An Active Learning Approach for Reject Inference in Credit Scoring using Conformal Prediction Intervals on Real and Semi-Artificial Data

Master Thesis

to obtain the degree
Master of Science
in Business Administration

Submitted by

Maximilian Suliga
616520

Submitted to:
Prof. Dr. Stefan Lessmann
and
Prof. Dr. Benjamin Fabian
Humboldt-Universität zu Berlin
School of Business and Economics
Chair of Information Systems

Berlin, June 2, 2023

Table of Contents

List of Tables

List of Algorithms

List of Figures

List of Abbreviations

AL	Active Learning
AUC	Area Under the Receiver Operating Curve
BAD	Defaulted Customer
bad	Predicted Defaulting Customer
BS	Brier Score
CFN	Costs Arising from a False Negative
CFP	Costs Arising from a False Positive
FN	False Negative
FP	False Positive
GOOD	Non-Defaulted Customer
good	Predicted Non-Defaulting Customer
ICP	Inductive Conformal Predictor
LGD	Loss Given Default
MC	Monte Carlo
NN	Nearest Neighbor
P2P	Peer-to-Peer
PAUC	Partial Area Under the Receiver Operating Curve
PCC	Percentage Correctly Classified
PD	Probability of Default
PG	Partial Gini Index
RI	Reject Inference
TCP	Transductive Conformal Predictor
TN	True Negative
TP	True Positive
WoE	Weight of Evidence

1. Introduction

The rise of credit cards marks a significant era in the credit industry, having led to a transition from human judgment to automated decision-making in loan granting (Dumitrescu et al., 2021). With this development, the field of Credit Scoring evolved in academic literature, which continues to evolve alongside ongoing advancements in the industry despite regulatory requirements constraining adoptions of state-of-the-art techniques in practice (Lessmann et al., 2015). The adoption of one of the industry's recent developments, buy now, pay later (BNPL), shows that the industry is still expanding[1].These point-of-sale loans, some lasting as little as two weeks[2], serve as a fitting counterpart to credit cards, which are already characterized by their relatively short loan durations. With the expansion of products allowing deeper penetration of existing markets, product and technology innovation also allow to tap into new markets of so-called unbanked populations, which have not been served by traditional banks in any manner. Microcredit, a form of small consumer loans without collateral and tailored to high-risk individuals, exemplifies this approach (Song et al., 2022). A popular example is Microcredit service M-Shwari, which provides loans based on the data gathered through the mobile money service M-Pesa. It has been shown that through data from M-Shwari, unbanked consumers were able to build themselves a credit score which made them eligible for loans at conventional banks (Suri et al., 2021).

Whereas unbanked consumers concentrate mainly on undeveloped and emerging economies, the problem of no lending data for rejected loan applicants exists for all economies. Financial institutions face the task of training their Machine Learning models on biased samples that only represent accepted applications, as creating a fully representative sample encompassing all applicants, including those likely to default, is economically unfeasible.

Reject Inference (RI) refers to the techniques that try to un-bias the Machine Learning model from its selection bias. The limited success of some has brought up the idea of selecting a limited number of applicants who were supposed to be rejected, giving them a loan and observing their repaying behavior in order to make a model learn from it (Crone & Finlay, 2012; Hand & Henley, 1993). The selection of only the most informative rejected applications within the constraints of economic viability, corresponding to selecting the most

[1] See https://www.mckinsey.com/industries/financial-services/our-insights/reinventing-credit-cards-responses-to-new-lending-models-in-the-us
[2] An example is Riverty, one of the biggest BNPL providers https://www.riverty.com/en/how-it-works/payment-methods/

informative instances under a defined budget in a general setting, is achieved through Active Learning (AL).

Despite its early suggestion and supporting factors like high class imbalance (Carcillo et al., 2018), cost imbalance (Min et al., 2019), bias (Richards et al., 2012), and potential data scarcity (Carta et al., 2020; Saia & Carta, 2016), the literature on AL in the Credit Scoring context is limited. One potential reason for this is the time delay associated with granting a loan and observing the complete repayment behavior of borrowers, particularly in cases where loan durations can be as long as 30 years. However, given the ongoing advancements in consumer loans, this time delay becomes less significant in the context of Credit Scoring. Furthermore, recent progress in survival analysis instills confidence in circumventing the time delay, which has hindered AL from realizing its full potential in Credit Scoring (e.g., Blumenstock et al., 2022).

AL relies on query functions that quantify the informativeness of an instance. Conformal prediction intervals are an established way of quantifying confidence in prediction results and have already been used in the fields of regulation for toxicology (Norinder et al., 2014), pharmaceutics (Ahlberg et al., 2015), face and object recognition (Angelopoulos & Bates, 2022; Matiz & Barner, 2020), medicine (Vazquez & Facelli, 2022) as well as weather prediction, outlier detection and Natural Language Processing (Angelopoulos & Bates, 2022). Due to their simplicity, interpretability, and model agnosticism, conformal predictors built on established Machine Learning models align with the regulatory requirements imposed on financial institutions (Ahlberg et al., 2015).

Despite this, previous research on AL in Credit Scoring did not use Conformity (Carta et al., 2020; Martens et al., 2009; Saia & Carta, 2016; Zhao et al., 2008). In their most recent paper, Matiz and Barner (2020) test AL strategies based on conformal prediction. The datasets they use come from the field of face and object recognition. While these fields are also classification tasks, Credit Scoring is a specific field for class labeling of instances, as it allows one label per instance only rather than all possible combinations of labels. Therefore, binary classification of this kind requires distinct AL strategies.

Denis and Hebiri (2015) demonstrate a labeling approach based on conformal prediction similar to AL on a binary classification problem with simulated data. They do not select uncertain instances that are assumed to be informative in AL but label only certain ones. Different from an AL setting, they assume plentiful, unbiased data for a smaller number of predictions. In their setting, there is no data collection from prediction and no consequence from not labeling an instance. Also, no data is lost when predicting a particular class. In contrast, the context of Credit Scoring involves scarce data on defaulting and rejected customers, as financial institutions strive for a healthy loan repayment-to-default ratio and do not collect repayment behavior data from rejected applicants. Consequently, the primary focus of AL in Credit Scoring is the selection of instances for labeling rather than their exclusion.

These distinct characteristics make Credit Scoring a worthy domain for further investigation. AL is particularly well-suited for Credit Scoring due to the cost imbalance that often makes models overly conservative. In addition, AL enables the extension of loans to informative applicants whilst simultaneously reducing costs, as repaying customers are more prevalent in conventional retail lending. Conformal prediction intervals, with their flexibility to accommodate different strategies, serve as a suitable tool for AL in Credit Scoring.

AL becomes even more critical in the case of a so-called cold start problem (Donmez et al., 2007; Zhu et al., 2008). This arises when the data does not contain enough information for a Machine Learning model to learn the patterns of a domain (Fernández-Tobías et al., 2016; Lika et al., 2014; Son, 2016), which is particularly prevalent for biased data (Attenberg & Provost, 2011; Thanuja et al., 2011). Credit Scoring as a domain is prone to the cold start problem (Saia & Carta, 2016), especially when a financial institution seeks to enter a new market with limited reliable data.

Since Credit Scoring allows exactly one label per instance, it is a particularly interesting field for Conformity. Multiple labels are not possible because the label for loan repayment (GOOD) and defaulting (BAD) contradict each other. Empty labels are impossible because a financial institution choosing not to classify an applicant results in the same outcome as classifying the applicant to default on her loan. By employing AL strategies rather than traditional Passive Learning approaches, financial institutions can expand their customer portfolios and increase revenue in the short and long term. Ultimately, profitability is expected to improve over time.

This master thesis aims to assess AL strategies based on conformal prediction intervals for RI and the main impacting factors of the strategies' performance. To mitigate the bias from real datasets, both real and semi-artificial datasets are used. In doing so, it makes several contributions to existing literature. First, it introduces an AL technique within the context of

RI. Second, Conformity with its compliance to financial regulation is introduced into AL in the Credit Scoring context. Third, it tests AL strategies on Credit Scoring data that closely resembles the overall population of applications, including a portion generated through a novel simulation approach. Fourth, it proposes a cost-sensitive nonconformity function for designing conformal AL strategies. The remainder of this thesis is structured as follows. Section 2 provides a detailed explanation of the theoretical framework, encompassing RI, AL, and Conformity. Section 3 describes the experimental design developed to test AL strategies for RI using conformal prediction in the Credit Scoring context. Section 4 presents the empirical results, while Section 5 offers a discussion and explanation of these findings. Finally, Section 6 suggests future research areas, followed by a conclusion in Section 7.

2. Theoretical Framework

A number of authors provide definitions for Credit Scoring (see Abdou & Pointon, 2011 for an extensive overview). One definition determines Credit Scoring to be a statistical approach able to evaluate the probability that a new instance is considered GOOD or BAD by exploiting a model defined on the basis of previous instances (Carta et al., 2020; Henley, William Edward, 1995; Mester, 1997). The probability scores are compared to a predefined threshold and the instances are assigned to their corresponding classes accordingly (Hand, 2005).

The models used in Credit Scoring are called scorecards (Hand, 2005), which are typically trained on data from earlier customers who have been given a loan and whose repayment behavior has been observed. With no data on the repayment behavior of applicants who have not been given a loan in the first place, the data related to these is typically discarded entirely. It is therefore essential to distinguish between the credit-granting process, performed by the scorecard (accept or reject), and the observation of loan performance, which is based only on accepted applicants (Marshall et al., 2010). This results in a selection bias: a model is trained on a non-random sample of a population, while the same model is supposed to predict instances from the original population (Banasik et al., 2003). Addressing the problem of sample selection bias in Credit Scoring is the topic of RI (Crook & Banasik, 2004; Hand & Henley, 1993).

AL may help overcome the bias. In the context of Credit Scoring, this means that a bank gives loans to customers intending to generate data that improves their existing model. Therefore, AL is seen as a way of RI, with RI referring to both AL and other RI techniques. With the aim of generating valuable data, financial institutions can update their scorecards better for the changes of external factors that force them to regularly update them (Bart Baesens, 2003; Hand et al., 2001; Nikolaidis et al., 2017).

2.1 Reject Inference

RI has been present for a considerable period of time, and numerous approaches have been suggested. For a recent overview of RI techniques, refer to Anderson (2022), El Annas (2022), Ogundimu (2022), and Song et al. (2022). The primary assumption of RI is that a scorecard trained with both the accepted and rejected applicants would be superior as it is more representative of the new applicants the financial institution would assess for a loan (Anderson et al., 2022). RI techniques can be distinguished by relying on one of the two assumptions about the rejected applicants (Kim & Sohn, 2007). One assumes that $P(\text{default}|X, \text{rejected}) = P(\text{default}|X, \text{accepted})$, where X is the vector of applicants' attributes, while the other assumes that $\mathrm{P}(\text{default}|\mathrm{X}, \text{rejected}) \neq \mathrm{P}(\text{default}|\mathrm{X}, \text{accepted})$. As the first implies that the distribution pattern of accepted applicants can be extended to that of rejected ones, the probability of default (PD) of the overall population, $P(\text{default}|X)$, can be approximated by the conditional model based on $P(\text{default}|X, \text{accepted})$ for an applicant selected randomly from the whole population. This is not the case when $P(\text{default}|X, \text{rejected}) \neq P(\text{default}|X, \text{accepted})$, where other methods need to be conducted (Ogundimu, 2022).

Traditionally, RI techniques are of statistical nature. However, semi-supervised approaches relying on specific models have proven to be superior (El Annas et al., 2022). Apart from the statistical approach, other RI techniques do also not rely on a particular model. One suggests incorporating external factors such as expert knowledge for manual labeling (Kozodoi et al., 2019; Montrichard, 2007). Another technique, first suggested by Hand & Henley (1993) is to generate real data by handing out loans to rejected applicants. Optimizing this is done with AL.

2.2 Active Learning

In contrast to Passive Learning, which is typically the default learning strategy in any environment, AL strategies select the most informative instances for a Machine Learning algorithm to achieve the most efficient prediction accuracy (Fu et al., 2013; Settles, 2009). AL applications are typically those where labeling data is associated with costs (Matiz & Barner, 2020; Settles, 2009). Labeling refers to gathering all information for an instance and is typically done after a particular class is predicted. Owing to costs, AL selection of unlabeled data is assumed to be bound to a budget b. After selection, the instances are labeled, and then the model in question is (re-)trained with the new instances (Carcillo et al., 2018).

The application areas for AL can be classified into pool-based AL and stream-based AL (Carcillo et al., 2018; Settles, 2009). While the former assumes the data in every labeling set to be from the same distribution, the latter is less simple with issues like concept drift (Carcillo et al., 2018; Martins et al., 2023). There are multiple different query functions that are used in order to select AL instances (Matiz & Barner, 2020). Generally, they can be classified into those based on the instance’s uncertainty, diversity, or a combination of both (Fu et al., 2013; Matiz & Barner, 2020; Nguyen et al., 2022; Settles, 2009; Zhan et al., 2021). Uncertainty describes how confident a classifier is in its prediction, generally measured by the deviation of the predicted probability and its true class in classification settings. Uncertain instances are assumed to be more informative (Settles, 2009; Settles & Craven, 2008; Sharma & Bilgic, 2017). Diversity is a measure of similarity between instances, with more unusual instances assumed to be more informative than others.

Besides that, it is common practice to set up AL experiments in rounds to measure the learner's improvement through the rounds (e.g., Nguyen et al., 2022). The underlying benchmark that every AL strategy tries to beat is the random selection of instances for labeling (Carcillo et al., 2018; Cohn et al., 1994; Fu et al., 2013; Yang & Loog, 2016; Zhan et al., 2021).

2.3 Conformity

Conformal prediction is characterized by using past experience to determine precise confidence levels in predictions (Shafer & Vovk, 2008). Same as with prediction intervals (Geisser, 1993, pp. 6–15) based on the Neyman-Pearson Theory for hypothesis testing and confidence intervals (Lehmann & Romano, 2005), confidence is measured as $1 - \varepsilon$, where ε is the significance level specifying the limit below which a p-value must fall in order to reject the hypothesis of $\hat{y} \neq y$.

The eponymous nonconformity measure quantifies how unusual an instance looks compared to previous examples (Shafer & Vovk, 2008). The nonconformity measure is then turned to prediction regions by a conformal predictor. This procedure goes through the following steps (Johansson et al., 2017): First, a predictive model is trained on a set of data. Next, the nonconformity function measures the nonconformity of the instances of the relevant dataset and assigns a calculated calibration score for each instance. When an instance is to be predicted, its nonconformity score is compared with those of the calibration scores and the share of calibration scores that is larger or equal is then the p-value or the ε value that indicates the 1 – ε confidence in the prediction. As it is the nonconformity relative to the respective calibration set that matters, a monotonical change of the error function does not affect the results at all (Shafer & Vovk, 2008). For example, the results are always the same when using absolute or squared errors. In the same way, one could use a conformity measure instead of a nonconformity one. However, the latter is not a common choice (Fontana et al., 2020).

Conformal predictions are always *valid*, i.e., the true value lies in the prediction interval 1 – ε times (Fontana et al., 2020; Vovk et al., 2005, p. 9) if the data is exchangeable. Exchangeability is a weaker form of the frequent randomness requirement and assumes the data to be from the same distribution but not necessarily independent (Linusson et al., 2017).

With validity guaranteed under weak assumptions, conformal algorithms are tuned to be most *efficient* (Vovk et al., 2005), i.e., their prediction interval size is to be minimized. This depends on the nonconformity function and its ability to sufficiently rank instances on their strangeness (Linusson et al., 2017). As a standard method, the nonconformity function is based on a traditional Machine Learning model, usually the same that is used for prediction. However, there are also nonconformity functions dependent on other measures, such as the Euclidean distance of an instance to instances of the same or different class (Fontana et al., 2020).

Papadopoulos et al. (2002a; 2002b) were the first to introduce conformal prediction for regression and classification. They developed Inductive Conformal Predictors (ICPs) for the one-time prediction setting, also known as off-line learning. Vovk (2002) expanded this concept by introducing Transductive Conformal Classifiers (TCPs) for stream-based learning, which is called on-line learning in the Conformity context. In accordance with the off-line setting, ICPs are trained only once and perform only one prediction of all instances of the prediction set at once. This requires them to designate part of the training set as a calibration set to measure nonconformity and obtain the list of calibration scores (Johansson et al., 2017). This is not the case for TCPs, which can use the whole labeled data for training. However, after every prediction of an instance, the instance needs to be appended to the training set and the model retrained, making TCPs much more computationally demanding than the ICP. Despite having access to more training data, TCPs are not always more confident in their predictions (Linusson et al., 2014). Furthermore, overcoming the loss of training data for ICPs is possible by using the same principle from cross-validation (Papadopoulos, 2015; Vovk, 2012).

3. Experimental Design

The intended output of scorecards varies. By the Basel II Capital Accord, financial institutions are required to state the PD, the exposure at default, and the loss given default (LGD) (Lessmann et al., 2015). Most European Union member states are also already on their way to comply with the EBA's guidelines for stating own estimates of LGDs[3]. Nevertheless, almost all publicly available datasets do not contain much more information other than whether the loan has defaulted. Despite suggestions for more classes of outcomes, the classification into GOOD and BAD clients remains the standard in research. (Abdou & Pointon, 2011). For this reason, the predictive task of the experiment is a binary classification problem that aligns with most of the research.

As part of a larger sum of different econometric modeling techniques, Logistic Regression (LR) was one of the earliest models that were first used for the new Credit Card business for banks in the 1960s and gradually rose in popularity (Dumitrescu et al., 2021). As a linear model, it does not capture non-linear effects, though feature engineering methods that introduce non-linear effects are common in the industry (Hurlin & Pérignon, 2020). Despite the advancement of many Machine Learning techniques, LR remains the industry standard for Credit Scoring (Abdou & Pointon, 2011; Anderson et al., 2022; Hand, 2005). It is the

[3] See the compliance table of the guidelines on Credit Risk Mitigation by the EBA: https://www.eba.europa.eu/regulation-and-policy/model-validation/guidelines-on-credit-risk-mitigation-for-institutions-applying-the-irb-approach-with-own-estimates-of-lgds

result of its simplicity and ease of interpretability that makes it the standard choice for financial institutions (Dastile et al., 2020). Regulatory frameworks like the ones from the European Banking Authority[4] make it also a favorable choice among numerous other frameworks (Dumitrescu et al., 2021).

