BANCA D'ITALIA
EUROSISTEMA

Mercati, infrastrutture, sistemi di pagamento

(Markets, Infrastructures, Payment Systems)

Credit Risk Assessment with Stacked Machine Learning

by Francesco Columba, Manuel Cugliari, Stefano Di Virgilio

January 2026

Number 73

BANCA D'ITALIA
EUROSISTEMA

Mercati, infrastrutture, sistemi di pagamento

(Markets, Infrastructures, Payment Systems)

Credit Risk Assessment with Stacked Machine Learning

by Francesco Columba, Manuel Cugliari, Stefano Di Virgilio

Number 73 – January 2026

The papers published in the 'Markets, Infrastructures, Payment Systems' series provide information and analysis on aspects regarding the institutional duties of the Bank of Italy in relation to the monitoring of financial markets and payment systems and the development and management of the corresponding infrastructures in order to foster a better understanding of these issues and stimulate discussion among institutions, economic actors and citizens.

The views expressed in the papers are those of the authors and do not necessarily reflect those of the Bank of Italy.

The series is available online at www.bancaditalia.it.

Printed copies can be requested from the Paolo Baffi Library: richieste.pubblicazioni@bancaditalia.it.

ISSN 2724-6418 (online)
ISSN 2724-640X (print)

Banca d'Italia
Via Nazionale, 91 - 00184 Rome - Italy
+39 06 47921

Designed and printing by the Printing and Publishing Division of the Bank of Italy

CREDIT RISK ASSESSMENT WITH STACKED MACHINE LEARNING

by Francesco Columba*, Manuel Cugliari*, Stefano Di Virgilio*

Abstract

Banca d'Italia's In-house Credit Assessment System (ICAS) for Italian non-financial corporations, used in the Eurosystem's collateral framework for monetary policy implementation, consists of a statistical model (S-ICAS) and of the analysts' evaluation. This paper compares the performance of S-ICAS with that of artificial intelligence, specifically of machine learning (ML) and deep learning models. The findings suggest that deep learning improves discriminative power; decision tree ensembles yield a further improvement, as does a meta-model that stacks random forests, extreme gradient boosting, and deep learning models. Applying eXplainable Artificial Intelligence (XAI) techniques to the meta-model predictions, this paper shows that XAI can support analysts in understanding the key factors behind the differences between ML and S-ICAS predictions, thus helping refine their assessment. While interpretability issues prevent ML-based models from being a full alternative to traditional models, XAI allows for their integration within the overall credit assessment process, thus increasing its effectiveness.

JEL Classification: C52, C55, G24, G32.

Keywords: credit risk, machine learning, deep learning, explainable artificial intelligence.

Sintesi

Il sistema interno della Banca d'Italia per la valutazione del merito creditizio delle imprese non finanziarie (ICAS), utilizzato nel quadro delle garanzie dell'Eurosistema per l'attuazione della politica monetaria, si compone di un modello statistico (S-ICAS) e della valutazione degli analisti. Il lavoro confronta le prestazioni di S-ICAS con quelle dei modelli di intelligenza artificiale (IA), in particolare i modelli di machine learning (ML) e di *deep learning* (reti neurali). I risultati suggeriscono che il *deep learning* migliora la capacità discriminante; gli insiemi di alberi decisionali apportano un ulteriore miglioramento, così come un meta-modello che combina *random forests*, *extreme gradient boosting* e reti neurali. Applicando tecniche di interpretazione dei risultati (XAI) dei modelli alle previsioni del meta-modello, si mostra che queste tecniche possono aiutare gli analisti nella comprensione dei fattori chiave alla base delle differenze tra le previsioni ML e quelle di S-ICAS, contribuendo così a raffinare la loro valutazione. Sebbene i problemi di interpretabilità impediscano ai modelli basati su IA di rappresentare un'alternativa completa ai modelli tradizionali, le tecniche di interpretazione dei risultati consentono l'integrazione nel processo complessivo di valutazione del merito di credito, aumentandone così l'efficacia.

* Financial Risk Management Directorate.

INDEX

1. Introduction[1]

Since 2013, Banca d'Italia has been managing the In-house credit assessment system (ICAS) to assess the creditworthiness of Italian non-financial firms within the Eurosystem credit assessment framework (ECAF).[2] ICAS is, in fact, one of the methodologies used to evaluate the eligibility of assets pledged in monetary policy operations (Giovannelli *et al*., 2023).

