 30	0.168
SVM	sampling_strategy: 0.7 kernel: rbf gamma: 1 C: 10 percentile: 20	0.137
RF	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.321

XGB	sampling_strategy: 0.5 subsample: 0.75 scale_pos_weight: 1 min_child_weight: 5 max_depth: 12 learning_rate: 0.2 colsample_bytree: 0.3 percentile: 30	0.269

ANN2	sampling_strategy: 0.5	0.229
	neurons2: 32	
	neurons1: 32	
	learn_rate: 0.01	
	epochs: 100	
	dropout_rate: 0.5	
	batch_size: 16	
	percentile: 20	
ANN3	sampling_strategy: 0.6	0.216
	neurons3: 32	
	neurons2: 32	
	neurons1: 64	
	learn_rate: 0.01	
	epochs: 150	
	dropout_rate: 0.5	
	batch_size: 16	
	percentile: 20	

F.3. OPTIMAL PARAMETERS PREDICTION APPROACH

Table F.3: Optimal RF hyperparamters for each prediction approach

Prediction approach	Optimal hyperparameters	F1 score
Negative Migration	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: 5 percentile: 30	0.321
Default Migration	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.307
Watchlist Migration	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.173

F.4. OPTIMAL PARAMETERS TIME WINDOW

Table F.4: Optimal RF hyperparameters for different time window sizes

Time window	Optimal hyperparameters	F1 score
2	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.258
4	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.281
6	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.321
8	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.343
10	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.343

F.5. OPTIMAL PARAMETERS TARGET WINDOW

Table F.5: Optimal RF hyperparameters for different target window sizes

Target window	Optimal hyperparameters	F1 score
1	sampling_strategy: 0.6 n_estimators: 10 min_samples_split: 3 min_samples_leaf: 1 max_features: 1 max_depth: 3 percentile: 30	0.093
2	sampling_strategy: 0.5 n_estimators: 100 min_samples_split: 4 min_samples_leaf: 2 max_features: 7 max_depth: None percentile: 15	0.135
4	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.244
6	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.321

F.6. OPTIMAL PARAMETERS TIME GAP

Table F.6: Optimal RF hyperparamters for different time gap sizes

Time gap	Optimal hyperparameters	F1 score
1	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.321
2	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.329
3	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.336
4	sampling_strategy: 0.5 n_estimators: 50 min_samples_split: 2 min_samples_leaf: 1 max_features: 5 max_depth: None percentile: 30	0.356

G

WL Trigger Metrics

G.1. Table with Early Warning Trigger Metrics

Table G.1 provides an overview of the data used to calculate the trigger precision and migration sensitivity for the ARIA triggers and WL triggers based on the ML models. Trigger frequency (Trigger Freq.) is the number of times a trigger was raised, and trigger hit is the number of times a negative migration occurred after the next six months the trigger was raised. Migration frequency (Mig. Freq.) is the number of times a migration occurred, and migration hit (Mig. Hit) is the number of times a migration received a trigger six months in advance.