Class imbalance is one of the biggest performance factors in classification problems (Brown & Mues, 2012; Chawla et al., 2004; Haibo He & Garcia, 2009; Japkowicz & Stephen, 2002). It is an omnipresent issue in Credit Scoring due to potential selection bias. For heavily imbalanced datasets, there is a risk of training a model to be a naïve classifier, i.e., a classifier that predicts only the majority class. To omit this risk for the partly heavily imbalanced datasets in this experiment, every LR-based learner is given the class weight parameter '*balanced'*, providing every class with a weight inversely proportional to its frequency in the training data.

The experiment is designed as a pool-based AL experiment with several rounds, or in different words, a semi-off-line setting (Linusson, 2017). The reason is that it is more common in Credit Scoring than stream-based learning and is much less computationally expensive for an experiment with the data at hand. An LR-based learner is trained with the training set. Then, a round dataset is predicted. Next, the prediction is evaluated, and the goods are stored in the training data, as is the case for real-world financial institutions that hand out loans and collect customer data. This is done for every round of the split test set, strategy, and dataset. The results are tables for each strategy and each dataset that document the performance for every round in every metric. A pseudo-code for the experiment is presented in the Appendix.

The detailed procedures for Feature Engineering and the strategies used are described in detail in the following sections. After preprocessing, each dataset is split into a 10 % training set and a remaining test set. Half of all test sets has been enhanced to resemble the general population of loan customers better, the other half has been left untreated. The test sets are then split into nine subsets that resemble AL rounds. The training set was not enhanced to ensure a real-world setting where RI causes a cold start problem with training data not resembling a distribution similar enough to the test data. Each model represented by a learning strategy is then trained on the starting data. Next, it predicts the unlabeled instances

[4] See the Guidelines on Credit Risk Mitigation by the EBA: https://www.eba.europa.eu/regulation-and-policy/model-validation/guidelines-on-credit-risk-mitigation-for-institutions-applying-the-irb-approach-with-own-estimates-of-lgds

in the round dataset and appends the applications to its training dataset according to the respective strategy. Then, the model is retrained with the larger training set and predicts the next round of unlabeled instances until all nine rounds are finished. The code for the experiment as well as the electronic Appendix is uploaded on GitHub[5].

3.1 Datasets

Table 3.1 summarizes the parameters for every dataset used in this experiment. These datasets are based on five datasets that are publicly available online. As it is recommended not to compare retail loans with corporate ones (Lessmann et al., 2015), all five original datasets resemble retail loan data or were stripped of their corporate loan data. They differ in size, number of features, default rate, and the dataset-specific threshold that was estimated cost-efficiently. The thresholds the imbalance of costs (see Section 3.4). The ones presented in the table are those calculated for the subset of training data that every strategy fully uses in round one. They are the only thresholds that are the same for all strategies, as with every round the acquired training data differs among strategies. Nevertheless, the difference from round to round is relatively small. Interested readers can find it in the electronic Appendix. They were also selected for their features that allow for determining the expected revenue of each loan in most detail, such as loan size, duration, interest rate, and installments.

The selection bias comes from scorecards being trained on biased data but being used on unbiased data. RI researchers therefore use unbiased data for their experiments, however they do not make their data publicly available most of the time, or it is not representative (e.g., Banasik et al., 2003; Barakova et al., 2011; Kozodoi et al., 2019; Verstraeten & Van den Poel, 2005). Anderson et al. (2022) use data from the Peer-to-Peer (P2P) lending company Lending Club, which distinguishes itself by recording data on accepted as well as rejected clients. This dataset, hereby called *LC*, contains all loan data of Lending Club from 2007 till 2018, when they were making all their data open to the public. P2P datasets are marked by an extremely high rejection ratio compared to other Credit Scoring datasets (El Annas et al., 2022), which is another reason why this dataset deserves a category for itself.

[5] Available at https://github.com/MaximilianSuliga/Conformal-Active-Learning-for-Reject-Inference

Type	**Name**[6]	**Cases**	**Features**[7]	**Default Rate**	**Starting Threshold**
Original Data	Small	614	12	0.3127	0.3679
	German	1,000	21	0.3	0.1544
	LC	31,378[8]	6	0.1962	0.2191
	Deloitte	67,463	32	0.0925	0.2673
	Large	127,908	32	0.2304	0.3485
Accepted & Rejected Instances	LC_all	376,757	6	0.9331	0.2191
Artificially Enhanced Data	Small_MC	752	12	0.4389	0.3679
	German_MC	1,225	21	0.4286	0.1544
	Deloitte_MC	82,642	32	0.2592	0.2673
	Large_MC	156,687	32	0.3484	0.3485

Table 3.1 Data Summary

The raw data from the Lending Club, encompassing over 2 million instances of accepted loans, is unusually large. Due to the oddness of this large size, the discrepancy between the second smallest dataset *German*, and the second largest original dataset *Deloitte*, frequent missing values, and limited computational resources, a subsample was taken to create an intermediate-sized dataset. To merge the data of the accepted clients with the rejected ones into a representative dataset, all features that were only available in one dataset were discarded. Consequently, the feature indicating the defaulting status was supposed to be discarded since no information is available on whether a customer defaulted on a loan when they were not initially granted one. Therefore, an assumption is made that all rejected customers would have defaulted if given a loan (Siddiqi, 2012), resulting in a high default rate of 93.31 %.

As it is common, the remaining original datasets do not provide any information on rejected clients. To enhance their representativeness of the general population of credit applications, they were extended with simulated denied clients as part of the preprocessing process.

[6] Links to the source are embedded in the respective dataset name
[7] Number of features after preprocessing
[8] Down-sized sample of the original data

Similar to *LC*, it is assumed that rejected customers would all default. The artificial defaulting customers are created with a Monte Carlo (MC) simulation based on the defaulting clients of the original datasets. It is essential to highlight that while only the target feature for LC is assumed for real rejected customers, the remaining datasets are extended with fully artificial customers based on accepted defaulted customers. Therefore, these datasets can be regarded as semi-artificial since they comprise both real and artificial instances. In essence, the original datasets represent the True Negatives (TN) and False Negatives (FN). By assuming that all rejected clients would have defaulted, the datasets are extended with TN instances. Since there is no information available regarding the defaulting behavior of customers who were not granted a loan, it is impossible to determine, without speculation, which rejected customers would have defaulted and which would not have defaulted in the absence of further information.

3.2 Preprocessing

Before the experiment, every dataset is preprocessed to meet the criteria required to be used in the experiment. This includes deleting uninformative features such as the ID of a loan, deleting non-retail loans, deleting features with more than 30 % of missing values, deleting duplicates, filling remaining missing values with the mean for numerical and the mode for categorical features as well as deleting features with only one unique value. No further procedures were conducted to ensure the largest possible size and dimensionality for each dataset in line with the recommendations by Lessmann et al. (2015). Following these authors, too much alteration was avoided to retain their natural characteristics and preserve their relative performance.

For *LC* and *LC_all*, preprocessing included deleting all features unavailable in both the accepted and the rejected application datasets. With no information available for the remaining datasets, their rejected customers had to be simulated. This process aimed to answer two key questions: the number of rejected customers and their corresponding characteristics. A frequently cited survey by Bankrate indicates that in 2020, 21% of American retail customers were denied a loan[9]. According to the New York Fed data, 18% of

[9] For more information on the survey, see https://www.bankrate.com/finance/credit-cards/credit-denial-survey/#methodology

retail customers were rejected for their loan applications[10]. The rates of both sources differ when examining the exact purpose of a loan. However, the number of different purposes is very limited. Furthermore, both rates are based on American customers. Still, only dataset *German* gives information on which country the data originates from. To simplify the analysis and maintain a pragmatic approach, each dataset was uniformly enhanced, approximating a 20% rejection rate. This approach avoids unnecessary complexity and ensures consistency across chosen datasets.

The simulations need to be based on the real dataset to have the simulated rejections resemble those of their respective dataset (precisely the default accepted customers). This is achieved by an MC Simulation that generates customers that resemble randomly chosen customers of a population with the same parameters as the underlying dataset. A detailed description of the procedure can be found in the appendix.

There are also other methods that can alter the balance of a dataset. The most common approaches are under- and over-sampling (Carta et al., 2020). Under-sampling removes the instances of the majority class that are more than the minority classes. Over-sampling, on the other hand, typically uses exact copies of the minority class instances, which includes the common SMOTE approach (Marqués et al., 2013). It has been shown that over-sampling is generally superior to under-sampling, especially when using LR (Crone & Finlay, 2012; Marqués et al., 2013). Additionally, the eventual data loss should not be underestimated when using under-sampling, especially in the case of heavily imbalanced datasets. However, it was also found that duplicates in training datasets are useless for the performance of Machine Learning models (Allamanis, 2019) or improve them marginally at best (Zhao et al., 2021) when used on imbalanced prediction sets.

Regarding the enhancement of datasets, it is unrealistic in the context of Credit Scoring that instances are entirely identical. Enhancing datasets with non-duplicates of the minority class instances is expected to yield the best and most representative results, similar to other more sophisticated cloning techniques (Jiang et al., 2015). MC simulation is a way of simulating specific data with particular characteristics (Dumitrescu et al., 2021).

[10] See https://www.newyorkfed.org/microeconomics/databank.html

3.3 Feature Engineering

Feature Engineering and Feature Selection are standard procedures to improve model performance. However, they are also time-consuming, computationally more expensive, very dataset specific, and may require human judgment. As a result, no additional feature engineering procedures are conducted on the 540 AL rounds besides the standard scaling and Weight of Evidence (WoE) transformation procedures. LR allows for Lasso regularization (Friedman et al., 2010), which can be used as a way of feature selection in that the betas of performance-reducing features are set to 0. The procedure is automatic, feature- and thus, also dataset-agnostic, and computationally inexpensive.

Features of numerical type are standard scaled, as it is recommended for LR with Lasso Regularization (Hastie et al., 2009, p. 125). Since LR takes only numerical features as input, categorical features need to be transformed into numeric ones to use them. The common approach in Credit Scoring is WoE transformation (Siddiqi, 2012, p. 81). Unlike the simpler use of dummy variables, the categorical feature with k categories is replaced by one numeric feature rather than k or $k-1$ dummies. This may reduce dimensionality drastically, save more information per feature and therefore, serve as a way of protection from the curse of dimensionality (Guyon & Elisseeff, 2003). Additionally, WoE transformation offers the possibility of feature assessment for feature selection. Nevertheless, due to its reliance on heuristic evaluations, it is discarded, leaving feature selection to rely solely on Lasso Regularization.

3.4 Cost-efficient learning

Apart from class imbalance, Credit Scoring is also marked by different costs that arise when an erroneous decision is made by a scorecard. While there is an obvious correlation between statistical and economic performance, the statistically best scorecard does not necessarily need to be the best in economic terms (Lessmann et al., 2015). As financial institutions use scorecards that aim to maximize their profits, cost efficiency is prioritized above statistical accuracy. Moreover, since Machine Learning models are designed for statistical accuracy by default, they need to be cost-sensitive to aim at economic performance.

Multiple ways exist of making Machine Learning models cost-sensitive (Viaene & Dedene, 2005). As a first step, the costs that arise need to be defined. Dumitrescu et al. (2021) identify two different approaches. The first one gives each error scenario fixed predefined costs

(Akkoç, 2012), whereas the second tests different possible scenarios (Lessmann et al., 2015). For a more convenient comparison of the empirical results of the experiment and the simplification of the cost sensitization process of the model, the first approach was chosen for this experiment.

There are two possible errors in a binary classification problem.: False Positive (FP) and FN which are accompanied by their correct classifications, True Positive (TP) and TN. Though a fixed amount for each type of error is commonly accepted in academic literature (Alejo et al., 2013; Beling et al., 2005; Oliver & Thomas, 2009; Verbraken et al., 2014), Bahnsen et al. (2014) argue that this is unrealistic in the Credit Scoring setting. Instead, the cost of an erroneous decision should depend on the loan size, terms, the interest rate, and other relevant factors that vary for each granted loan. The costs for each error scenario rely on each application's independent features and are supposed to resemble the revenue that would have been generated if the loan was given to a GOOD customer and the cost of capital. Costs from an FP (CFP) are therefore the cost of opportunity (B. Baesens et al., 2003; Nayak & Turvey, 1997), and costs from an FN (CFN) the cost of default (Beling et al., 2005; Oliver & Thomas, 2009; Verbraken et al., 2014). No costs are assumed for correct classifications.

One might argue that a defaulting customer is also making the financial institution lose the revenue the loan would have generated. However, it is crucial to consider that the cost of opportunity is applicable in only one error scenario. In practice, it boils down to whether more customers or loans are available. If customers are abundant, rejecting a creditworthy customer does not result in a loss, as the loan can be offered to another customer. In other words, the loan will not remain without a creditor. However, if it is given to a defaulting customer, the anticipated revenue is lost.

On the contrary, if applicants are scarce, rejecting applicants risks the scenario of “dead capital”, which is associated with opportunity costs. In this scenario, the defaulting customer does not generate opportunity costs since rejecting them would not have resulted in any revenue, given the unavailability of alternative customers to provide loans to. For this experiment, opportunity costs were assigned to FP and LGD to FN, as an FP of 0 and FN of opportunity costs plus LGD would enable a naïve classifier to achieve the best economic performance by predicting 0 for all instances, since this would omit the possibility of defaults and the associated opportunity costs.

Each dataset is given cost functions that were individually specified according to the dataset that it was based on (see Table 3.2). While a defaulted loan's recovery varies in practice (Bahnsen et al., 2014; Petrides et al., 2020), a fixed LGD was chosen for this experiment. The reason is that information on LGD is very scarce among Credit Scoring datasets and is not available for any of the datasets selected for this experiment. To ensure the generally agreed rule of CFN > CFP in Credit Scoring (Dumitrescu et al., 2021), LGD was set at 100 %, making CFN the total size of the given loan for every dataset[11]. Despite this and trying to define the revenue for each original dataset as reasonably as possible, some datasets made the CFN require including the CFP to guarantee the general assumption of CFN > CFP. Therefore, the results presented are therefore computed with one too many CFP. Results generated with a CFN = LGD only can be found in the electronic Appendix.

Dataset	**Cost Functions**

Small	CFP = loan amount * term * 0.05[12]
	CFN = loan amount + CFP

German	CFP = term / 24[13]
	CFN = 5[14]

LC	CFP = loan amount * 3[15] *0.13[16]
	CFN = loan amount+ CFP

Deloitte	CFP = loan amount * term * interest rate
	CFN = loan amount+ CFP

Large	CFP = loan amount * term * interest rate + upfront charge
	CFN = loan amount+ CFP

Table 3.2 Dataset Specific Cost Equations

Thresholding is a simple and model-agnostic way of making binary classifiers cost-sensitive (Sheng & Ling, 2006). The threshold τ that separates predicted probabilities into those

[11] *German* does not have the loan amount specified for every application. However, it comes with a cost matrix specifying CFN = 1 CFP = 5. CFN was approximated to be on average around 1 and CFP was fixed at 5.
[12] Interest rate assumed to be 5 %
[13] Term adjusted to have a mean close to 1 to match given cost matrix
[14] CFN given in cost matrix from source
[15] Most frequent term for accepted customers
[16] Average interest rate for accepted customers

rounded to one or the other class is fully determined by the cost matrix of the training set (see Equation 1). In the case of case-specific costs, the average cost for each error is used.

$$\tau = \frac{\overline{\mathrm{CFP}}}{\overline{\mathrm{CFP}}+\overline{\mathrm{CFN}}} \tag{1}$$

This ensures the same threshold for every strategy in the starting round given a specific dataset, as the training data is the same. With every round, every strategy collects more data to retrain its model, which is expected to be biased, and, hence, the threshold is expected to change in every round for every strategy individually for each dataset.

3.5 Strategies

A total of seven strategies were tested for the experiment summarized in Table 3.3. The strategies. *No AL* represents conventional passive learning with no RI technique applied. *Benchmark RI* represents the benchmark for general RI. A random sample of predicted defaults of size *b* is given the label BAD and appended to the training data together with goods (Siddiqi, 2012). The actual outcome, known in this experiment but unknown in a real setting, is discarded. The selected instances are not incorporated when evaluating the performance of the respective round. *Random AL* marks the benchmark for AL techniques. A random sample of bads of size *b* is reclassified to goods and appended to the other goods. In practice, this means that a random sample of usually rejected customers is given a loan. Different from the sole classification of already rejected applications, any AL relabeling influences a scorecard's economic performance in a round. Thus, both accepted and AL applications are considered when evaluating the scorecard’s costs in a round. The conformal classifiers were built with the nonconformist Python library[17] by Linusson (2017). A total of four conformal classifiers were chosen with different nonconformity functions. The nonconformity functions Probability Error and NN Margin Error aim to represent uncertainty and a combination of uncertainty and diversity and were both modified into a cost-sensitive version each. The error functions are assessed in detail in the sub-sequent sections.

[17] https://github.com/donlnz/nonconformist

Name	Description
No AL	No Reject Inference Technique
Benchmark RI	Classify a random sample of bads as BAD and append them to the training data
Random AL	Select a random sample of bads and reclassify them to good
ICP Prob	Select the most nonconforming bads according to Probability Error and relabel them to good
ICP Prob Cost	Select the most nonconforming bads according to cost-sensitive Probability Error and relabel them to good
ICP NN Margin	Select the most nonconforming bads according to NN Margin Error and relabel them to good
ICP NN Margin Cost	Select the most nonconforming bads according to cost-sensitive NN Margin Error and relabel them to good

Table 3.3 Strategies Summary

In line with the recommendation of Linusson et al. (2014), a calibration set size of 20 % of the training set was chosen. For some datasets, especially for the smaller ones, this violates the recommendations of using a calibration set of around 500 to 1000 instances (Angelopoulos & Bates, 2022; Linusson et al., 2014). A variable relative size could omit this for smaller datasets or a fixed absolute size for larger datasets. However, to ensure better comparison, the fixed relative size was chosen. For every bad, the nonconformity score is calculated, and the top b are classified as GOOD in contradiction to their PD. The nonconformity score represents the confidence in the PD. An application with a high nonconformity score means low confidence in the PD and vice versa. The idea is to grant some applicants a loan whose prediction score is too low for getting a loan, but their PD is unconfident, when at the same time rejecting those applicants, whose prediction is confident and too low for getting a loan.