The ICAS rating process follows a two-stage procedure. In the first stage, a statistical model (S-ICAS) generates a one-year probability of default[3] (PD) for each non-financial firm. The second stage involves an expert assessment carried out by at least two financial analysts, who examine additional quantitative and qualitative information – typically a broad set of credit risk drivers –, after which the statistical PD can be either confirmed or revised. Currently, the S-ICAS is applied to approximately 370,000 Italian non-financial firms on a monthly basis, while the second stage is applied to a subset of the approximately 4,000 firms most relevant for ICAS each year.

S-ICAS is divided in two components: a financial component, which draws upon indicators derived from firms' financial statements, as provided by Cerved Group, and a credit behaviour component, which uses indicators derived from the National Credit Register (NCR). In turn, both components consist of sub-models, mainly in relation to sectors of activity and firm size (as explained in more detail in Appendix 1). Both the financial and the credit behaviour components generate a one-year PD, referred to as the financial PD and credit behaviour PD, respectively. These probabilities are subsequently integrated through a third step (performed with four sub-models by firm size), which yields the statistical PD. All the components of S-ICAS are estimated by means of logistic regressions (Narizzano *et al*., 2024).

This study compares the performance of widely used machine learning (ML) methods to the performance of S-ICAS and discusses how differences between the outcome obtained with the former and those of S-ICAS can be treated to improve the quality of the ICAS second stage, and thus the overall assessment. We contribute to the literature in two ways. First, while earlier studies compare ML techniques with statistical methods mostly based on a model and a single dataset, we carry out this comparison with a model, S-ICAS, that exploits multiple datasets, a multistep estimation process and that has been employed for monetary policy purposes for over a decade, with a consistently high performance. Second, unlike previous studies in which model explainability is discussed mainly as a theoretical exercise, we propose an application of eXplainable Artificial Intelligence (XAI) methods that can be used by financial analysts in the second stage of the ICAS evaluation process.

---

[1] The authors would like to thank Paolo Del Giovane, Paolo Parlamento, Antonio Scalia and an anonymous referee for useful comments and suggestions.

[2] https://www.ecb.europa.eu/mopo/coll/risk/ecaf/html/index.en.html.

[3] See Appendix 1 for the definition of ICAS default.

This analysis is motivated by the significant advancements in machine learning observed in recent years – thanks to the availability of large datasets and the improvements in computational capabilities – and by the fact that ML methods, being intrinsically different from traditional statistical approaches, have the potential to provide useful results to enhance the accuracy and robustness of our assessment of firms' creditworthiness. Classical techniques, such as logistic regression, rely on assumptions about the underlying stochastic process generating the observed data; the model's parameters are estimated from the data and subsequently used for population inference or statistical testing. In contrast, ML methods make minimal or no assumptions about the data-generating process; instead, they identify the optimal function that maps input variables into the target variable, prioritizing maximum predictive accuracy (Breiman, 2001b). These methods frequently outperform traditional statistical models (Barboza *et al.*, 2017) as they are capable of capturing complex, non-linear and non-monotonic relationships with the dependent variable (Moscatelli *et al*., 2020).

One important aspect, however, is that due to the non-parametric or complex functional form, explaining ML predictions remains a challenge (Loyola-Gonzalez, 2019). As a result, their application is often restricted, particularly in regulatory environments where explainability is crucial (Cascarino *et al*., 2022). In contrast, most traditional statistical models – among which S-ICAS – have simpler structures and their output can be easily interpreted by financial analysts. To address this issue, a sub-field of artificial intelligence, XAI, has been developed to enhance both the explainability and the interpretability of ML models (Štrumbelj and Kononenko, 2014; Ribeiro *et al.*, 2016; Apley and Zhu, 2020).[4] In this paper we explore whether XAI developments can be leveraged to support the analysis conducted within the ICAS framework.

In the first part of our empirical analysis, we replicate the structure of S-ICAS by training three different ML models – random forests, extreme gradient boosting, and deep learning – and assess whether these techniques can achieve significant improvements in discriminatory power compared to S-ICAS. To further enhance performance and robustness, we also develop a meta-model that, by stacking the three models, integrates their predictions. We then use XAI techniques to identify which factors, among those considered in the models, are more relevant in explaining the discrepancies between the PDs estimated by ML models and those produced by S-ICAS.