Table G.1: Early warning trigger metrics for ARIA and WL triggers

Feature	**Trigger Hit**	**Trigger Freq.**	**Trigger Precision**	**Trigger Precision SE**	**Mig. Hit**	**Mig. Freq.**	**Migration Sensitivity**	**Migration Sensitivity SE**
RF	1202	9474	0.127	0.009	393	581	0.676	0.05
XGB	1151	12247	0.094	0.007	404	581	0.695	0.049
LG	475	7438	0.064	0.007	214	581	0.368	0.052
GBM	542	9053	0.06	0.006	221	581	0.38	0.052
FBS	6	112	0.054	0.054	6	581	0.01	0.011
PD	20	412	0.049	0.027	60	581	0.103	0.033
LE	448	9523	0.047	0.006	225	581	0.387	0.052
DT	447	10362	0.043	0.005	192	581	0.33	0.05
SVM	488	11514	0.042	0.005	182	581	0.313	0.05
IR	169	4228	0.04	0.008	207	581	0.356	0.051
LDA	567	15883	0.036	0.004	229	581	0.394	0.052
IFRSS	89	2790	0.032	0.009	203	581	0.349	0.051
ANN2	90	3012	0.03	0.008	68	581	0.117	0.034
ANN3	73	2626	0.028	0.008	56	581	0.096	0.032
ESRT	50	2294	0.022	0.008	25	581	0.043	0.022
DPD	144	7560	0.019	0.004	158	581	0.272	0.048
HR	1	60	0.017	0.03	0	581	0	0
RWA	239	18123	0.013	0.002	249	581	0.429	0.053
ROD	80	6809	0.012	0.003	66	581	0.114	0.034
EQU	2	173	0.012	0.016	5	581	0.009	0.009
LGD	14	1257	0.011	0.008	28	581	0.048	0.023
EAD	164	15433	0.011	0.002	215	581	0.37	0.052
RCF	25	2390	0.01	0.005	24	581	0.041	0.021
RUD	108	10397	0.01	0.003	83	581	0.143	0.037
CVNT	11	1134	0.01	0.007	7	581	0.012	0.012
SNC	1	156	0.006	0.011	1	581	0.002	0.003
BNK	4	709	0.006	0.006	1	581	0.002	0.003
SS	4	827	0.005	0.006	2	581	0.003	0.005
MA	5	1196	0.004	0.004	6	581	0.01	0.011
FRD	2	593	0.003	0.005	1	581	0.002	0.003
ECC	0	53	0	0	0	581	0	0

G.2. MIGRATION SENSITIVITY

Figure G.1 shows the migration sensitivity for the ARIA and WL triggers with the 99% confidence intervals. This graph indicates that the proposed WL triggers based on the XGB and RF models significantly outperform the other triggers when it comes to migration sensitivity.

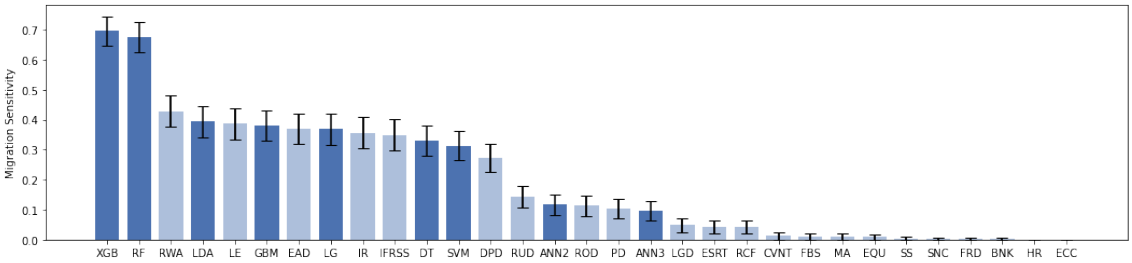


Figure G.1: Migration sensitivity for the ARIA (light blue) and WL triggers (dark blue)

G.3. TRIGGER PRECISION

Figure G.2 shows the trigger precision for the ARIA and WL triggers with the 99% confidence intervals. Also, in this case, the WL triggers based on the XGB and RF models outperform the other triggers when looking at the trigger precision

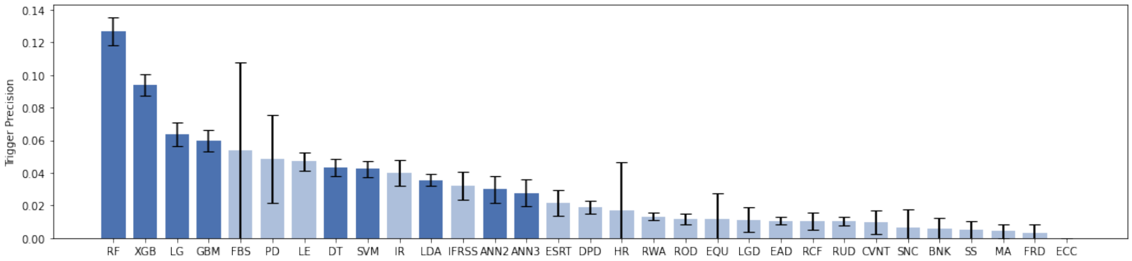


Figure G.2: Trigger precision for the ARIA (light blue) and WL triggers (dark blue)

H

Recommendations

This Appendix provides more detailed recommendations based on the future research and limitations described in Section 7.2.