While passive RI techniques may utilize the entirety of rejected loan applications, AL techniques face economic constraints related to the costs associated with loan defaults. To select rejected applications for AL, one can either classify them into *informative* and *uninformative* or give them a continuous score, rank them, and select the top b. Previous literature on AL in the Credit Scoring context has focused on the first approach (Carta et al., 2020; Saia & Carta, 2016; Zhao et al., 2008). By specifying an error term, nonconformal scores can be cut off and classified into confident and unconfident. However, this does not work for the benchmarking RI techniques that rely on specified b number of instances. For this reason, b was chosen as a fixed share of all acceptances.

The determination of b is based on the cost matrix approach, where the CFN typically is a share of CFP (Dumitrescu et al., 2021) and resembles the AL setting budget. Assuming a perfect scorecard whose accepted applicants all turn out to be GOODs and its selected AL instances in turn all turn out to be BADs, a selection of AL customers of the size of $\frac{\overline{\mathrm{CFP}}}{\overline{\mathrm{CFP}}+\overline{\mathrm{CFN}}}$ of accepted applications should guarantee to offset the costs of AL customers by the good customers whilst generating training data more representative of the overall credit applicant population. This relies on data that is available before a prediction and hence is expected to differ more between strategies with every round. The same applies to the alternative of using the class imbalance. For this reason, the approach of using the cost matrix of each dataset available for each strategy was discarded despite achieving better performances on average. Interested readers can find the detailed results in the electronic appendix.

Instead, b was assumed as a fixed portion of acceptances for every dataset and every RI strategy to guarantee comparability. Remaining with the idea of using a cost matrix, the Credit Scoring literature assumes $CFP = 1$ and $CFN = 5$ (Abdou & Pointon, 2011; West, 2000), which is taken from *German*. Following the reliance on it, $b_{static\ share}$ was set to:

$$b_{static\ share} = \frac{CFP}{CFN} * n_{accepted} = 0.2 * n_{accepted} \tag{2}$$

with $n_{accepted}$ resembling the number of applications that are classified as repaying. This makes use of the simplifying assumption that every loan is of the same size in every application.

Benchmark RI classifies only $b_{static\ share}$ rejections to compare all strategies better with each other. However, one could also assess the performance of classifying all rejections as BAD for all RI strategies. In that setting, every strategy would predict applications as good or bad, eventually label a share of bads as part of an AL strategy, classify the remaining rejections as BAD, and append the full prediction set to the training set. The performance of all RI strategies for this approach can be assessed in the electronic Appendix.

3.5.1 Probability Error Function

The ICPs with a Probability Error function, also known under hinge loss (Johansson et al., 2017), represent the AL strategies relying on uncertainty. They use the general idea of using the error as a nonconformity measure (Angelopoulos & Bates, 2022; Nouretdinov et al., 2001; Shafer & Vovk, 2008). According to Johansson et al. (2017) the general Probability Error function is defined as:

$$\triangle\, [h(x_i), y] = 1 - \hat{P}(y_i|x_i) \tag{3}$$

The Probability Error function was modified to be cost-sensitive for another strategy. The idea is to penalize FN and FP differently by CFN and CFP. This modified Probability Error function is defined as:

$$\triangle\, [h(x_i), y] = \left(1 - \hat{P}(y_i|x_i)\right) * (1 - y_i) * \tau + \left(1 - \hat{P}(y_i|x_i)\right) * y_i * (1 - \tau). \tag{4}$$

The insecurity of prediction is penalized with the respective cost that would arise in the case of an erroneous prediction. Compared to the default Probability Error function, this skews the distribution of the calibration set in that BADs are higher in the ranking of insecurity towards prediction than GOODs, given $\tau < 0.5$. The model is supposedly more uncertain about goods that might result in an FN with a higher CFN than bads that might result in an FP with a lower CFP, respectively.

3.5.2 Nearest Neighbor Margin Error Function

The NN Margin function combines the popular Margin Error function and the concept of Nearest Neighbor (NN). According to Johansson et al. (2017) the general Margin Error function is defined as follows:

$$\triangle\, [h(x_i), y_i] = \max_{y \neq y_i} \widehat{P_h}\,(y|x_i) - \widehat{P_h}(y_i|x_i). \tag{5}$$

An instance is therefore nonconforming if it has a low predicted probability for the true class label and/or a high predicted probability for any other eventual multiple (incorrect) class labels. As the predicted class probabilities of the same instance are compared to each other, Margin Error functions still fall under the category of uncertainty-based AL strategies, where the relative uncertainty is being assessed rather than the absolute uncertainty as in the Probability Error function. Yet, it always yields the exact same results as the Probability Error function when used in a binary classification setting (Johansson et al., 2017). For this reason, Margin Error was modified to incorporate the idea of NN.

The use of Euclidean distance belongs to the group of alternative nonconformity measures that, unlike those provided by the nonconformist library (Probability Error and Margin Error), are not based on the prediction error (Fontana et al., 2020). The underlying idea is to compare an instance with its closest neighbors of a specified class in the spirit of the k-NN algorithm. For example, an instance surrounded by neighbors of a different class can be considered strange, whereas an instance surrounded by neighbors of the same class is rather ordinary (Nouretdinov et al., 2020). As an alternative to the commonly used Euclidean distance, one can also take the average of independent features and compare these values (Fontana et al., 2020). Since the nonconformist Python library requires the predicted probabilities and the true outcomes as inputs for nonconformity functions, the predicted probability was taken as a proxy for the measure of *look*. From an AL perspective, this is a strategy based on a conformal predictor with an error function based on a combination of uncertainty and diversity. This modified nonconformity function is defined as:

$$\triangle\,[h(x_i), y_i] = \frac{\min_{j \neq i} \left| \hat{P}(y_i \mid x_i) - \hat{P}\left(1 - y_i \mid x_j\right) \right|}{\min_{j \neq i} \left| \hat{P}(y_i \mid x_i) - \hat{P}\left(y_i \mid x_j\right) \right|}. \tag{6}$$

In the same fashion as the cost-sensitive Probability Error function, the cost-sensitive NN Margin Error function is defined as:

$$\begin{aligned} \triangle\,[h(x_i), y_i] = {} & \frac{\min_{j \neq i} \left| \hat{P}(y_i \mid x_i) - \hat{P}\left(1 - y_i \mid x_j\right) \right|}{\min_{j \neq i} \left| \hat{P}(y_i \mid x_i) - \hat{P}\left(y_i \mid x_j\right) \right|} * (1 - y_i) * \tau \\ & + \frac{\min_{j \neq i} \left| \hat{P}(y_i \mid x_i) - \hat{P}\left(1 - y_i \mid x_j\right) \right|}{\min_{j \neq i} \left| \hat{P}(y_i \mid x_i) - \hat{P}\left(y_i \mid x_j\right) \right|} * y_i * (1 - \tau). \end{aligned} \tag{7}$$

3.6 Performance Metrics

A variety of performance metrics is available for classification tasks. Lessmann et al. (2015) have identified three performance metrics that should be used in a scorecard assessment: Area under receiver operating characteristics curve (AUC), Partial Gini Index (PG), and Brier Score (BS). The PG is the Gini Index focused on a subset of predicted probabilities

$\hat{P}(+|x) > s$ (Pundir & Seshadri, 2012). Schechtman and Schechtman (2016) have proven that there is a linear relationship between AUC and the Gini coeffienct with $Gini = 2 * AUC - 1$. PG can therefore be replaced by an alternative performance metric, the Partial AUC (PAUC). The predicted probabilities were subset by setting $s = 0.5$. This was chosen to ensure a focus on accepted credit applicants who, by construction of the cost-efficient threshold and the higher FN costs than FP costs, are all available in this sample.

AUC and PAUC are performance indicators that can be seen as more advanced forms of the performance indicator percentage correctly classified (PCC). PCC is the fraction of correctly classified observations and only considers the discrete labels obtained when rounding the predicted probabilities. Although thresholds that contain class imbalance or cost-sensitive tuning may mitigate the effect to some extent (Lessmann et al., 2015), it is still commonly seen as prone to favor naïve classifiers that assign the majority class to every instance. AUC and PAUC omit this drawback by using observed probabilities rather than rounded classes and compare the TP rate against the FP rate across all possible thresholds. This way, class imbalance and therefore, favoring naïve classification is omitted (Fawcett, 2006). However, it does not incorporate the different costs that arise for each type of error.

BS is a classification performance metric that measures the squared difference between the predicted probability and its true class. It is the classification equivalent of the mean squared error in regression. As an error function, it measures the accuracy of probability predictions, with smaller values indicating better results. It is not entirely robust against class imbalance (Wallace & Dahabreh, 2014) and does not consider error costs if they are incorporated after predicting probabilities, as done with thresholding.

Statistical performance metrics allow for a general comparability of models between different domains. As scorecards are used primarily by financial institutions aiming to maximize profits, statistical measures can be seen as proxies for economic performance. Hence, a measure called *"Cost"* is incorporated to measure the economic costs of misclassification directly. It can also be seen as a weighted PCC that differentiates between error cases. Generally, economic costs can be measured in three ways: Absolute currency values, relative deviation from the total that can be incurred (e.g., Bahnsen et al., 2014), or relative deviation from a benchmark (e.g., Lessmann et al., 2015). Absolute currency values limit the comparison to models used on the same dataset. Therefore, it is unsuitable for general comparison and relative deviation should be favored. For this experiment, the benchmark comparison was chosen because it allows a clearer readability of the performance differences of each strategy.

4. Empirical Results

The empirical results from each dataset are organized into four distinct groups, each presented in its own section. Within each group, the strategies are evaluated based on their average performance across all rounds. This is followed by the development of all strategies throughout the rounds by each performance metric.

The strategies are to be compared with each other in the following way. No AL marks the overall benchmark, whose Cost score has been used as the basis to compare the costs of all other strategies with. Assuming RI is not used a lot in practice, it also represents the industry's status quo. Benchmark RI represents the most simple and general way of RI, against which other RI strategies are to be evaluated. Comparing it to No AL, one can assess whether RI is generally suitable. By comparing it with other strategies, it can be evaluated whether the different strategies can be classified as competitive RI strategies. Random AL represents the AL benchmark. Whereas the comparison with No AL may incorporate the specific characteristics of each dataset and whether AL is generally applicable, the comparison of Random AL with other AL strategies provides a benchmark that ignores the overall applicability of AL. In addition, each conformal AL strategy may be compared to the other AL strategies, particularly their cost-efficient version.

The statistical performance metrics have fixed ranges and hence can be compared to each other with their absolute values. Due to the fixed upper and lower boundaries, it can also be solely assessed to see whether it is performing well or bad. This is not the case for the economic performance metric *Cost*, which varies in scale for each dataset. As a result, dataset-specific fluctuations are smoothed out and it cannot be said whether a strategy generally improves its costs over the rounds. It can only be assessed whether it performs better or worse than the *No AL* benchmark throughout the rounds.

Although a $b_{dynamic\ share}$ in accordance with the threshold yielded better performance, the presented results utilized a $b_{static\ share}$ of 20% of accepted applicants. This decision was made because the performance of the strategies becomes less apparent when the sample size changes for each round, strategy, and dataset. Furthermore, the relative performance of the strategies remains almost identical when using either a dynamic threshold-sized or fixed AL size. The same observation applies to target imbalance as an AL sample size but with inferior performance compared to the fixed AL size. The results for both dynamic AL sample size versions can be found in the electronic Appendix.

A detailed look into the results of each individual dataset reveals that conformal predictors are generally better performing when there is more data available, which can be easily explained by their need for a calibration set. The evolution of thresholds over the rounds cannot be used as an indicator of the performance of a strategy. The only thing that points out are outliers among the threshold developments over the rounds for each strategy examined per dataset, where a deviation from the general threshold development indicates either exceptionally bad performance (in most cases by *ICP NN Margin*) or exceptionally good performance (by *ICP Prob Cost*). All performance scores and thresholds for every individual dataset are reported in the electronic Appendix.

4.1 All Datasets

In this group, the performance for all datasets has been averaged to get the most dataset-agnostic comparison of all strategies. The exact scores can be found in Table 4.1.

Strategies	**AUC**	**PAUC**	**BS**	**Cost**
No AL	0.666696	0.578594	0.232348	0
Benchmark RI	0.665266	0.580790	0.238759	-0.006230
Random AL	0.671139	0.577311	0.226071	0.052685
ICP Prob	0.691305	0.562067	0.276309	-0.057751
ICP Prob Cost	0.709833	0.629145	0.179892	0.065709
ICP NN Margin	0.362899	0.488751	0.329716	0.475513
ICP NN Margin Cost	0.623844	0.553088	0.220079	0.025694

Table 4.1 Performance Scores for All Datasets Averaged

Figures 4.1 to 4.4 depict each performance metric progression by round. The AUC scores for most strategies in Figure 4.1 are partially concave. From the second round, *ICP Prob* and *ICP Prob Cost* start outperforming their benchmarks regarding AUC. In comparing AUC per round between individual datasets, it can be confirmed that this comes from the requirement of a calibration set for conformal predictors. The calibration set shrinks the training set compared to *No AL* or *Random AL* (see electronic Appendix). This does not seem relevant regarding PAUC, where *ICP Prob Cost* outperforms all other strategies from round one (see Figures 4.1, 4.2). Like AUC in Figure 4.1, Figure 4.3 shows a BS for *ICP Prob Cost* above its benchmarks for round one that sharply drops in round two and stays below its benchmarks till the last round. However, this is not the case for *ICP Prob*, which scores higher in round one and deviates more and more with every round. From round two, there seems to be a general upward trend for all three benchmarks. Whereas other AL strategies start having a negative slope from one of the beginning rounds (including *ICP NN Margin*), *ICP Prob Cost's* slope is steeper than the one of all benchmarks. Figure 4.4 depicts the relative deviation of costs to the *No AL* benchmark. As a result, the *No AL* curve is constantly set at 0.

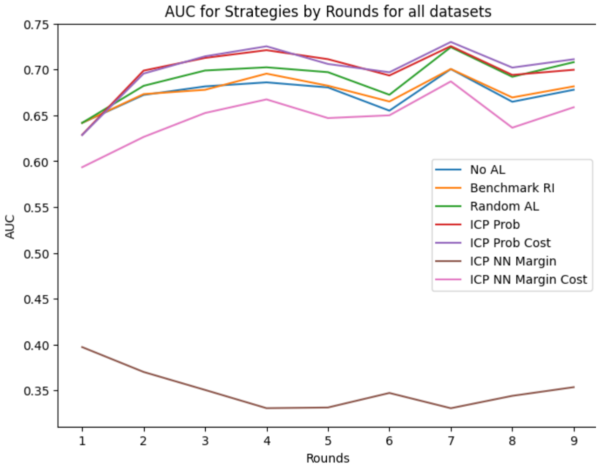


Figure 4.1 AUC by Rounds for All Datasets Averaged

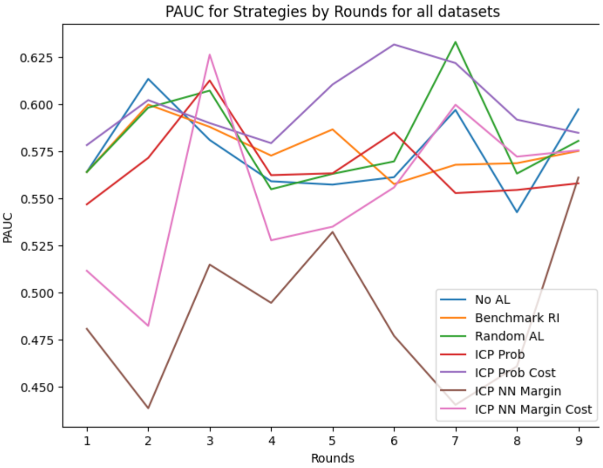


Figure 4.2 PAUC by Rounds for All Datasets Averaged

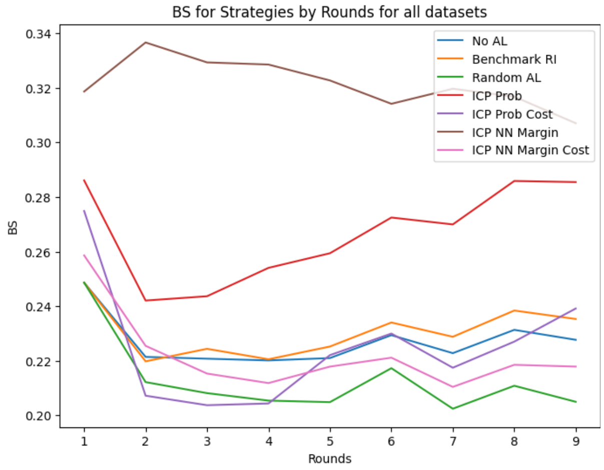


Figure 4.3 BS by Rounds for all Datasets Averaged

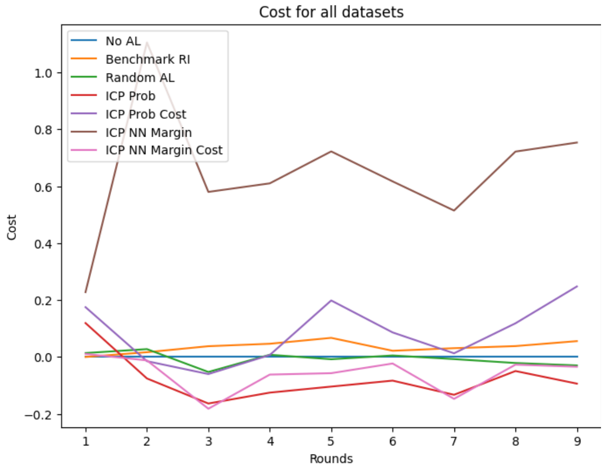


Figure 4.4 Cost by Rounds for all Datasets Averaged

4.2 Original Datasets

This group consists of all datasets that have not been altered (i.e., Small, German, LC, Deloitte, Large). The exact scores can be found in Table 4.2. Apart from PAUC, the performance metrics indicate that AL seems useful if evaluated by widely available data. The progression of each performance metric for the original datasets is shown in Figures 4.5 to 4.8. Though resembling the graphs in Figures 4.1 to 4.4, they differ in their dispersion.