Our first finding is that ML models provide useful results for S-ICAS, which complement but cannot substitute it, due to their complexity, that prevents corporate credit analysis from relying only on ML to

[4] A model is deemed explainable if it allows a person to understand either its internal mechanisms or the rationale behind its outputs. In the first case, the model is also called interpretable (European Banking Authority, 2020).

fully characterize firms' creditworthiness. This result aligns with warnings against overreliance on automated decisions when applying artificial intelligence (Narayanan and Kapoor, 2024), as human oversight remains essential (Angelini, 2025). This challenge is a source of concern also in banking regulation and supervision (European Banking Authority, 2023), which care for transparency and accountability of models. The XAI techniques, in this respect, are valuable and may streamline the analysts' work, enabling them to pay attention to the most problematic issues, specifically where ML models and S-ICAS provide diverging signals (although caution is needed, since not all the cases are easily explainable). Our second finding is that ML models are particularly valuable in periods in which the quality of information deteriorates, as they are more accurate and faster to adapt to changes in economic developments, as shown for the pandemic crisis period. We also have indications that the ML models may lead to correction of possible biases in the assessments of S-ICAS and of the analysts, though we leave for future work a fully-fledged evaluation of such possibility. All in all, these results suggest that the use of ML models may increase the efficiency and quality of the whole ICAS process, offering insights for the calibration of its output and making it more robust, also due to the possibility of integrating additional information from unstructured databases.

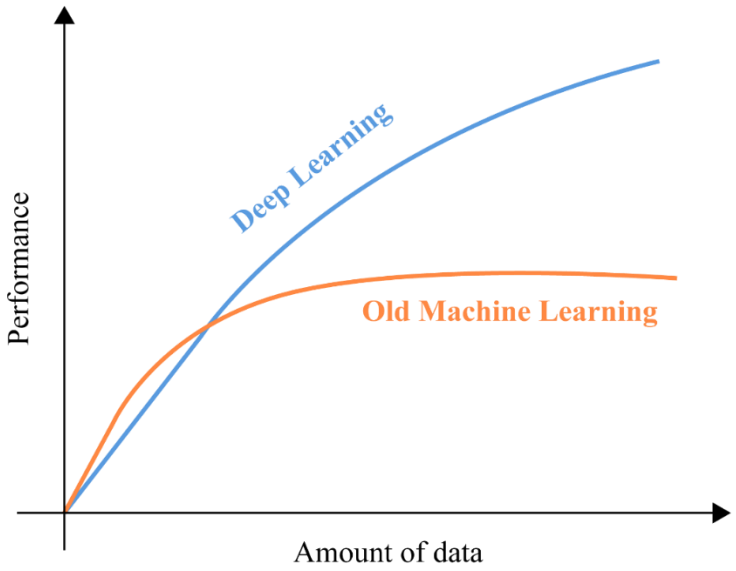
The remainder of the paper is structured as follows. Section 2 reviews the literature related to the use of ML techniques for credit risk and of the XAI methods. Section 3 outlines the dataset and the methods employed to estimate the ML models. Section 4 presents the main results of the models. Section 5 describes the robustness checks. Section 6 introduces the application of XAI methods developed to explain the differences in the PDs produced by the various models. Section 7 concludes.

2. Machine learning for credit risk and XAI methods

The literature on credit risk indicates that ML techniques frequently outperform traditional statistical models in terms of accuracy and discriminatory power (Shi *et al*., 2022). Several studies have compared machine learning models — especially ensemble methods, which use various learning algorithms, such as random forests and gradient boosting machines — with traditional techniques like logistic regression. These comparisons show that ML models may outperform traditional models in terms of discriminatory power and/or accuracy when predicting corporate failures and bankruptcy (Li and Wang, 2017; Moscatelli *et al*., 2020; Schalck and Yankol-Schalck, 2021).

Since the early 2010s, deep learning models have gained prominence in the ML literature (Alom *et al*., 2019; Dell, 2024), due to the increasing size of datasets and the advancements in computational power. Indeed, deep learning methods tend to outperform other ML techniques when the dataset is very large (Fig. 1).

Figure 1 – Deep learning vs other ML techniques



Source: Alom *et al.* (2019).