H.1. Predictive Performance and Earliness Tradeoff

This research defines and analyses three metrics related to the effectiveness of the WL trigger. However, due to time constraints, how these dimensions relate to each other has not been investigated. For future research, risk managers and account managers could be consulted to discuss how important these dimensions are compared to each other. Based on their judgement, the model could be fine-tuned and configured accordingly to the desires of the stakeholders. For instance, the target window and the gap need to be set based on a tradeoff between the timeliness of the models and their predictive performance.

Furthermore, the threshold could be configured to create the ideal balance between the migration sensitivity and trigger precision of the WL trigger. Instead of the F1 score, the more general F-beta score could be used to incorporate the difference in importance of the trigger precision and migration sensitivity [83]. The F-beta score has a parameter beta that controls the importance of precision and sensitivity (See Equation H.1). This metric could then be used for model tuning and evaluation.

$$F_\beta = \frac{(1+\beta^2)\cdot(\text{precision}\cdot\text{sensitivity})}{(\beta^2\cdot\text{precision})+\text{sensitivity}} \tag{H.1}$$

In addition, due to limited historical data, measuring the time lag for the proposed WL triggers was not possible. Therefore, when more data is available, more extensive experimentation could be done to measure how the increasing time lag influences the predictive performance of the models. Also, this was the reason that only time windows of six months were considered to measure the time lag, migration sensitivity, and trigger precision. For future research, these time windows could also be extended to measure the possible long-term predictability of the triggers.

H.2. MODEL PERFORMANCE ESTIMATION

In this research, the analysis of the model performance is only limited to calculating the estimates of the metrics but it overlooks the uncertainty of these measurements. Noise introduces random fluctuations that contribute to uncertainty when trying to measure the performance of a model. As a result, it is difficult to precisely measure the true value of the performance metrics. For that reason, measuring the uncertainty of an estimate is essential to gain a more comprehensive understanding of the reliability and robustness of the results.

This uncertainty can be measured with confidence intervals providing a range of values within which the true measurement is likely to fall [84]. Confidence intervals can be constructed using the cross-validation or the bootstrap sampling method.

The expanding-window forward cross-validation technique has been used to calculate the average performance metrics over four consecutive folds (See Section **??**). Unfortunately, in our case, the cross-validation method is not an appropriate method for calculating the confidence intervals since we cannot assume independence between the different folds due to their temporal dependency [85].

Bootstrap sampling is a resampling technique that involves iteratively drawing samples with replacement from the data [86]. This way different data sets can be constructed so that the models can be tested several times resulting in multiple measurements of the performance. Consequently, the confidence interval can be derived from the distribution of the measurements by taking the according percentile ranges. A disadvantage of bootstrap sampling is that it is computationally very expensive. For each constructed sample the models need to be fine-tuned and trained again. For that reason, calculating the confidence intervals was left out of the scope of this research. For future research when more time and computational resources are available, it would be interesting to calculate the confidence interval so that conclusions can be made based on statistical significance.

H.3. PREDICTION APPROACH

The proposed financial distress prediction model only focuses on predicting negative migrations. For future research, it could be interesting to investigate the possibility of establishing a model that can detect other types of migrations. For example, the model could be extended to incorporate positive migrations where a watchlisted client receives a regular status again because of improvements in their credit risk.

One suggestion is to research a different prediction approach that focuses on predicting the status of a client instead of migrations. This model could try to predict if a client will have a regular, default or watchlist status in the next month. As a result, this model would be able to determine if a client will keep its status or whether the client will migrate to another status. The difficult part of such a model is that most of the time, the status of a client remains the same. This could lead to misleading results when evaluating the model performance, so considerations need to be made to tackle this problem.

Another suggestion is that other target variables that relate to possible actions could be researched in the future. The proposed model only identifies which clients are in financial distress, but it does not consider what possible actions are the most suitable to take based on the data. For instance, some clients might only require additional monitoring, while others require more critical forbearance measures. For that reason, it could be interesting to investigate how the proposed model would work on a multi-class problem consisting of multiple actions for different severity levels.