Strategies	AUC	PAUC	BS	Cost
No AL	0.656968	0.565756	0.248733	0
Benchmark RI	0.649751	0.572214	0.260050	-0.019684
Random AL	0.658213	0.566770	0.240536	0.008235
ICP Prob	0.665911	0.550830	0.299510	-0.120530
ICP Prob Cost	0.679802	0.589869	0.160994	-0.177787
ICP NN Margin	0.383401	0.466797	0.327348	0.226263
ICP NN Margin Cost	0.611621	0.530536	0.215245	0.065218

Table 4.2 Performance Scores for Original Datasets Averaged

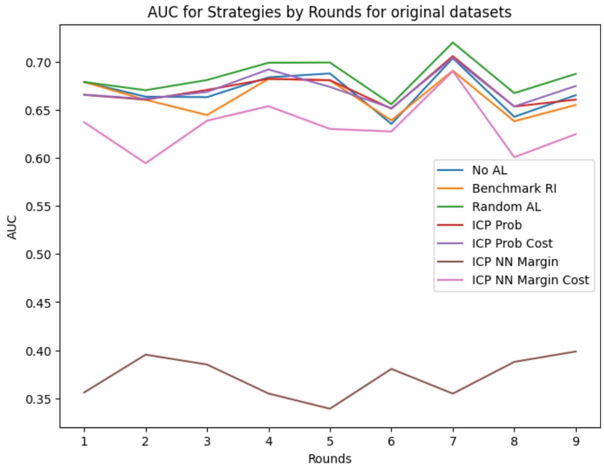


Figure 4.5 AUC by Rounds for Original Datasets Averaged

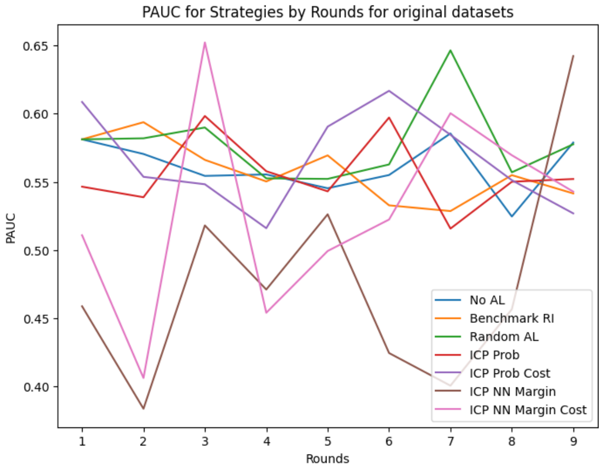


Figure 4.6 PAUC by Rounds for Original Datasets Averaged

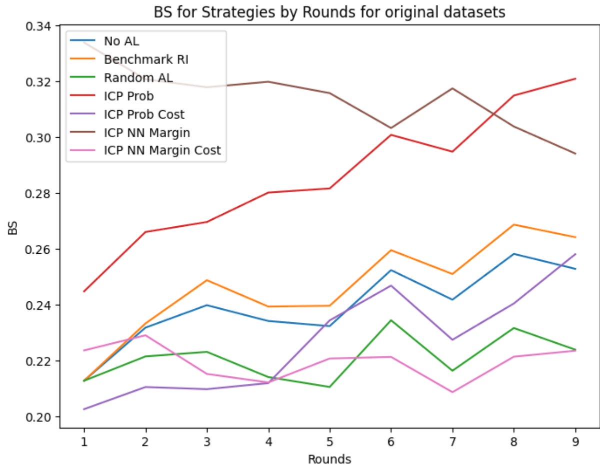


Figure 4.7 BS by Rounds for Original Datasets Averaged

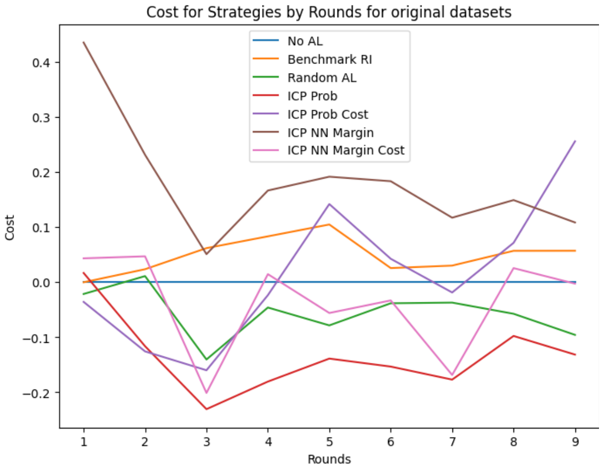


Figure 4.8 Cost by Rounds for Original Datasets Averaged

4.3 Lending Club Full Data

The *LC_all* dataset is the only dataset with real independent variables on both accepted and rejected applications. Only the dependent variable BAD is assumed for the rejected applications to be constantly 1.

Strategies	AUC	PAUC	BS	Cost
No AL	0.850440	0.601950	0.111923	0
Benchmark RI	0.851600	0.587793	0.103894	0.092408
Random AL	0.853536	0.587668	0.104912	0.425888
ICP Prob	0.857101	0.576793	0.108311	0.205660
ICP Prob Cost	0.853043	0.675299	0.180462	0.784457
ICP NN Margin	0.185823	0.577234	0.373880	2.955725
ICP NN Margin Cost	0.848196	0.730519	0.189816	-0.309581

Table 4.3 Performance Scores for LC_all

Benchmark RI scores the best in BS and the lowest above *No AL* regarding economic costs. Albeit *ICP NN Margin Cost* scores a bit worse than its benchmarks regarding AUC and BS, it is by far the best scoring strategy regarding PAUC and economic costs. Apart from *ICP NN Margin*, all strategies in Figure 4.9 improve extensively from round one to round two and stay on that level. While no clear trend is visible for PAUC in Figure 4.10, BS in Figure 4.11 improves in a similar manner as AUC through the rounds. This is because of the significant increase in training data from 3,137 cases to potentially 44,650 that comes from the extreme target feature imbalance in that dataset. *ICP NN Margin Cost* achieves the lowest *Cost* in round two and rises until it approaches the *No AL* benchmark in round nine.

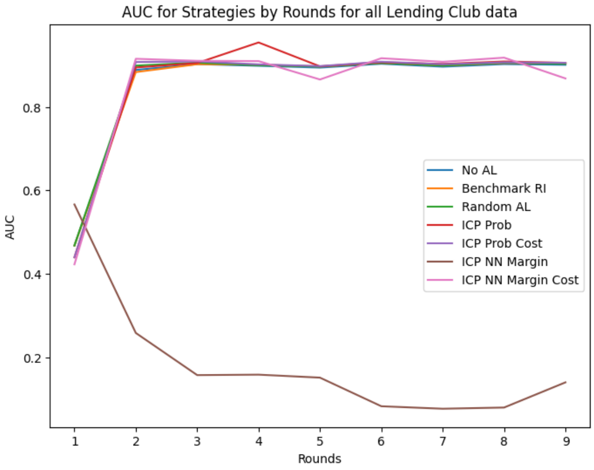


Figure 4.9 AUC by rounds for LC_all

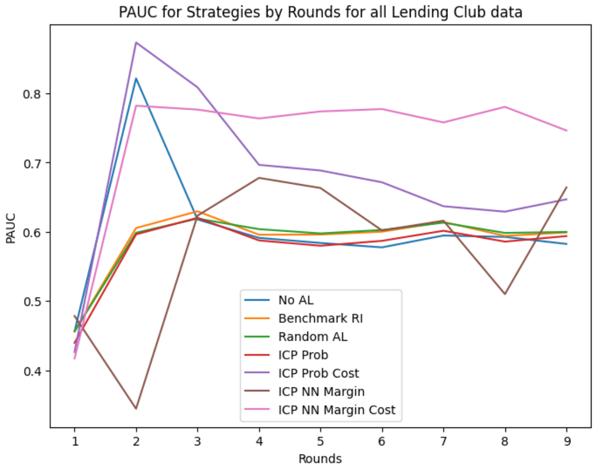


Figure 4.10 PAUC by rounds for LC_all

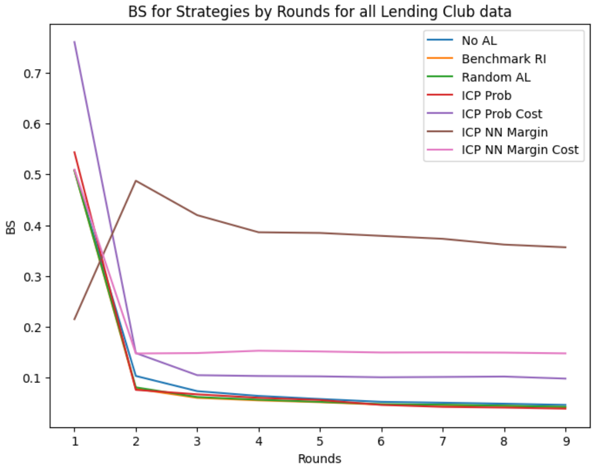


Figure 4.11 BS by rounds for LC_all

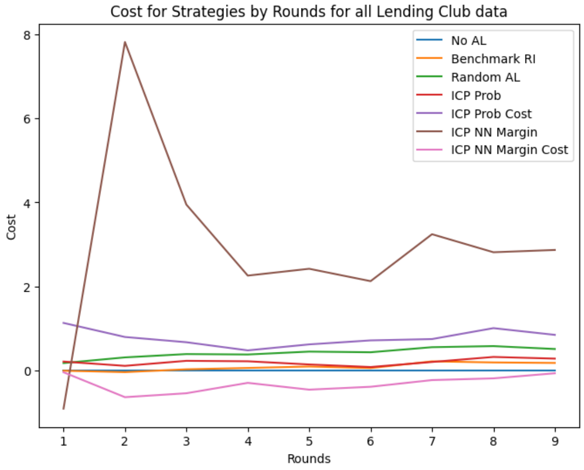


Figure 4.12 Cost by rounds for LC_all

4.4 Monte Carlo Datasets

This group consists of all datasets that have been enhanced with simulated defaulted customers (Small_MC, German_MC, Deloitte_MC, Large_MC). Table 4.4 presents the average performance scores for all rounds averaged for every strategy.

Strategies	AUC	PAUC	BS	Cost
No AL	0.642554	0.596268	0.236761	0
Benchmark RI	0.646839	0.599367	0.243265	-0.029194
Random AL	0.656511	0.596404	0.233087	0.020458
ICP Prob	0.700047	0.563942	0.270610	-0.049046
ICP Prob Cost	0.726493	0.661014	0.213831	0.232596
ICP NN Margin	0.373346	0.476613	0.326456	0.183609
ICP NN Margin Cost	0.586441	0.528336	0.239994	0.062844

Table 4.4 Performance Scores for MC datasets

No clear trend can be seen in Figure 4.13 for the benchmarks *No AL*, *Benchmark RI*, and *Random AL*. This differs for *ICP Prob and ICP Prob Cost*, which start outperforming from round two. There is no clear trend in PAUC progression for *No AL*, *Benchmark RI*, or *Random AL*. *ICP Prob Cost* shows a positive trend, whereas the remaining strategies' trends are mostly negative. *ICP Prob* shows a clear positive trend with an underperforming BS from round one, while *ICP Prob Cost* starts outperforming from round two with a negative BS curve slope. The *Cost* graphs in Figure 4.16 have a slightly negative trend at best.

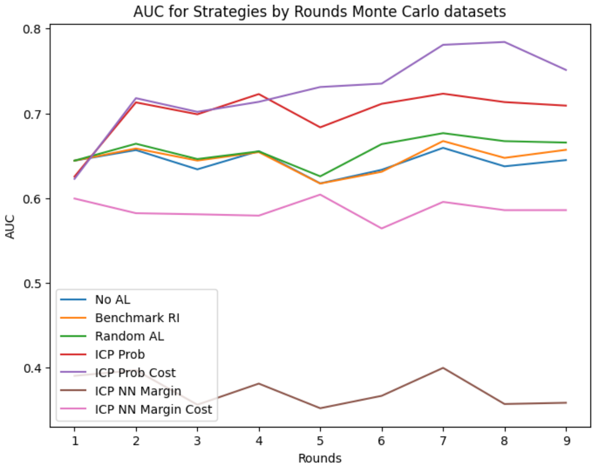


Figure 4.13 AUC by rounds for MC datasets averaged

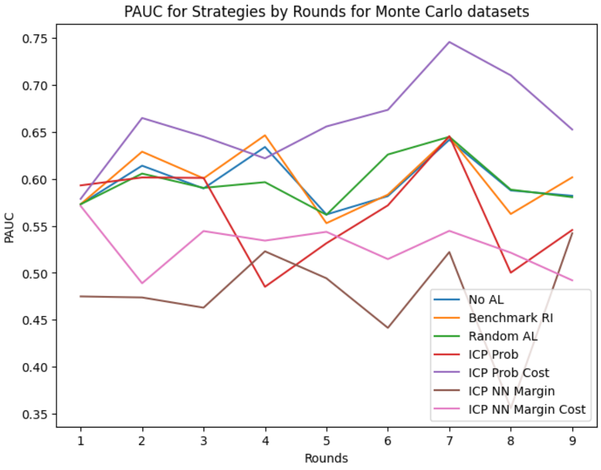


Figure 4.14 PAUC by rounds for MC datasets averaged

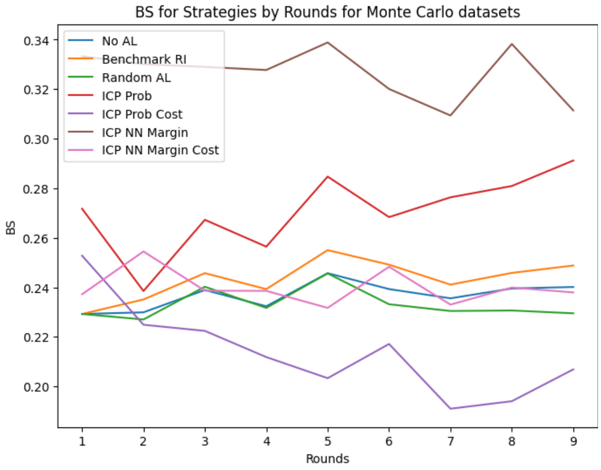


Figure 4.15 BS by rounds for MC datasets averaged

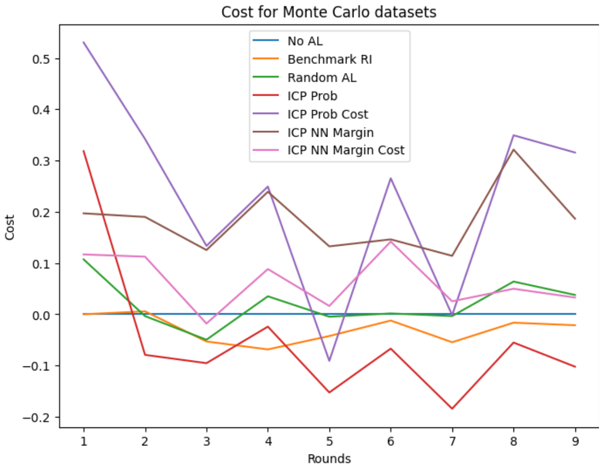


Figure 4.16 Cost by rounds for MC datasets averaged

5. Discussion

ICP NN Margin proves to be a completely impractical strategy, performing worst in all datasets and in virtually every performance metric, scoring even lower than the 0.5 threshold for AUC that marks the "randomly guessing a class" prediction type. Interestingly, this is not the case for its cost-efficient form, which underperforms in most cases apart from *LC_all*. For this reason, only *ICP NN Margin* will be excluded from further analysis.

While the performance rankings for all groups are comparable, *LC_all* falls out of line regarding overall performance and performance on each metric. It is no surprise as it is the only dataset with real rejections that are not based on defaulted customers known to the model from round one on. It is further the only dataset with more bad customers than good ones and the one with the highest target feature imbalance.

Other than its default form, *ICP NN Margin Cost* does not always perform worse than the benchmark strategies and probabilistic ICPs when used on *LC_all*. This comes probably from the unique design of *LC_all*, which incorporates a vast number of rejected customers with characteristics unknown to the learner in the beginning. This change reflected in predicted probabilities is heavily penalized when made cost-efficient. Paradoxically, this advantage shrinks in economic costs with every round. This can be explained by the reversed target feature imbalance in the dataset. While generally, the odds favor changing an FP to a TN for 20 % of the predicted rejections, this is not the case for this dataset with almost only bad customers. It is why *ICP NN Margin Cost* economic costs drop so low in the second round when *No AL* is exposed to nearly only unknown customer characteristics, but then rise with every round as both *No AL* and *ICP NN Margin Cost* learn more from their earlier predictions, but *ICP NN Margin Cost* provides expensive AL loans.

For all dataset groups, the AUC curve is concave for all strategies. Given a useful model, more data always improves, although with a diminishing gradient. PAUC in turn shows no clear slope for any dataset group or strategy. The slopes of BS are mixed, with some increasing and some decreasing across datasets and strategies. As a metric of precision, it is noteworthy that it is rising for all benchmarking strategies when used on original datasets. This is a result of the selection bias in Credit Scoring. As the models become more biased with every round while at the same time fed with more data, the precision of predictions is decreasing, whereas the actual detection of classes reflected by AUC is increasing (PAUC shows no relationship, see Figures 4.2, 4.6, 4.10, 4.14). This also explains why there cannot be seen a clear trend for the MC datasets, which together form the most balanced dataset group, and even a negative trend for *LC_all*, which is the most reverse imbalanced dataset.

Although *Benchmark RI* always performs better than *No AL* regarding *Cost*, *Random AL* generates a similar *Cost* as *No AL* except for *LC_all*. This can be interpreted as a low risk for significant costs when using AL, while there is a guaranteed gain in statistical performance. However, it is still related to a higher economic cost than simple RI techniques.

In beating their benchmarks most of the time, conformal learners are valid AL learners and suitable for RI. But do they pick instances that are most informative or that are the cheapest? Informative instances are generally considered unusual when receiving prediction probabilities close to 0.5 (Settles, 2009; Settles & Craven, 2008; Sharma & Bilgic, 2017). In a scenario with a cost-sensitive Machine Learning model, where CFN > CFP is typically the case (Dumitrescu et al., 2021), and in this experiment, guaranteed unusual instances are always rejected if no AL strategy is implemented. This is because the threshold indicates only the imbalance in costs. A low threshold indicates a significant cost imbalance but does not necessarily imply a lower default rate, and vice versa. Thresholding serves to balance error costs, but cost imbalance does not equate to class imbalance. A 0.4 PD with a 0.3 threshold is therefore not more probable for a default than a 0.5 threshold, yet one scenario rejects the application whereas the other accepts. Hence, with a higher default cost relative to opportunity cost and a higher rate of repaying customers, the share of GOODs in the pool of rejections will be higher. Classifying all rejections as BADs is thus more likely to generate erroneous data, while AL strategies are expected to perform optimally in this scenario.