H.4. MODEL IMPROVEMENTS

The proposed model could be improved by trying out other techniques and doing more extensive model tuning. RUS was selected to deal with the class imbalance because it significantly reduces the computational time required to train the models. If more computational resources are available, other techniques, such as ROS and SMOTE could be used. These methods might improve performance because these methods allow for all the data to be used. Furthermore, only mutual information was tested for feature selection due to time constraints. In the future, trying out different feature selection techniques to determine which method performs best could be interesting. Also, more extensive feature engineering could be performed. For instance, the difference in values between the current month and the previous month could be designed as a feature to reflect changes over time. Also, features based on (weighted) moving averages could be considered. Besides, more extensive hyperparameter optimization could be done by increasing the number of iterations or defining a larger parameter space.

Moreover, other ML models could be considered. In this research, only a selection of supervised learning models was tested, but in the future other classification models, including LightGBM, CatBoost, K-Nearest Neighbour, Naive Bayes, or Graphical Models, could be incorporated into the proposed financial distress prediction model. Two other classification approaches that could be interesting to investigate further are Hidden Markov Models (HMM) and early time series classification.

HMM could be a suitable approach for modelling the client status. HMM can be used to model sequential data where the underlying process generating the data is assumed to be a Markov process [87]. The states of the Markov process cannot be directly observed, but they can be inferred from observed data related to these hidden states. In our case, the transitions between a regular, watchlist, and client status could be modelled according to a Markov process. Then, the trigger data could be used as observed data to infer the state of the Markov process by calculating the probability that a particular sequence of statuses could occur for a client. Several HMM models have been implemented in a financial context, including default prediction, but more research would be needed to determine their applicability to financial distress prediction modelling [88–90].

The earliness of the proposed model can be altered by configuring the time gap and target window. However, a disadvantage of this approach is that there is no incentive for the model to make predictions as early as possible. Ideally, some logic or function should be in place that favours early predictions and penalizes late ones when training the models. Early time series classification is a supervised learning approach that aims to classify time series using as few observations as possible [91]. These models try to balance timeliness and accuracy by incorporating these objectives into the cost function of the classification model. These models have been adopted in different domains, such as medical diagnostics, process monitoring, and electricity usage [92, 93]. However, to our knowledge, no studies implement these kinds of models for the early detection of financial distress. Therefore, for future research, it would be interesting to investigate the applicability of these models for EWSs.

The approaches mentioned above are based on the supervised learning paradigm. This assumes that the labels are known and that these labels can be predicted using classification algorithms. However, other approaches based on different assumptions might improve the earliness and preciseness of detecting financial distress. In contrast to supervised learning, unsupervised learning is a type of machine learning where the algorithm learns patterns and relationships in data without being explicitly trained or labelled. Unsupervised learning could be used to cluster clients based on their credit risk. These clusters might be able to identify clients in financial distress (watchlist or default status) or with similar characteristics. In addition, these clusters could be included as a new feature in the supervised learning models to increase model performance.

Besides, semi-supervised learning could offer a middle way for both approaches. Semi-supervised learning assumes that only a few labels are known while the rest of the instances are unlabeled. In our case, we only have the labels of clients in default or on the watchlist. But these labels might be incomplete as there might still be undetected clients with neither a default nor watchlist status but might still be in financial distress. This is similar to a positive unlabelled learning problem, where there is only positively labelled data, and the remaining data is unlabeled [94].

H.5. DATA IMPROVEMENTS

The experimental results in Chapter 6 show that more historical data significantly increases the performance of the models by extending the time and target window. More research is needed to find the optimal window sizes that perform best. Also, more historical data would make it possible to incorporate seasonality into the model by, for instance, adding the months as a feature. In addition, more historical data would mean that there are more samples for training the models resulting in a better performance.

In addition, other external data sources could be added to improve the model's predictive performance. For instance, macroeconomic variables related to interest rates, Gross Domestic Product (GDP), inflation, and unemployment rates could be incorporated. Also, next to the equity trigger, market variables like market volatility, commodity prices, and market indices could be used to improve predictability. Depending on the sector and country of incorporation, these variables might impact clients' credit risk. For example, interest rates could influence the credit risk of other financial institutions, while oil prices have a more significant impact on companies in the energy sector.