Although conformal classifiers should still select informative instances under high class imbalance, they are more prone to generate a training set consisting of mostly GOOD customers than if they were instead to select some more BAD customers, given that GOOD customers are in the majority. This ensures superior economic performance but may slightly diminish their statistical performance compared to strategies that select a greater number of BAD customers or accurately classify them without granting loans. On the contrary, statistical performance should increase the most for ICPs if uncertain instances are more likely BADs, harming economic performance. This occurs when the pool of rejected applications contains a higher proportion of BAD customers.

The following subsections intend to conceptualize the interplay of class and cost imbalance and verify it with the empirical results. Starting with an exemplary dataset balanced in both classes and costs, the individual influence of class and cost imbalance is visualized for the representative strategies *Benchmark RI*, *Random AL*, *ICP Prob*, and *ICP Prob Cost*.

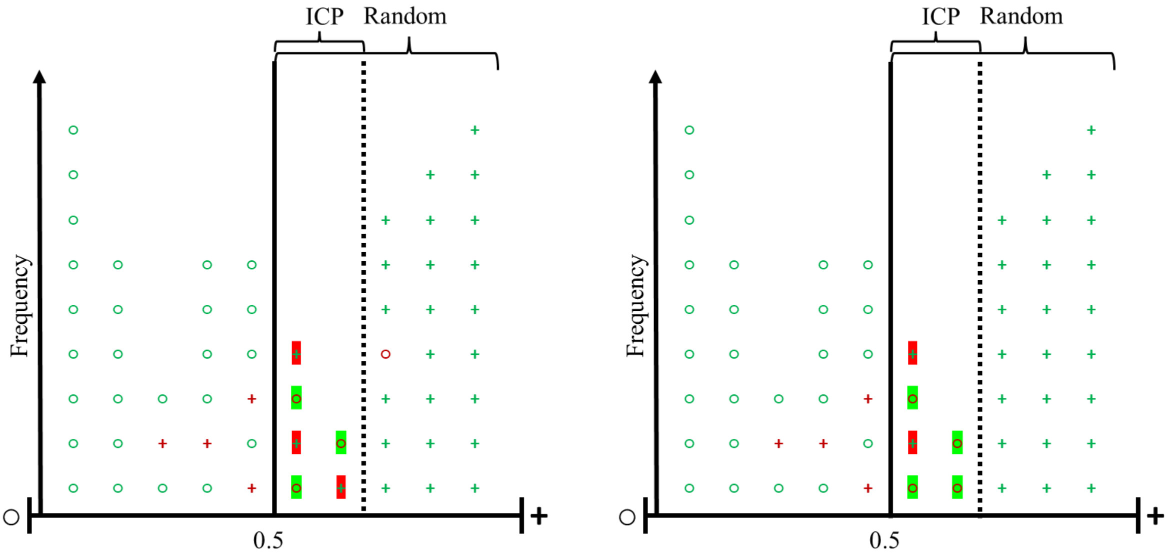
Every scenario has a performance for each strategy that is supposed to scheme the overall performance of the strategy as recorded in the experiment. As an approximation, only the PCC of the selected RI instances is shown here. Since *Benchmark RI* and *Random AL* choose random rejections, the share of correct guesses from all rejections is recorded. The idea is that since both take a random sample from the population of rejections, the expected PCC of their sample is supposed to be the same as that of the population of rejections. *ICP Prob* and *ICP Prob Cost* select the most uncertain instances in accordance with their nonconformity functions. The reclassified instances by ICPs are marked with a painting reverse to their Passive Learning class.

The distributions of predicted probabilities used in the illustrations are only exemplary and do not intend to represent any observed predicted probability distribution. Since all performances depend highly on the distribution of predicted probabilities, which are bound to the effectiveness of the underlying model, which in turn are bound to the data that they were trained on, the presented PCC performances of the strategies are only a possible recording for an indefinite selection of possible predicted probability distributions. As an example, it is supposed to illustrate the tendency of each strategy at best or serve as an illustration for an explanation of recorded performances.

While the predicted probability always determines the predicted class dependent on the cost-efficient threshold, it also determines the nonconformity score for the ICPs. For *ICP Prob*, it is $1-\hat{P}(y|x)$ in every case, whereas for *ICP Prob Cost* it varies in dependence on the actual class of the respective instance in the calibration set. As the calibration scores are bound to the relative nonconformity of the instances in the calibration set, the change in distribution through the dependence of the actual class in the calibration set is also relative. Taking the original cost matrix of *German* as an example, it does not matter whether instances are multiplied by 1 and 4 or 0.25 and 1. For simplicity, only the nonconformity of GOODs in the rejection set is altered to be higher than for *ICP Prob*. This would correspond to multiplying instances by 0.25 and 1. Therefore, the schemes for *ICP Prob Cost* show a nonconformity distribution only (other than for *ICP Prob*, which depicts a predicted probability distribution and a nonconformity distribution at the same time), that is altered for rejected instances in that GOODs are moved one spot to the left at the expense of BADs, as they advance a row in the distribution of nonconformity scores and hence also calibration scores. As in the experiment, a $b_{static\ share}$ is picked, resembling 20 % of the number of acceptances.

5.1 Simple Scenarios

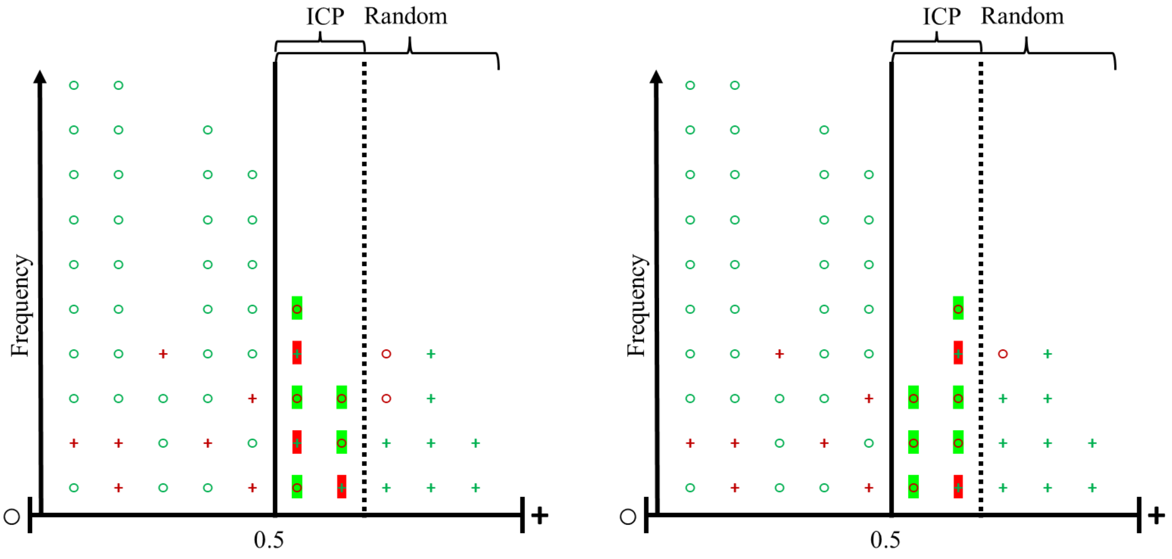
The starting point is a balanced dataset regarding the binary target variables and the same misclassification costs for both error cases. Given a proper learner, one can assume that *Benchmark RI* would achieve a higher PCC for RI instances than *Random AL*. This is shown in the illustrative example Figure 5.1 with a PCC of 0.87 for *Benchmark RI* and 0.13 for *Random AL*. Assuming more FN in the uncertain part of rejections than the certain one, *ICP Prob* is expected to have a higher PCC than *Random AL*, here with 0.5. In shifting GOODs in the rejection set to the left, *ICP Prob Cost* scores even better with 0.67. While *Benchmark RI* scores the best in this example, it is important to note that this resembles only the starting round where every strategy is given the same data to train the same model with, as the distribution of prediction probabilities is the same for all strategies (other than for nonconformity scores). Whereas AL would gather useful data at an economical cost, *Benchmark RI* would create no additional costs but collect erroneous data that would harm the Machine Learning model’s performance in the subsequent rounds.



Expected PCC of chosen RI instances:
Benchmark RI: 0.87
Random AL: 0.13
ICP Prob (left): 0.5
ICP Prob Cost (right): 0.67

Figure 5.1 RI Scenario Class & Cost Balance

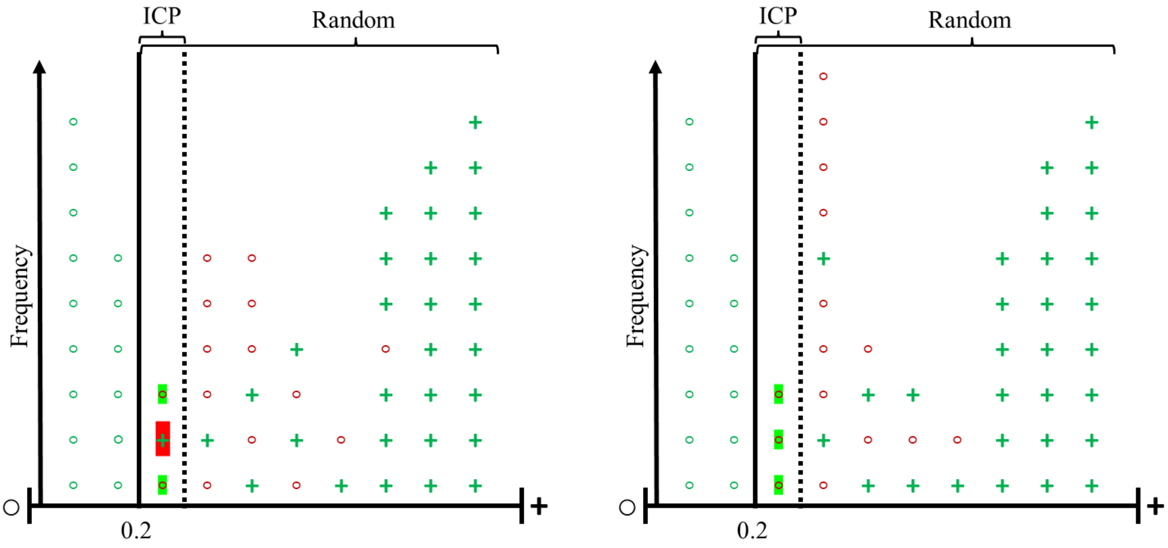
In the case of class imbalance with more GOODs than BADs, AL strategies are expected to work better simply because there are more repaying customers in the set of rejections than in a class-balanced scenario, as shown in Figure 5.2. Logically, *Benchmark RI* would perform worse in this scenario compared to the fully balanced scenario.



Expected PCC of chosen RI instances:
Benchmark RI: 0.61
Random AL: 0.39
ICP Prob (left): 0.63
ICP Prob Cost (right): 0.75

Figure 5.2 RI Scenario GOOD>BAD, Cost Balance

In the case of different misclassification costs, when using a cost-sensitive threshold, the distribution of prediction probabilities and nonconformity scores does not change to the default balanced scenario. Rather, a smaller set of applications with a more confident prediction is given a loan, see Figure 5.3. In contrast, less confident predictions are not granted a loan despite having a lower PD. Naturally, informative applications, i.e., prediction probability close to 0.5, will yield a higher chance of selecting repaying customers than in the default setting if selected.

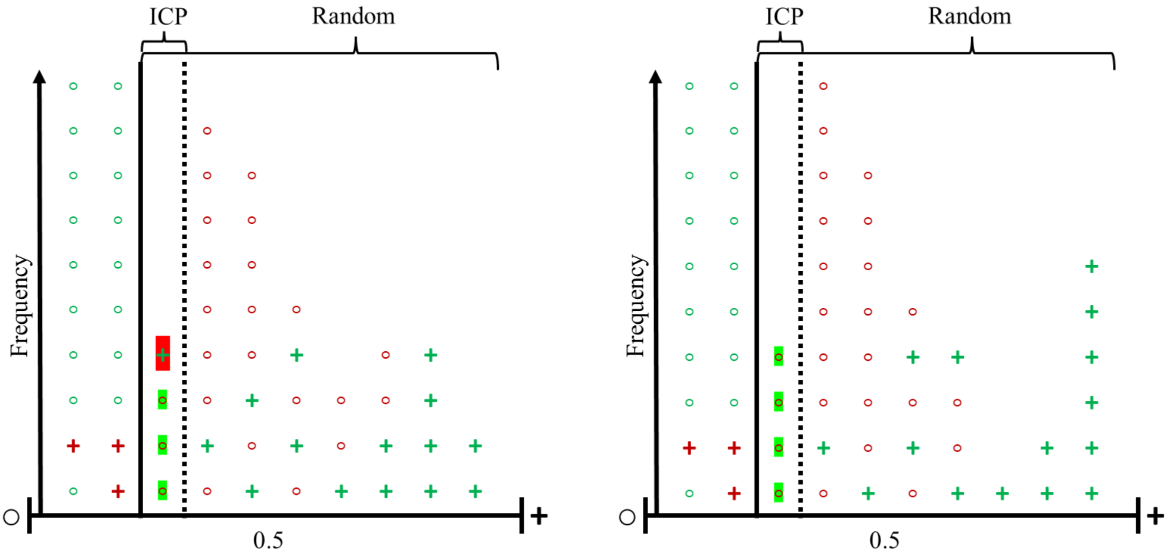


Expected PCC of chosen RI instances:
Benchmark RI: 0.67
Random AL: 0.33
ICP Prob (left): 0.67
ICP Prob Cost (right): 1

Figure 5.3 RI Scenario Class Balance, CFN>CFP

5.2 Conventional Retail Lending

In the typical scenario for Credit Scoring (excluding P2P lending), there are fewer customers who default, but the cost of default is substantially higher than the revenue gained by a repaying customer (Brown & Mues, 2012; Dumitrescu et al., 2021). As illustrated in Figure 5.4, this results in many good customers being rejected due to a lack of confidence in their repaying behavior. In this scenario, *Benchmark RI* performs the worst since many repaying customers are subject to being predicted as bad (not yet accounting for the worse future model performance resulting from erroneous data gathering). AL works best in this scenario, with *ICP Prob Cost* marking the best performance. A detailed look at the performance tables of the individual datasets available in the electronic appendix reveals that indeed there is a visible relationship between AL performance and a class imbalance towards GOODs as well as a cost imbalance towards CFN.



Expected PCC of chosen RI instances:
Benchmark RI: 0.38
Random AL: 0.62
ICP Prob (left): 0.75
ICP Prob Cost (right): 1

Figure 5.4 RI Scenario GOOD>BAD, CFN>CFP

Of all groups, high cost imbalance and class imbalance apply the most to those datasets that were not altered, as they do not contain any rejections that have been assumed to be all BADs. On the contrary, the MC datasets are the most class balanced, whereas their cost imbalance is the same as the one of the original datasets (excluding *LC*). *LC_all* marks the special case of P2P lending where cost imbalance is typical for the Credit Scoring context, however, repayments are clearly outnumbered by rejections, in this experiments defaults. The scenario applicable for *LC_all* is therefore analyzed separately.

The statistical performance of *Benchmark RI* depends mainly on the ability of the underlying model not to predict GOOD customers to be BAD. A model that rejects only customers that would have defaulted does not generate any false data. A model that rejects all defaulting customers and accepts all repaying customers would create a new training data sample that perfectly resembles the population of loan applicants and their distribution. *Benchmark RI* scores hence much worse in the experiment than in the shown illustration, as the average statistical model performance is not the highest (0.655913 AUC of *No AL* on original datasets and 0.643536 AUC of *No AL* on MC datasets). Therefore, it is no surprise that with *No AL*, *Benchmark RI* distinguishes better between GOOD and BAD customers for original datasets than for MC datasets, as the prediction sets are more similar to the training sets.

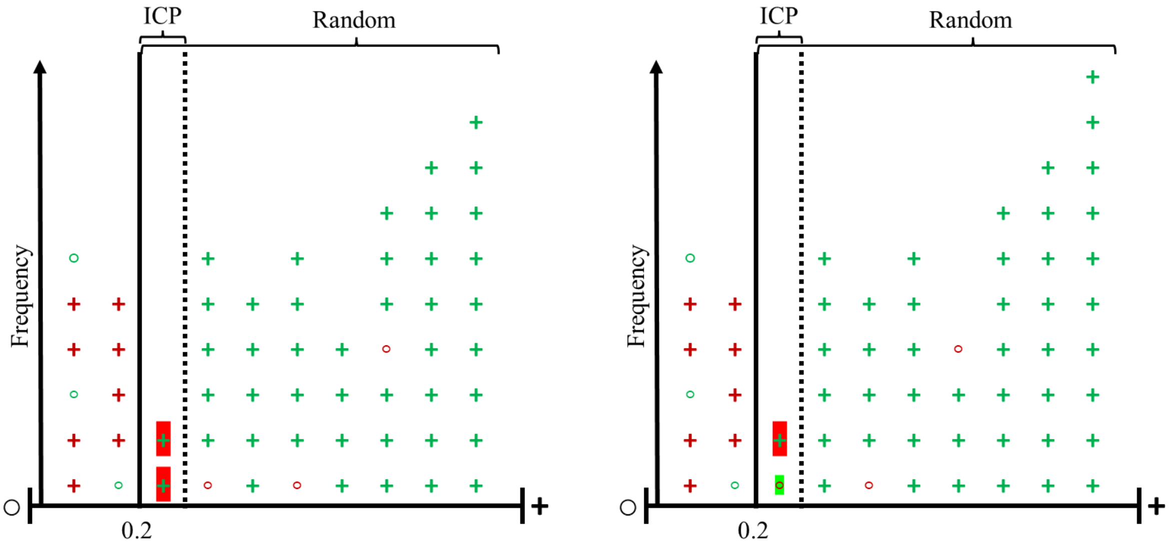
Indeed, *Benchmark RI* outperforms *No AL* more the higher the share of BAD customers in the group of datasets. A similar trend regarding statistical performance can be observed for *Random AL*, albeit with higher economic costs. Albeit selecting defaulting customers, i.e., generating non-erroneous data for *Benchmark RI*, this means generating economic costs for *Random AL*. As only *Benchmark RI* may generate erroneous data, particularly in the case of datasets containing few BADs, *Random AL* outperforms *Benchmark RI* statistically more the smaller the share of defaulting customers (see electronic Appendix).
ICP Prob and *ICP Prob Cost* confirm expectations by selecting unusual instances needed for a better training set. The instances needed for ideal training data, defaults, are still omitted. The performance of ICPs is more favorable for MC datasets compared to original datasets in labeling instances, but this comes at the cost of decreased precision and increased costs, possibly due to the cold start problem and violation of the exchangeability assumption.

ICP Prob selects the cheapest but not the most informative instances, as evident from the steep slope in BS. Moreover, BS rises in every round of original datasets, but from round 1 to round 2, there is a drop in BS for MC datasets, which may indicate progress in addressing the cold start problem. However, after that, the strategy continues to select cheap but not most informative instances, resulting in worse calibration reflected by a steeper BS curve.

Confirming the expectation, *ICP Prob Cost* performs the best for original datasets. In scenarios with greater data imbalance and adherence to the exchangeability assumption, *ICP Prob Cost* selects the least costly and most informative instances, resulting in superior scores across all performance metrics. This is not the case for the more class-balanced MC datasets, where *ICP Prob Cost* generates the best statistical model but at a high economic cost (0.161024).

5.3 Peer-to-Peer Lending

P2P lending is marked by an extreme imbalance towards BAD customers, while the cost imbalance does not differ from conventional retail lending. Figure 5.5 illustrates this scenario. *Benchmark RI* is expected to perform by far the best, whereas *Random AL* catches a few GOOD customers and *ICP Prob Cost* manages to capture at least more than *ICP Prob*.



Expected PCC of chosen RI instances:
Benchmark RI: 0.94
Random AL: 0.06
ICP Prob (left): 0
ICP Prob Cost (right): 0.5

Figure 5.5 RI Scenario BAD>GOOD, CFN>CFP

The empirical results clearly contradict this scenario. *Benchmark RI* scores only slightly better than *Random AL*, although better regarding Cost. As expected, a random sample from a sample representing rejected applicants of a population with an assumed share of defaults of over 90% makes it cheaper to use simple classifying rather than giving loans.

ICP Prob seems to sample informative instances again, but still cheaper ones rather than the most informative ones. In an imbalanced setting like this, achieving the best AUC (0.857101) comes at a significant cost compared to not implementing any RI technique (0.205660). This economic cost balances out the worsening calibration, as *ICP Prob's* BS curve is not rising. Other than expected, *ICP Prob Cost* does not outperform *ICP Prob* either. Despite achieving the best PAUC (0.675299) and a competitive AUC (0.853043), it exhibits the poorest calibration (0.180462) and the highest *Cost* (0.784457) among the four strategies. The modification to the calibration set has clearly not produced the intended improvement and instead resulted in worsened performance. The substantial violation of the assumption of exchangeability shows the strongest harm of this strategy.

6. Future Research

As an AL strategy, conformal prediction does not differ extensively from other AL strategies. Its distinguishing feature lies in assigning nonconformity scores to a distribution through calibration ranks, providing contextualization for raw scores. Instead of using the nonconformity scores and picking b with the largest score, the alternative approach involves classifying rejections as certain and uncertain based on a specified p-value to make the number of AL customers dependent on their individual case and compare their rank with the calibration set rather than prediction set. Finding ways for an appropriate p-value for minimizing costs and/or maximizing predictive accuracy and/or predictive discrimination offers avenues for future research.

The experiment shed light on the many factors influencing the performance of RI, AL, and AL strategies based on conformal predictors. In using multiple datasets that all differ in their characteristics, multiple factors were observed, although complete isolation of individual factors from one another proved challenging. A deeper look at every individual aspect should verify the conclusions from this experiment in more detail. Making use of the rich data of the Lending Club presents an excellent opportunity for such in-depth exploration.

Trying out different ways of altering the nonconformity function for specific purposes is a further topic for future research. For instance, altering the default or cost-sensitive nonconformity function based on the proportion of defaults could be explored. As thresholding for model cost-sensitivity is only exemplary, other ways of threshold tuning and cost-sensitivity are recommended to be explored for testing conformal AL strategies in Credit Scoring. Examples include MetaCost (Domingos, 1999) and Expected Maximum Profit (Verbraken et al., 2014). Furthermore, augmenting the experiment with additional performance metrics commonly employed in Credit Scoring(Lessmann et al., 2015) or performance metrics that are specific to RI, like the kickout score introduced by Kozodoi et al. (2019), would further enhance its comprehensiveness.

Although AL is recommended in the case of data scarcity, ICPs require a minimum size of calibration data set aside to yield optimal results (Angelopoulos & Bates, 2022; Linusson et al., 2014). As setting aside data has a more pronounced impact when data is already scarce, it is worth trying TCPs in this scenario, as the computational resources needed should be limited due to the scarcity of data.

This experiment tested biased scorecards with data that included rejected applicants. The rejections were assumed to be all FPs, which may be somewhat unrealistic. Trying AL for complete Credit Scoring datasets containing information on FPs should be interesting, particularly in the case of extreme rejection rates, as they are common in P2P lending. For MC simulations, this could be accomplished with different ways of rounding the simulated random numbers for the default status to achieve a desired balance of TPs and FPs in the set of simulated rejections.

The MC simulation of defaulting customers was used in this experiment solely to alter the prediction set of each round. The assumed setting was that a biased model is used on unbiased data. Alternatively, the biased model could be retrained before prediction by simulating defaulting customers and retraining the model using semi-artificial data, as done for the test data of each round in the experiment. Other multivariate MC simulation approaches, such as copulas (Jaworski et al., 2010), can be further explored for generating multidimensional data following specific distributions.

7. Conclusion

In this master thesis, conformal predictors are shown to be suitable for AL as a RI technique in Credit Scoring. Their performance is dependent on their nonconformity function and its eventual cost-sensitive recalibration. A detailed look reveals that in the case of a nonconformity function based on probabilistic error, a cost-insensitive nonconformity function selects cheaper but not necessarily the most informative instances. In contrast, the cost-sensitive nonconformity function allows one to select both the cheapest and most informative applications if used in an environment where opportunity cost is smaller than the cost of default and repaying customers outnumber defaulting ones. While cost-insensitive nonconformity functions perform rather robustly in different class and cost imbalance scenarios, its cost-sensitive form highly depends on these factors that shape the calibration set used for assessing nonconformity. In the most extreme case of *ICP NN Margin Cost*, these factors can change an underdog to a best-performing strategy.

The importance of several factors for AL strategies in Credit Scoring are revealed, with class imbalance being one of the most important regarding the economic feasibility of AL. The relevance for RI in general and the superiority of AL is the highest when used for imbalanced datasets, where labeling the minor class is associated with costs rather than the major class

(Attenberg & Provost, 2011; Carcillo et al., 2018). This is entirely fulfilled in the conventional lending scenario, whereas P2P lending with the assumption of non-erroneous rejections does not fulfill it.

Based on these findings, it is suggested to implement AL in the industry where the repayment of loans is short, such as for credit cards or BNPL, or where an early detection system can effectively identify potential defaults well in advance. For example, a proxy for default could involve payments that need to be made more frequently than the established monthly payment. A specified number of missed payments could then serve as an indicator of default, treating the respective loan as such if sufficiently certain, even if the customer has not defaulted yet. A conformal prediction of time to default by combining the fields of Conformity and Survival Analysis might give promising results. Concave AUC and flattening *Cost* curves indicate that it is particularly beneficial when a cold start problem exists in that high-quality data is scarce and must be gathered first. A low proportion of defaulting customers and a small revenue-to-loan ratio further enhance the suitability of AL in Credit Scoring. Hence, BNPL or Microcredit for unbanked consumers are only two of many possible application areas.

Regardless of the above-mentioned factors, financial institutions are recommended to employ at least some AL to update their scorecards appropriately and ensure their performance over time (Abdou & Pointon, 2011), although a well-performing scorecard may make AL redundant and simpler RI assumptions work better. By granting loans to consumers who would have been denied otherwise, financial institutions allow individuals to improve their credit scores, assuming the denied consumer repays the loan[18]. Repayment of the loan enables credit bureaus to acquire positive data about the consumer, allowing former subprime borrowers to update their low credit score to a more accurate representation in a shorter timeframe. Thus, consumers can rebuild their credit scores more quickly, whilst financial institutions gain informative data for their scorecards. In the case of a formerly unbanked consumer, the financial institution further created itself a new customer and makes her part of the formal financial system.

[18] See this blog post of one of the three largest credit reporting companies in the US and one of the largest worldwide: https://www.experian.com/blogs/ask-experian/will-paying-off-a-loan-improve-credit/

References

Abdou, H. A., & Pointon, J. (2011). Credit Scoring, Statistical Techniques and Evaluation Criteria: A Review of the Literature. *Intelligent Systems in Accounting, Finance and Management*, *18*(2–3), 59–88. https://doi.org/10.1002/isaf.325

Ahlberg, E., Spjuth, O., Hasselgren, C., & Carlsson, L. (2015). Interpretation of Conformal Prediction Classification Models. In A. Gammerman, V. Vovk, & H. Papadopoulos (Eds.), *Statistical Learning and Data Sciences* (Vol. 9047, pp. 323–334). Springer International Publishing. https://doi.org/10.1007/978-3-319-17091-6_27

Akkoç, S. (2012). An empirical comparison of conventional techniques, neural networks and the three stage hybrid Adaptive Neuro Fuzzy Inference System (ANFIS) model for credit scoring analysis: The case of Turkish credit card data. *European Journal of Operational Research*, *222*(1), 168–178. https://doi.org/10.1016/j.ejor.2012.04.009

Alejo, R., García, V., Marqués, A. I., Sánchez, J. S., & Antonio-Velázquez, J. A. (2013). Making Accurate Credit Risk Predictions with Cost-Sensitive MLP Neural Networks. In J. Casillas, F. J. Martínez-López, R. Vicari, & F. De La Prieta (Eds.), *Management Intelligent Systems* (Vol. 220, pp. 1–8). Springer International Publishing. https://doi.org/10.1007/978-3-319-00569-0_1

Allamanis, M. (2019). The adverse effects of code duplication in machine learning models of code. *Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, 143–153. https://doi.org/10.1145/3359591.3359735

Anderson, B. (2022). Naive Bayes using the expectation-maximization algorithm for reject inference. *Communications in Statistics: Case Studies, Data Analysis and Applications*, *8*(3), 484–504. https://doi.org/10.1080/23737484.2022.2106325

Anderson, B., Newman, M. A., Grim, P. A., & Hardin, J. M. (2022). A Monte Carlo simulation framework for reject inference. *Journal of the Operational Research Society*, 1–17. https://doi.org/10.1080/01605682.2022.2057819

Angelopoulos, A. N., & Bates, S. (2022). *A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification*. https://doi.org/10.48550/ARXIV.2107.07511

Attenberg, J., & Provost, F. (2011). Inactive learning?: Difficulties employing active learning in practice. *ACM SIGKDD Explorations Newsletter*, *12*(2), 36–41. https://doi.org/10.1145/1964897.1964906

Auerswald, M., & Moshagen, M. (2015). Generating Correlated, Non-normally Distributed Data Using a Non-linear Structural Model. *Psychometrika*, *80*(4), 920–937. https://doi.org/10.1007/s11336-015-9468-7

B. Baesens, T. Van Gestel, S. Viaene, Stepanova, M., J. Suykens, & J. Vanthienen. (2003). Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring. *The Journal of the Operational Research Society*, *54*(6), 627–635. JSTOR.

Bahnsen, A. C., Aouada, D., & Ottersten, B. (2014). Example-Dependent Cost-Sensitive Logistic Regression for Credit Scoring. *2014 13th International Conference on Machine Learning and Applications*, 263–269. https://doi.org/10.1109/ICMLA.2014.48

Banasik, J., Crook, J., & Thomas, L. (2003). Sample selection bias in credit scoring models. *Journal of the Operational Research Society*, *54*(8), 822–832. https://doi.org/10.1057/palgrave.jors.2601578

Barakova, I., Glennon, D., & Palvia, A. A. (2011). Adjusting for Sample Selection Bias in Acquisition Credit Scoring Models. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1722382

Bart Baesens. (2003). *Developing intelligent systems for credit scoring using machine learning techniques*. Katholieke Universiteit Leuven.

Beling, P., Covaliu, Z., & Oliver, R. M. (2005). Optimal scoring cutoff policies and efficient frontiers. *Journal of the Operational Research Society*, *56*(9), 1016–1029. https://doi.org/10.1057/palgrave.jors.2602021

Blumenstock, G., Lessmann, S., & Seow, H.-V. (2022). Deep learning for survival and competing risk modelling. *Journal of the Operational Research Society*, *73*(1), 26–38. https://doi.org/10.1080/01605682.2020.1838960

Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, *39*(3), 3446–3453. https://doi.org/10.1016/j.eswa.2011.09.033

Carcillo, F., Le Borgne, Y.-A., Caelen, O., & Bontempi, G. (2018). Streaming active learning strategies for real-life credit card fraud detection: Assessment and visualization. *International Journal of Data Science and Analytics*, *5*(4), 285–300. https://doi.org/10.1007/s41060-018-0116-z

Carta, S., Ferreira, A., Reforgiato Recupero, D., Saia, M., & Saia, R. (2020). A combined entropy-based approach for a proactive credit scoring. *Engineering Applications of Artificial Intelligence*, *87*, 103292. https://doi.org/10.1016/j.engappai.2019.103292

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, *6*(1), 1–6. https://doi.org/10.1145/1007730.1007733

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving Generalization with Active Learning. *Machine Learning*, *15*(2), 201–221. https://doi.org/10.1023/A:1022673506211

Crone, S. F., & Finlay, S. (2012). Instance sampling in credit scoring: An empirical study of sample size and balancing. *International Journal of Forecasting*, *28*(1), 224–238. https://doi.org/10.1016/j.ijforecast.2011.07.006

Crook, J., & Banasik, J. (2004). Does reject inference really improve the performance of application scoring models? *Journal of Banking & Finance*, *28*(4), 857–874. https://doi.org/10.1016/S0378-4266(03)00203-6

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

Denis, C., & Hebiri, M. (2015). Confidence Sets for Classification. In A. Gammerman, V. Vovk, & H. Papadopoulos (Eds.), *Statistical Learning and Data Sciences* (Vol. 9047, pp. 301–312). Springer International Publishing. https://doi.org/10.1007/978-3-319-17091-6_25

Domingos, P. (1999). *Metacost: A general method for making classifiers cost-sensitive*. 155–164.

Donmez, P., Carbonell, J. G., & Bennett, P. N. (2007). Dual Strategy Active Learning. In J. N. Kok, J. Koronacki, R. L. de Mantaras, S. Matwin, D. Mladenič, & A. Skowron (Eds.), *Machine Learning: ECML 2007* (pp. 116–127). Springer Berlin Heidelberg.

Dumitrescu, E.-I., Hué, S., Hurlin, C., & Tokpavi, S. (2021). Machine Learning or Econometrics for Credit Scoring: Let's Get the Best of Both Worlds. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3553781

El Annas, M., Benyacoub, B., & Ouzineb, M. (2022). Semi-supervised adapted HMMs for P2P credit scoring systems with reject inference. *Computational Statistics*. https://doi.org/10.1007/s00180-022-01220-9

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Fernández-Tobías, I., Tomeo, P., Cantador, I., Di Noia, T., & Di Sciascio, E. (2016). Accuracy and Diversity in Cross-domain Recommendations for Cold-start Users with Positive-only Feedback. *Proceedings of the 10th ACM Conference on Recommender Systems*, 119–122. https://doi.org/10.1145/2959100.2959175

Fontana, M., Zeni, G., & Vantini, S. (2020). *Conformal Prediction: A Unified Review of Theory and New Challenges*. https://doi.org/10.48550/ARXIV.2005.07972

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, *33*(1). https://doi.org/10.18637/jss.v033.i01

Fu, Y., Zhu, X., & Li, B. (2013). A survey on instance selection for active learning. *Knowledge and Information Systems*, *35*(2), 249–283. https://doi.org/10.1007/s10115-012-0507-8

Geisser, S. (1993). *Predictive inference: An introduction*. Chapman & Hall.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182. https://doi.org/10.1162/153244303322753616

Haibo He, & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Hand, D. J. (2005). Good practice in retail credit scorecard assessment. *Journal of the Operational Research Society*, *56*(9), 1109–1117. https://doi.org/10.1057/palgrave.jors.2601932

Hand, D. J., & Henley, W. E. (1993). Can reject inference ever work? *IMA Journal of Management Mathematics*, *5*(1), 45–55. https://doi.org/10.1093/imaman/5.1.45

Hand, D. J., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer New York. https://doi.org/10.1007/978-0-387-84858-7

Henley, William Edward. (1995). *Statistical aspects of credit scoring*. https://doi.org/10.21954/OU.RO.0000E061

Hurlin, C., & Pérignon, C. (2020). Machine learning et nouvelles sources de données pour le scoring de crédit: *Revue d'économie Financière*, *N° 135*(3), 21–50. https://doi.org/10.3917/ecofi.135.0021

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study1. *Intelligent Data Analysis*, *6*(5), 429–449. https://doi.org/10.3233/IDA-2002-6504

Jaworski, P., Durante, F., Härdle, W. K., & Rychlik, T. (Eds.). (2010). *Copula Theory and Its Applications: Proceedings of the Workshop Held in Warsaw, 25-26 September 2009* (Vol. 198). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-12465-5

Jiang, L., Qiu, C., & Li, C. (2015). A Novel Minority Cloning Technique for Cost-Sensitive Learning. *International Journal of Pattern Recognition and Artificial Intelligence*, *29*(04), 1551004. https://doi.org/10.1142/S0218001415510040

Johansson, U., Linusson, H., Lofstrom, T., & Bostrom, H. (2017). Model-agnostic nonconformity functions for conformal classification. *2017 International Joint Conference on Neural Networks (IJCNN)*, 2072–2079. https://doi.org/10.1109/IJCNN.2017.7966105

Kim, Y., & Sohn, S. Y. (2007). Technology scoring model considering rejected applicants and effect of reject inference. *Journal of the Operational Research Society*, *58*(10), 1341–1347. https://doi.org/10.1057/palgrave.jors.2602306

Kong, Q., Siauw, T., & Bayen, A. M. (2021). Chapter 9—Representation of Numbers. In Q. Kong, T. Siauw, & A. M. Bayen (Eds.), *Python Programming and Numerical Methods* (pp. 145–156). Academic Press. https://doi.org/10.1016/B978-0-12-819549-9.00018-X

Kozodoi, N., Katsas, P., Lessmann, S., Moreira-Matias, L., & Papakonstantinou, K. (2019). *Shallow Self-Learning for Reject Inference in Credit Scoring*. https://doi.org/10.1007/978-3-030-46133-1_31

Lehmann, E. L., & Romano, J. P. (2005). *Testing statistical hypotheses* (3rd ed). Springer.