In addition, more detailed internal data could be added as well. When looking at the Information Gain and SHAP values, we observe that the internal data on which the triggers are based contain the most information about negative migrations. Therefore, we think that adding more detailed data instead of only using the triggers would improve performance. For instance, the actual equity prices could be used instead of only using the EQU trigger, and the actual sentiment scores from the topic models could be used instead of the news-based triggers (BNK, MA FRD, ECC, SNC, HR). Another reason for using the underlying data instead of the triggers is that the logic of the triggers can change over time which would require recalculating or collecting the trigger data to get enough training data.

Finally, the data quality could be improved considerably. The internal client data set contained many missing values, which decreased the performance of the models. Also, outliers in the number of negative migrations discussed in Section 3.2.1 imply issues with data quality. So, by ensuring the quality of the input data used for the models, the performance could be increased.

I

MODEL DEPLOYMENT

Some thoughts and recommendations are dedicated to the future deployment of the proposed WL trigger. Machine Learning Operations (MLOps) is a set of practices and technologies designed to help manage the lifecycle of machine learning models [95]. This research establishes a foundation for a future WL trigger by exploring and experimenting with the feasibility of such a trigger based on a financial distress prediction model. However, for the final implementation, many steps still need to be considered before the new trigger can be deployed into the ARIA application. Some recommendations are given on the following MLOps components:

- Data management
- ML pipeline
- Automated retraining
- Performance monitoring

Firstly, access to data sources needs to be ensured through reliable data pipelines. Currently, the model has been tested on extracted files, but connections need to be established with the according data sources for future implementation. The proposed model requires access to the records of the ARIA triggers and the internal customer data. Furthermore, more data sources could be utilized in the future. The research showed that the underlying data on which the triggers have the biggest impact on the predictions. Therefore, other data pipelines could be created that give access to internal customer data (e.g. data used to calculate the PD or RWA), sentiment scores from the topic models, and market data from Refinitiv.

Secondly, ML models are implemented in a Jupyter Notebook, but for the deployment, the ML pipeline needs to be built. A ML pipeline is a series of interconnected steps that automate the flow of data and tasks for the financial distress prediction models. During the research, several ML models were tested. However, only the best-performing model will be deployed when the model is implemented. Therefore, we suggest that the ML pipeline would train several classifiers and then selects and deploys the model with the highest F-beta score where beta depends on the tradeoff between sensitivity and precision. Furthermore, the time window and target window were manually configured during the experiments, but in the future, these models could also be treated as hyperparameters that could be optimized using the random search. Besides, the time lag, threshold, and target variable of the model could be configured based on the desires of the users.

Moreover, the retraining of the pipeline needs to be considered to ensure that the newly available data can be incorporated into the models. For this, we propose setting a trigger that is activated at the beginning of each month because the predictions are made on a monthly basis. This would start the retraining of the ML pipeline so that the last month's data can be incorporated into the predictions for the coming months. Such a system could be orchestrated by a platform like Airflow, which is also used to retrain the topic models for the external triggers at ING. Airflow can be used to orchestrate the workflow as a Directed Acyclic Graph (DAG). An example of such a DAG for our financial distress prediction model is depicted in Figure I.1.