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

Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. *Expert Systems with Applications*, *41*(4), 2065–2073. https://doi.org/10.1016/j.eswa.2013.09.005

Linusson, H. (2017). *An introduction to conformal prediction*. 6th Symp. Conformal and Probabilistic Prediction Appl.

Linusson, H., Johansson, U., Boström, H., & Löfström, T. (2014). Efficiency Comparison of Unstable Transductive and Inductive Conformal Classifiers. In E. Bayro-Corrochano & E. Hancock (Eds.), *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (Vol. 8827, pp. 261–270). Springer International Publishing. https://doi.org/10.1007/978-3-662-44722-2_28

Linusson, H., Norinder, U., Boström, H., Johansson, U., & Löfström, T. (2017). On the Calibration of Aggregated Conformal Predictors. *Proceedings of Machine Learning Research :*

Marqués, A. I., García, V., & Sánchez, J. S. (2013). On the suitability of resampling techniques for the class imbalance problem in credit scoring. *Journal of the Operational Research Society*, *64*(7), 1060–1070. https://doi.org/10.1057/jors.2012.120

Marshall, A., Tang, L., & Milne, A. (2010). Variable reduction, sample selection bias and bank retail credit scoring. *Journal of Empirical Finance*, *17*(3), 501–512. https://doi.org/10.1016/j.jempfin.2009.12.003

Martens, D., Baesens, B. B., & Van Gestel, T. (2009). Decompositional Rule Extraction from Support Vector Machines by Active Learning. *IEEE Transactions on Knowledge and Data Engineering*, *21*(2), 178–191. https://doi.org/10.1109/TKDE.2008.131

Martins, V. E., Cano, A., & Barbon Junior, S. (2023). Meta-learning for dynamic tuning of active learning on stream classification. *Pattern Recognition*, *138*, 109359. https://doi.org/10.1016/j.patcog.2023.109359

Matiz, S., & Barner, K. E. (2020). Conformal prediction based active learning by linear regression optimization. *Neurocomputing*, *388*, 157–169. https://doi.org/10.1016/j.neucom.2020.01.018

Mester, L. (1997). What Is the Point of Credit Scoring? *Business Review*, *Sep/Oct*, 3–16.

Min, F., Liu, F.-L., Wen, L.-Y., & Zhang, Z.-H. (2019). Tri-partition cost-sensitive active learning through kNN. *Soft Computing*, *23*(5), 1557–1572. https://doi.org/10.1007/s00500-017-2879-x

Montrichard, D. (2007). Reject inference methodologies in credit risk modeling. *The Proceedings of the South-East SAS Users Group*.

Nayak, G., & Turvey, C. G. (1997). Credit Risk Assessment and the Opportunity Costs of Loan Misclassification. *Canadian Journal of Agricultural Economics-Revue Canadienne D Agroeconomie*, *45*, 285–299.

Nguyen, V.-L., Shaker, M. H., & Hüllermeier, E. (2022). How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, *111*(1), 89–122. https://doi.org/10.1007/s10994-021-06003-9

Nikolaidis, D., Doumpos, M., & Zopounidis, C. (2017). Exploring Population Drift on Consumer Credit Behavioral Scoring. In E. Grigoroudis & M. Doumpos (Eds.), *Operational Research in Business and Economics* (pp. 145–165). Springer International Publishing. https://doi.org/10.1007/978-3-319-33003-7_7

Norinder, U., Carlsson, L., Boyer, S., & Eklund, M. (2014). Introducing Conformal Prediction in Predictive Modeling. A Transparent and Flexible Alternative to Applicability Domain Determination. *Journal of Chemical Information and Modeling*, *54*(6), 1596–1603. https://doi.org/10.1021/ci5001168

Nouretdinov, I., Gammerman, J., Fontana, M., & Rehal, D. (2020). Multi-level conformal clustering: A distribution-free technique for clustering and anomaly detection. *Neurocomputing*, *397*, 279–291. https://doi.org/10.1016/j.neucom.2019.07.114

Nouretdinov, I., Melluish, T., & Vovk, V. (2001). *Ridge Regression Confidence Machine.* 385–392.

Ogundimu, E. O. (2022). On Lasso and adaptive Lasso for non-random sample in credit scoring. *Statistical Modelling*, 1471082X2210921. https://doi.org/10.1177/1471082X221092181

Oliver, R. M., & Thomas, L. C. (2009). *Optimal score cutoffs and pricing in regulatory capital in retail credit portfolios*.

Papadopoulos, H. (2015). Cross-Conformal Prediction with Ridge Regression. In A. Gammerman, V. Vovk, & H. Papadopoulos (Eds.), *Statistical Learning and Data Sciences* (Vol. 9047, pp. 260–270). Springer International Publishing. https://doi.org/10.1007/978-3-319-17091-6_21

Papadopoulos, H., Proedrou, K., Vovk, V., & Gammerman, A. (2002). Inductive Confidence Machines for Regression. In T. Elomaa, H. Mannila, & H. Toivonen (Eds.), *Machine Learning: ECML 2002* (pp. 345–356). Springer Berlin Heidelberg.

Papadopoulos, H., Vovk, V., & Gammerman, A. (2002). *Qualified Prediction for Large Data Sets in the Case of Pattern Recognition.* 159–163.

Petrides, G., Moldovan, D., Coenen, L., Guns, T., & Verbeke, W. (2020). Cost-sensitive learning for profit-driven credit scoring. *Journal of the Operational Research Society*, *73*(2), 338–350. https://doi.org/10.1080/01605682.2020.1843975

Pundir, S., & Seshadri, R. (2012). *A Novel Concept of Partial Lorenz Curve and Partial Gini Index*.

Richards, J. W., Starr, D. L., Brink, H., Miller, A. A., Bloom, J. S., Butler, N. R., James, J. B., Long, J. P., & Rice, J. (2012). Active Learning to Overcome Sample Selection Bias: Application to Photometric Variable Star Classification. *The Astrophysical Journal*, *744*(2), 192. https://doi.org/10.1088/0004-637X/744/2/192

Saia, R., & Carta, S. (2016). An Entropy Based Algorithm for Credit Scoring. In A. M. Tjoa, L. D. Xu, M. Raffai, & N. M. Novak (Eds.), *Research and Practical Issues of Enterprise Information Systems* (Vol. 268, pp. 263–276). Springer International Publishing. https://doi.org/10.1007/978-3-319-49944-4_20

Schechtman, E., & Schechtman, G. (2016). The Relationship between Gini Methodology and the ROC curve. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2739245

Settles, B. (2009). *Active Learning Literature Survey*.

Settles, B., & Craven, M. W. (2008). An Analysis of Active Learning Strategies for Sequence Labeling Tasks. *Conference on Empirical Methods in Natural Language Processing*.

Shafer, G., & Vovk, V. (2008). *A tutorial on conformal prediction*. https://doi.org/10.48550/ARXIV.0706.3188

Sharma, M., & Bilgic, M. (2017). Evidence-based uncertainty sampling for active learning. *Data Mining and Knowledge Discovery*, *31*, 164–202.

Sheng, V., & Ling, C. (2006). Thresholding for Making Classifiers Cost Sensitive. *Proceedings of the National Conference on Artificial Intelligence*, *1*.

Siddiqi, N. (2012). *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. Wiley. https://books.google.de/books?id=SEbCeN3-kEUC

Son, L. H. (2016). Dealing with the new user cold-start problem in recommender systems: A comparative review. *Information Systems*, *58*, 87–104. https://doi.org/10.1016/j.is.2014.10.001

Song, M., Wang, J., & Su, S. (2022). *Towards a Better Microcredit Decision* (arXiv:2209.07574). arXiv. http://arxiv.org/abs/2209.07574

Suri, T., Bharadwaj, P., & Jack, W. (2021). Fintech and household resilience to shocks: Evidence from digital loans in Kenya. *Journal of Development Economics*, *153*, 102697. https://doi.org/10.1016/j.jdeveco.2021.102697

Thanuja, V., Venkateswarlu, B., & Anjaneyulu, G. (2011). Applications of data mining in customer relationship management. *Journal of Computer and Mathematical Sciences*, *2*(3), 399–580.

Vazquez, J., & Facelli, J. C. (2022). Conformal Prediction in Clinical Medical Sciences. *Journal of Healthcare Informatics Research*, *6*(3), 241–252. https://doi.org/10.1007/s41666-021-00113-8

Verbraken, T., Bravo, C., Weber, R., & Baesens, B. (2014). Development and application of consumer credit scoring models using profit-based classification measures. *European Journal of Operational Research*, *238*(2), 505–513. https://doi.org/10.1016/j.ejor.2014.04.001

Verstraeten, G., & Van den Poel, D. (2005). The Impact of Sample Bias on Consumer Credit Scoring Performance and Profitability. *The Journal of the Operational Research Society*, *56*(8), 981–992.

Viaene, S., & Dedene, G. (2005). Cost-sensitive learning and decision making revisited. *European Journal of Operational Research*, *166*(1), 212–220. https://doi.org/10.1016/j.ejor.2004.03.031

Vovk, V. (2002). On-line confidence machines are well-calibrated. *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, 187–196. https://doi.org/10.1109/SFCS.2002.1181895

Vovk, V. (2012). *Cross-conformal predictors* (arXiv:1208.0806). arXiv. http://arxiv.org/abs/1208.0806

Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic Learning in a Random World*. Springer-Verlag. https://doi.org/10.1007/b106715

Wallace, B. C., & Dahabreh, I. J. (2014). Improving class probability estimates for imbalanced data. *Knowledge and Information Systems*, *41*(1), 33–52. https://doi.org/10.1007/s10115-013-0670-6

West, D. (2000). Neural network credit scoring models. *Computers & Operations Research*, *27*(11–12), 1131–1152. https://doi.org/10.1016/S0305-0548(99)00149-5

Yang, Y., & Loog, M. (2016). *A Benchmark and Comparison of Active Learning for Logistic Regression*. https://doi.org/10.48550/ARXIV.1611.08618

Yeo, I.-K., & Johnson, R. A. (2000). A New Family of Power Transformations to Improve Normality or Symmetry. *Biometrika*, *87*(4), 954–959. JSTOR.

Zhan, X., Liu, H., Li, Q., & Chan, A. B. (2021). A Comparative Survey: Benchmarking for Pool-based Active Learning. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 4679–4686. https://doi.org/10.24963/ijcai.2021/634

Zhao, Y., Cao, Y. C., Pan, X. Q., Yong Lu, & Xu, X. N. (2008). A telecom clients credit risk rating model based on active learning. *2008 IEEE International Conference on Automation and Logistics*, 2590–2593. https://doi.org/10.1109/ICAL.2008.4636608

Zhao, Y., Li, L., Wang, H., Cai, H., Bissyandé, T. F., Klein, J., & Grundy, J. (2021). On the Impact of Sample Duplication in Machine-Learning-Based Android Malware Detection. *ACM Transactions on Software Engineering and Methodology*, *30*(3), 1–38. https://doi.org/10.1145/3446905

Zhu, J., Wang, H., Yao, T., & Tsou, B. K. (2008). Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. *Proceedings of the 22nd International Conference on Computational Linguistics - COLING ’08*, *1*, 1137–1144. https://doi.org/10.3115/1599081.1599224

Appendices

Appendix A: Monte Carlo Simulation Procedure

The MC simulation procedure works essentially by generating random standard normally distributed numbers and correlating them with the Cholesky decomposed correlation matrix of the underlying dataset. The pseudo-code for this procedure is given below in Algorithm 2. As a first step, the dataset needs to be transformed to contain only variables of numeric value. This is achieved by using a dummy variable for every category of a categorical feature. Next, a Yeo-Johnson Power Transformation (Yeo & Johnson, 2000) is performed to make the distribution of every variable become more like a Gaussian one. This is necessary because the Cholesky decomposition as a means to correlate simulations based on the correlation matrix of an underlying dataset works only if the variables of the underlying dataset all have a Gaussian distribution (Auerswald & Moshagen, 2015). The Cholesky decomposition requires a matrix to be symmetric positive definite, which means that the matrix is square, symmetric, and all its eigenvalues are positive. Theoretically, this is always fulfilled for any correlation matrix. Working with Python however, it is possible to generate so-called “numeric fuzz”, which is the result of round-off errors (Kong et al., 2021). This may lead to eigenvalues extremely close to 0 but still negative. To solve this, the correlation matrix is modified by adding a small constant number to its diagonal (Nocedal & Wright, 2006, pp. 52–54), ensuring positive eigenvalues without changing the correlation between variables. The effect of this procedure can be examined in the tables below. The distribution of each dataset at every step of the procedure can be examined visually in the electronic Appendix.

Algorithm A.1 Monte Carlo Simulation of Credit Scoring Data

```
    Input:
        df: dataset of accepted Credit Scoring applications
        n: number of simulations
Dummify categorical columns
Power transform
Calculate correlation matrix
Modify correlation matrix
Cholesky decompose correlation matrix
Simulate n many random standard normally distributed numbers for each column
of dummified df
Correlate simulation
Back-transformation
De-dummify columns
return df
```

Datasets	Original vs. Transformed	Transformed vs. Cleaned
Small	0.014890	0.004762
German	0.001974	0.001613
Deloitte	0.001726	0.001316
Large	0.003007	0.001493

Table A.1 Correlation Matrix Comparison before Simulation

Number of Simulations	Dataset	Original vs. Simulated & Correlated	Original vs. Simulated, Correlated & Back-Transformed
1,000	**Small**	0.035361	0.049475
	German	0.022821	0.046313
	Deloitte	0.020632	0.047104
	Large	0.019878	0.047679
10,000	**Small**	0.026302	0.036445
	German	0.010628	0.026296
	Deloitte	0.007066	0.025729
	Large	0.006603	0.027398
100,000	**Small**	0.025429	0.026406
	German	0.008570	0.009509
	Deloitte	0.004039	0.005115
	Large	0.002817	0.004369
1,000,000	**Small**	0.027390	0.033771
	German	0.011136	0.021283
	Deloitte	0.007114	0.018356
	Large	0.006242	0.019208

Table A.2 Correlation Matrix Comparison after Simulation

Appendix B: Pseudo Code for Experiment

Algorithm B.1 Reject Inference Loop

```
   Input:
      starters: list of starting datasets
      testers: list of testing datasets, partly enhanced strategies: list
      of strategies
      LR: Logistic Regression learner
for (start, test) in (starters, testers) do
    Split test into 9 rounds X
    Append round to splits
    for strategy in strategies do
        for round in splits do
            split round into X, y
            prepare start, X
            train LR with start
            predict $\widehat{P}(y_i|x_i)$ for $x_i$ in X
            measure statistical performance on $\widehat{P}(y_i|x_i)$
            append statistical measures to metrics
            round $\widehat{P}(y_i|x_i)$ to $\hat{y}_i$
            if strategy = AL strategy then
                select AL instances $\hat{y}_j$ with $j \in I$ where $I = \{i|\hat{y}_i = 1\}$
                change to $\hat{y}_j = 0$
            end if
            measure costs on $\hat{y}$
            append costs to metrics
            if strategy = Benchmark RI then
                select RI instances $\hat{y}_j$ with $j \in I$ where $I = \{i|\hat{y}_i = 1\}$
                change to $y_j = 1$
                $start = start + x_k, y_k + x_j, y_j$ with $k \in K$ where $K = \{k|\hat{y}_k = 0\}$
            else
                $start = start + x_k, y_k$ with $k \in K$ where $K = \{k|\hat{y}_k = 0\}$
            end if
            return metrics, start
        end for
    end for
end for
```

Appendix C: Assessment of Rounds Relevancy

The difference in performance when rounds are used compared to using no rounds at all is assessed in this appendix for every dataset group. When no rounds are used, the models are trained with the same training data as in the experiment, predict $\frac{8}{9}$ of the prediction data, are retrained again and predict the last round of the prediction data, which is the exact same as the last round from the presented experiment.

All Datasets

Last Round

Strategy	Rounds	AUC	PAUC	BS	Cost
No AL	Rounds	0.663588	0.588291	0.237039	0
	No Rounds	0.669739	0.523566	0.234790	0.159962
Benchmark RI	Rounds	0.673432	0.608419	0.246159	-0.019864
	No Rounds	0.673326	0.553418	0.248993	0.108942
Random AL	Rounds	0.675772	0.610117	0.226545	0.033476
	No Rounds	0.680724	0.555446	0.223734	0.213313
ICP Prob	Rounds	0.697176	0.569438	0.297514	-0.110536
	No Rounds	0.705858	0.580862	0.264152	0.087549
ICP Prob Cost	Rounds	0.730537	0.632380	0.164310	0.012153
	No Rounds	0.736775	0.664827	0.170806	0.261009
ICP NN Margin	Rounds	0.355673	0.485258	0.319670	0.454174
	No Rounds	0.344860	0.476150	0.330308	1.176768
ICP NN Margin Cost	Rounds	0.630398	0.567868	0.21254	0.000094
	No Rounds	0.618950	0.550922	0.21930	0.085116

Table C.1 Last Round Assessment for All Datasets

All Rounds Averaged

Strategy	Rounds	AUC	PAUC	BS	Cost
No AL	Rounds	0.666455	0.581019	0.233248	0
	No Rounds	0.630354	0.557191	0.255626	0.37348
Benchmark RI	Rounds	0.666341	0.586833	0.240323	-0.017444
	No Rounds	0.630752	0.560508	0.257204	0.370056
Random AL	Rounds	0.669084	0.569778	0.227486	0.038905
	No Rounds	0.631574	0.560733	0.254397	0.488284
ICP Prob	Rounds	0.691555	0.565021	0.275803	-0.066051
	No Rounds	0.621252	0.545460	0.293684	0.521994
ICP Prob Cost	Rounds	0.712789	0.633543	0.178885	0.034592
	No Rounds	0.624062	0.571006	0.250485	0.996615
ICP NN Margin	Rounds	0.364458	0.474852	0.330618	0.501958
	No Rounds	0.427851	0.492471	0.308442	0.150305
ICP NN Margin Cost	Rounds	0.626094	0.552336	0.219637	0.011683
	No Rounds	0.568989	0.515302	0.254708	0.437186

Table C.2 Average Round Assessment for All Datasets

Original Datasets

Last Round

Strategy	**Rounds**	**AUC**	**PAUC**	**BS**	**Cost**
No AL	Rounds	0.638200	0.580962	0.268225	0
	No Rounds	0.642608	0.485896	0.262515	0.01436
Benchmark RI	Rounds	0.651797	0.611583	0.27986	-0.034942
	No Rounds	0.645182	0.515914	0.28335	-0.028382
Random AL	Rounds	0.657121	0.595380	0.251301	-0.000837
	No Rounds	0.650392	0.519628	0.247254	-0.032663
ICP Prob	Rounds	0.661644	0.541603	0.338593	-0.159398
	No Rounds	0.659810	0.551430	0.311247	-0.129686
ICP Prob Cost	Rounds	0.687623	0.587666	0.15272	-0.290091
	No Rounds	0.684726	0.597475	0.15688	-0.301299
ICP NN Margin	Rounds	0.379952	0.464339	0.315849	0.183818
	No Rounds	0.371811	0.484039	0.320721	0.173447
ICP NN Margin Cost	Rounds	0.603597	0.53957	0.212670	0.000951
	No Rounds	0.596797	0.51346	0.218815	0.118525

Table C.3 Last Round Assessment for Original Datasets

All Rounds Averaged

Strategy	**Rounds**	**AUC**	**PAUC**	**BS**	**Cost**
No AL	Rounds	0.655913	0.565684	0.250197	0
	No Rounds	0.652386	0.560307	0.231754	-0.047322
Benchmark RI	Rounds	0.652631	0.581423	0.261750	-0.032952
	No Rounds	0.652672	0.563642	0.234069	-0.052388
Random AL	Rounds	0.658748	0.555900	0.241575	-0.002153
	No Rounds	0.653251	0.564055	0.230059	-0.024845
ICP Prob	Rounds	0.666183	0.549728	0.298816	-0.125726
	No Rounds	0.639318	0.544510	0.269648	-0.08598
ICP Prob Cost	Rounds	0.682271	0.594231	0.160569	-0.181933
	No Rounds	0.641142	0.574744	0.179467	-0.137212
ICP NN Margin	Rounds	0.382967	0.455381	0.327585	0.22371
	No Rounds	0.410414	0.499340	0.319738	0.16456
ICP NN Margin Cost	Rounds	0.612116	0.527340	0.215355	0.048140
	No Rounds	0.585832	0.513316	0.233241	0.106743

Table C.4 Average Round Assessment for Original Datasets

Lending Club Full Data

Last Round

Strategy	Rounds	AUC	PAUC	BS	Cost
No AL	Rounds	0.900791	0.582420	0.046692	0
	No Rounds	0.909914	0.615092	0.080999	1.673759
Benchmark RI	Rounds	0.904295	0.599005	0.042314	0.184936
	No Rounds	0.909234	0.615206	0.065088	1.368391
Random AL	Rounds	0.903022	0.599747	0.042901	0.515983
	No Rounds	0.912579	0.618234	0.073863	2.462231
ICP Prob	Rounds	0.905427	0.593768	0.039325	0.287562
	No Rounds	0.914932	0.608898	0.067659	2.008125
ICP Prob Cost	Rounds	0.905449	0.646856	0.098442	0.852123
	No Rounds	0.960125	0.871160	0.151088	3.198734
ICP NN Margin	Rounds	0.140240	0.661699	0.356684	2.870222
	No Rounds	0.137295	0.594323	0.426077	10.403157
ICP NN Margin Cost	Rounds	0.868106	0.746252	0.148020	-0.060450
	No Rounds	0.916802	0.786458	0.153482	0.017921

Table C.5 Last Round Assessment for LC_all

All Rounds Averaged

Strategy	Rounds	AUC	PAUC	BS	Cost
No AL	Rounds	0.850440	0.60195	0.111923	0
	No Rounds	0.507154	0.46417	0.461936	4.011298
Benchmark RI	Rounds	0.851600	0.587793	0.103894	0.092408
	No Rounds	0.507078	0.464183	0.460168	4.002144
Random AL	Rounds	0.853536	0.587668	0.104912	0.425888
	No Rounds	0.507450	0.464519	0.461143	4.851356
ICP Prob	Rounds	0.857101	0.576793	0.108311	0.205660
	No Rounds	0.482678	0.446512	0.492334	5.059885
ICP Prob Cost	Rounds	0.853043	0.675299	0.180462	0.784457
	No Rounds	0.487704	0.466053	0.692871	9.496926
ICP NN Margin	Rounds	0.185823	0.577234	0.373880	2.955725
	No Rounds	0.521032	0.492558	0.227534	0.211445
ICP NN Margin Cost	Rounds	0.848196	0.730519	0.189816	-0.309581
	No Rounds	0.466402	0.449845	0.420642	3.621950

Table C.6 Average Round Assessment for LC_all

Monte Carlo Datasets

Last Round

Strategy	Rounds	AUC	PAUC	BS	Cost
No AL	Rounds	0.647124	0.594589	0.239268	0
	No Rounds	0.661218	0.554139	0.227833	-0.058187
Benchmark RI	Rounds	0.651187	0.607963	0.252554	-0.047569
	No Rounds	0.661421	0.562691	0.242457	-0.021815
Random AL	Rounds	0.65589	0.608707	0.232986	-0.005865
	No Rounds	0.68304	0.591189	0.220440	-0.089599
ICP Prob	Rounds	0.706635	0.558173	0.290978	-0.150363
	No Rounds	0.722104	0.565589	0.239465	-0.097218
ICP Prob Cost	Rounds	0.751609	0.650034	0.205113	0.225875
	No Rounds	0.755525	0.673936	0.203001	0.279988
ICP NN Margin	Rounds	0.377944	0.454338	0.310394	0.227694
	No Rounds	0.343842	0.373263	0.323527	0.190425
ICP NN Margin Cost	Rounds	0.610366	0.518449	0.230288	0.023364
	No Rounds	0.588920	0.513442	0.241432	0.081732

Table C.7 Last Round Assessment for MC Datasets

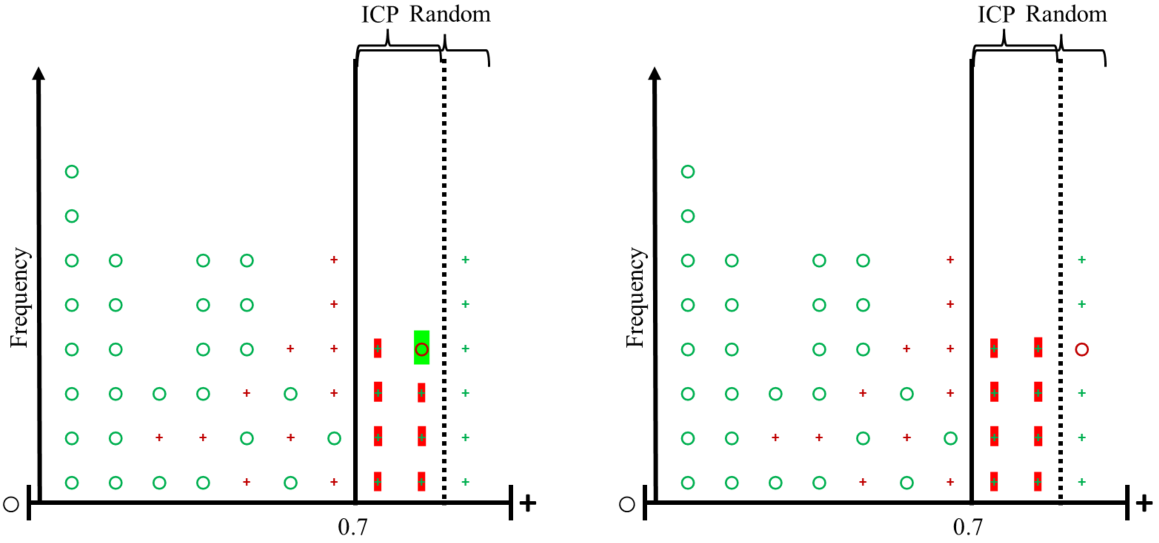
All Rounds Averaged

Strategy	Rounds	AUC	PAUC	BS	Cost
No AL	Rounds	0.643536	0.602438	0.236812	0
	No Rounds	0.635263	0.570586	0.236533	-0.007084
Benchmark RI	Rounds	0.645206	0.600663	0.244623	-0.037327
	No Rounds	0.635285	0.571537	0.238158	-0.003212
Random AL	Rounds	0.650572	0.593873	0.235065	0.001590
	No Rounds	0.637688	0.574703	0.235712	0.052984
ICP Prob	Rounds	0.700264	0.572981	0.270389	-0.062001
	No Rounds	0.626156	0.558600	0.274349	0.191825
ICP Prob Cost	Rounds	0.730182	0.665465	0.211952	0.161024
	No Rounds	0.629086	0.579154	0.246610	0.346187
ICP NN Margin	Rounds	0.377908	0.459338	0.328355	0.253551
	No Rounds	0.424954	0.467088	0.313466	0.157385
ICP NN Margin Cost	Rounds	0.591324	0.531251	0.238723	0.053431
	No Rounds	0.573423	0.522505	0.244000	0.057692

Table C.8 Average Round Assessment for MC Datasets

Appendix D: Discussion of Reversed Cost Imbalance

The datasets used for the presented results were all used with cost functions that always guarantee $CFN < CFP$. It was achieved by adding CFP on top of LGD for CFN. Leaving CFN to be LGD only as economically more reasonable (see Section 3.4), the datasets Small, Large, Small_MC, and Large_MC have a $CFP < CFN$ (see electronic appendix). This makes their cost-efficient threshold $\tau > 0.5$. The effect of a reversed cost imbalance on a somewhat balanced dataset is demonstrated in Figure D.1 and verified by comparing it with the performance of Small_MC (class imbalance 0.4389, threshold 0.581928) with a reversed cost imbalance in accordance with Small_MC (class imbalance 0.4389, threshold 0.3679) as used in the experiment.



Expected PCC of chosen RI instances:
Benchmark RI: 0.93
Random AL: 0.07
ICP Prob (left): 0.13
ICP Prob Cost (right): 0

Figure D.1 RI Scenario class balance, CFP>CFN

With reversed costs, rejecting a GOOD customer is economically riskier than accepting a BAD one. As a result, more applications are given a loan, and more are selected for RI. As most of the applicants are already granted a loan in the first place, the share of repaying customers among the rejections is meager, making *Benchmark RI* a suitable strategy. Whereas *ICP Prob* manages to select applicants better than *Random AL*, *ICP Prob Cost* does the opposite. This is because due to the reversal of cost weights, bads are given a lower score in confidence, as losing a potential customer is now more expensive than a customer defaulting. However, it is still the data of accepted applicants that is gathered, and it is the rejected ones that are subject to RI techniques. For this reason, *ICP Prob Cost* is expected to perform worse than *ICP Prob*.

Strategies	AUC	PAUC	BS	Cost
Benchmark RI	0.741909	0.503747	0.205255	0.153784
Random AL	0.758689	0.585326	0.196619	0.275839
ICP Prob	0.744432	0.548228	0.210513	0.085355
ICP Prob Cost	0.734147	0.473326	0.281454	0.618458

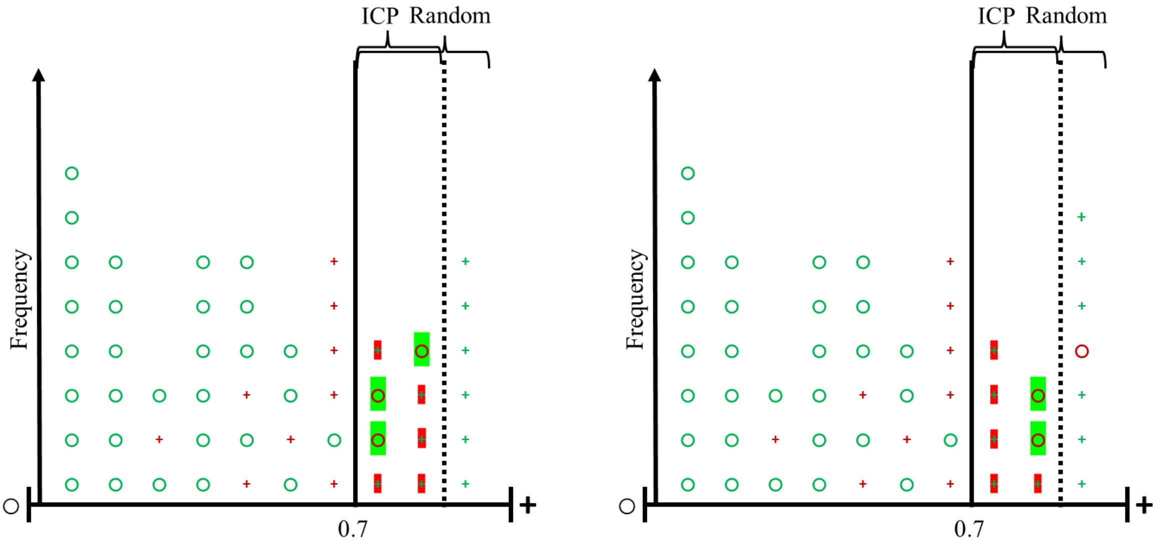
Table D.1 Selected Performance Scores for Small MC with CFP>CFN

As expected, *Benchmark RI* performs better statistically, although not economically (see Table D.1 and D.2). *Random AL* indeed selects RI instances worse in the cost reverse setting than in the standard setting harming economic performance but manages to select more informative applicants. This might be due to the cost reverse threshold being closer to 0.5, which allows to select more informative instances that *No AL* would reject. The same holds for *ICP Prob Cost*. *ICP Prob Cost* also fulfills its expectations as it performs worse in all metrics for the reversed cost imbalance scenario than the standard scenario. Though competitive in the typical scenario, it scores the lowest of all observed strategies. Its high BS (0.281454) and *Cost* (0.618458) confirm the expected performance deviation.

Strategies	AUC	PAUC	BS	Cost
Benchmark RI	0.684804	0.610647	0.249660	-0.080171
Random AL	0.717470	0.600788	0.219768	0.030281
ICP Prob	0.739707	0.516869	0.224728	-0.136192
ICP Prob Cost	0.739504	0.567115	0.202828	-0.008969

Table D.2 Selected Performance Scores for Small MC with CFN>CFP

In addition to a relatively balanced dataset with reverse class imbalance, the appendix also offers the scenario of a class imbalance towards GOODs when $CFN < CFP$. This is the case for Small (see electronic appendix), with $\tau = 0.581928$ compared to the presented Small with $\tau = 0.3679$. Both datasets have a default rate of 0.3127 and are hence imbalanced.



Expected PCC of chosen RI instances:
Benchmark RI: 0.79
Random AL: 0.21
ICP Prob (left): 0.38
ICP Prob Cost (right): 0.25

Figure D.1 RI Scenario GOOD>BAD, CFP>CFN

The class imbalance is expected to improve the AL learner’s performance compared to the class-balanced scenario (see Figure D.2). Indeed, the class imbalance improved the AL learners’ performances in almost all metrics (see Tables D.3 and D.4). While for *Random AL* it is clear, it is not fully clear whether it is the class imbalance alone or the fact that an ICP is used on a different dataset distribution that the ICPs are working better on the original data that they have also been trained on. *ICP Prob* does not show a big difference in performance when the cost imbalance is reversed, given a class imbalance towards GOODs. *ICP Prob Cost* shows underperformance again. That means that a random selection of AL instances works better for a scenario where costs are smaller for the pool containing AL instances, *ICP Prob* performs robustly, and *ICP Prob Cost* worsens. The tendency of *Random AL* performing better than *ICP Prob* and *ICP Prob* performing better than *ICP Prob Cost* when cost imbalance is reversed and even more when class imbalance is higher is also confirmed by comparing the performance of the strategies for Large_MC and Large (see electronic appendix).

The underlying reason is that AL in Credit Scoring intends to select more instances than Passive Learning to mitigate the selection bias. However, if a threshold is chosen that makes the pool of rejections contain mostly BADs and uninformative cases, giving out AL loans is unsuitable for model improvement under an economic budget. In a setting like this, ICPs should perform as intended if their goal is to select risky acceptances that would have been granted a loan but should be reclassified with the consequence of loan rejection for the applicant. This resembles the scenario of Denis & Hebiri (2015), who kick uncertain predictions for labeling them for future use in model training. In their setting however, confident predictions are not rejected for information gathering as it is the case for confident rejections in the Credit Scoring context that causes the selection bias. Not labeling uncertain acceptances would increase the selection bias achieved by conventional Passive Learning, as it is still accepted applications whose information is gathered from labeling rather than rejections.

Strategies	AUC	PAUC	BS	Cost
Benchmark RI	0.729017	0.571692	0.214936	0.175365
Random AL	0.750342	0.589897	0.182347	-0.214557
ICP Prob	0.738217	0.540437	0.222399	-0.199282
ICP Prob Cost	0.739544	0.550686	0.309954	0.421029

Table D.3 Selected Performance Scores for Small with CFP>CFN

Strategies	AUC	PAUC	BS	Cost
Benchmark RI	0.720537	0.614792	0.256073	-0.019045
Random AL	0.730821	0.512167	0.211013	-0.049046
ICP Prob	0.733922	0.539857	0.239147	-0.202224
ICP Prob Cost	0.739979	0.545034	0.184067	-0.215531

Table D.4 Selected Performance Scores for Small with CFN>CFP

Declaration of Academic Honesty

Hiermit erkläre ich, Maximilian Suliga, dass ich die vorliegende Arbeit noch nicht für andere Prüfungen eingereicht habe. Ich habe die Arbeit selbständig verfasst. Sämtliche Quellen einschließlich Internetquellen, die ich unverändert oder abgewandelt wiedergegeben habe, insbesondere Quellen für Texte, Grafiken, Tabellen und Bilder, habe ich als solche kenntlich gemacht. Ich bin mir darüber bewusst, dass bei Verstößen gegen diese Grundsätze ein Verfahren wegen Täuschungsversuchs bzw. Täuschung eingeleitet wird.

I, Maximilian Suliga hereby declare that I have not previously submitted the present work for other examinations. I wrote this work independently. All sources, including sources from the Internet, that I have reproduced in either an unaltered or modified form (particularly sources for texts, graphs, tables and images), have been acknowledged by me as such. I understand that violations of these principles will result in proceedings regarding deception or attempted deception.

Maximilian Suliga

Berlin, June 2, 2023