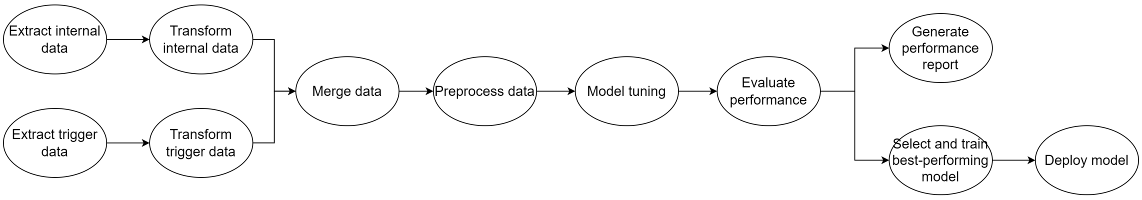


Figure I.1: Proposed DAG architecture

Finally, the performance of the ML models needs to be continuously monitored. Currently, the model has been evaluated based on historical data sets. A limitation of this approach is that it does not consider model drift and interaction effects with the users. Model drift occurs when there are changes in the distribution of the underlying data, which can result in the deterioration of the model over time. This input drift could be detected using univariate statistical tests like the Kolmogorov-Smirnov and Chi-square tests for continuous and categorical features, respectively [95]. In addition, mutual information and SHAP values could also be used to analyse how the impact of each feature changes over time. Moreover, when the model has been implemented, the behaviour of the users might change, resulting in different watchlist labels in the future. For future experimentation, applying A/B testing for the online monitoring of several candidate models could be interesting. In this case, several candidate models are tested on different groups of users. Then, based on the evaluation metrics and early warning triggers, the models could be compared to determine which model performs best. Appendix I.2 provides a flow chart of how the overall pipeline could be designed inspired by Google [96].

I.1. AUTOMATED PIPELINE

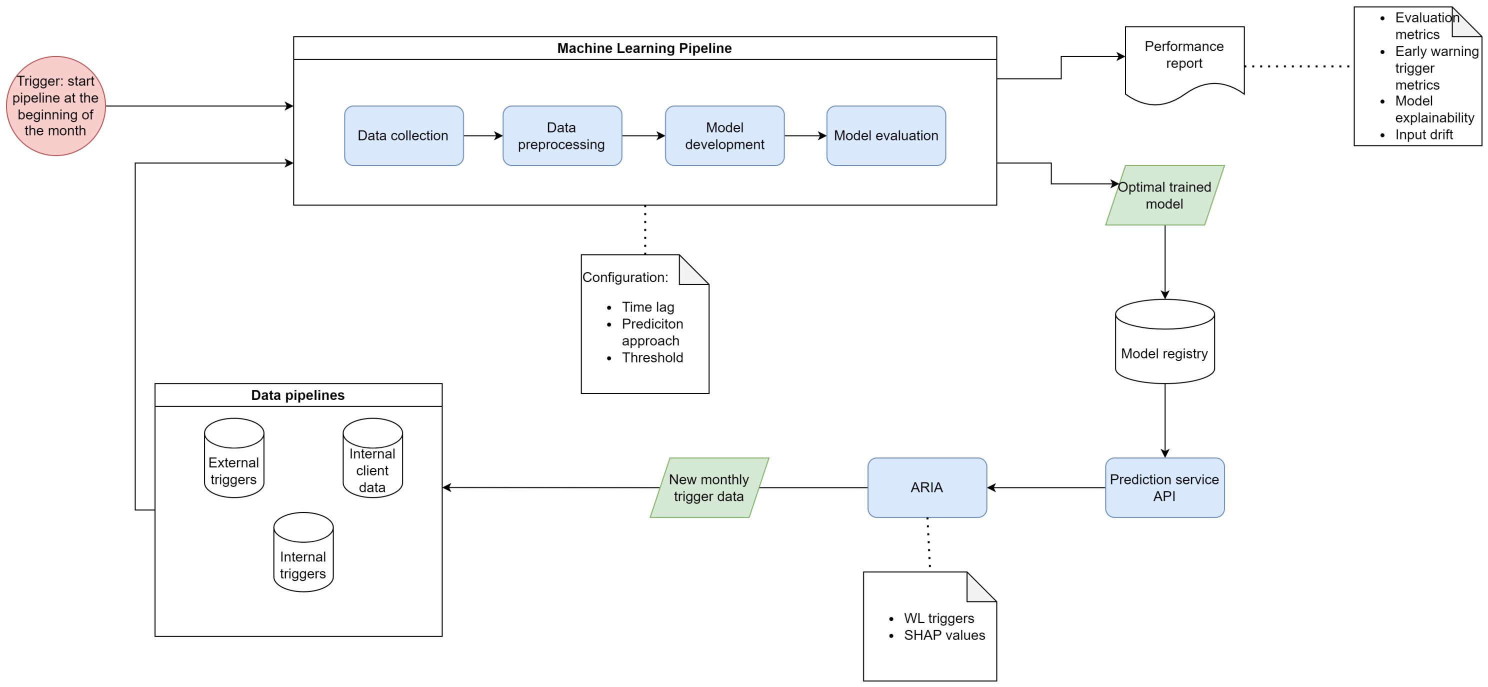


Figure I.2: Flow chart of the automated pipeline architecture