For these reasons, some authors have started to train models using also deep learning — which use artificial neural networks — to predict corporates' default and financial distress (Ciampi and Gordini, 2013; Barboza *et al*., 2017; Petropoulos *et al*., 2019; Gregova *et al*., 2020; Zhang *et al*., 2022). The performance of ML models, including deep learning, is generally superior to that of traditional models according to a number of key metrics, particularly in terms of accuracy and Area Under the Receiver Operating Characteristic (AuROC).[5] However, when comparing deep learning to other ML techniques, the results are mixed, without a clear best performer. In the aforementioned studies, the comparison between ML and traditional statistical methods has often relied on relatively simple models constructed specifically for this purpose.

In recent years, global and local XAI techniques have also been used to improve the interpretability and the explainability of machine learning models. Global XAI methods provide insights into the model as a whole. Notable examples include the permutation variable importance method (Breiman, 2001a; Altmann *et al*., 2010) and accumulated local effects (ALE) plots (Apley and Zhu, 2020). The permutation variable importance method identifies the contribution of each variable to the model's discriminatory power, while

[5] The Area Under the Receiver Operating Characteristic is a performance measurement for classification problems at various threshold settings. The AuROC represents how well the model distinguishes between classes. A higher AuROC value means better model performance, with a value of 1.0 representing a perfect model and a value of 0.5 representing a model with no discriminatory power.

ALE plots illustrate the average relationship between the independent variables and the model's predictions, making it possible to detect non-linear or non-monotonic effects.

Conversely, local XAI methods aim to explain single predictions. Examples are the 'Shapley values' (Štrumbelj and Kononenko, 2014; Lundberg and Lee, 2017a; Lundberg and Lee, 2017b; Lundberg *et al.*, 2020) and 'Local interpretable model-agnostic explanations' (LIME) (Ribeiro *et al.*, 2016). Shapley values, derived from game theory (Shapley, 1953), are grounded in robust theoretical properties (see Section 6) and are used to fairly distribute the difference between a model's prediction and the average prediction among the independent variables. LIME approximates a non-linear model with a linear one, in the neighbourhood of a specific prediction. However, LIME relies on a heuristic procedure that contains different sources of uncertainty (Zhang *et al.*, 2019).

Explaining the output of an ML model is crucial in credit risk assessment, as highlighted by Cascarino *et al.* (2022). Specifically, in the case of corporate default forecasting, it is essential to link a given prediction to the firm's characteristics, particularly when an automated scoring procedure directly influences the firm's ability to access credit. Furthermore, in the case of ICAS, since the output of the statistical model serves as input of the expert assessment, it is imperative to explain statistical PDs to credit analysts.

In recent years, XAI techniques have increasingly been applied to explain ML models trained to assess companies' credit risk (Bussmann *et al.*, 2021; Cascarino *et al.*, 2022; Bitetto *et al.*, 2023). In these studies, since the objective is to show how these techniques can be used to improve the models' explainability, the exercises performed by the authors are relatively simple.

3. Data and methods

3.1 Dataset

The dataset is assembled using Banca d'Italia's data and extends from 2014 to 2023; financial statement and credit behaviour indicators are built using the methodology described in Narizzano *et al.* (2024). We consider the data from 2020 and 2021 to be significantly different from the other, as these years were heavily influenced by extraordinary government measures aimed at supporting businesses during the Covid-19 pandemic.[6] These measures had an impact on the default rates recorded in those years. Consequently, the data from 2020 and 2021 are not used for training ML models and are considered only

[6] The Italian government implemented several economic measures to support Italian companies, following the outbreak of Covid-19 pandemic. In particular, the government introduced the possibility of benefiting from payment delays ('debt moratorium') and public guarantees for bank loans. These measures increased the amount of liquidity available to companies during the pandemic (see D.L. 'Cura Italia' and D.L. 'Sostegni').

in the robustness checks. The dataset contains about 2.5 million observations. The trend of the default rate is generally decreasing over the sample period (Table 1).

Table 1 - Default rates and firms

(percentages; units)

Year	Default rate	Firms
2014	6.2	237,109
2015	5.3	229,134
2016	4.3	252,740
2017	3.2	260,061
2018	2.9	259,768
2019	2.7	260,985
2020	2.1	261,328
2021	1.7	246,287
2022	1.9	245,972
2023	2.0	278,186

As expected, the number of companies in default is much lower than the total, which is typical when dealing with corporate insolvencies. Consequently, the dataset is unbalanced. In the ML literature, unbalanced datasets are often addressed through rebalancing techniques, such as undersampling (He and Garcia, 2009). Undersampling involves randomly removing observations from the majority class (in this case, companies not in default) until the dataset achieves an approximate balance. Although research has shown that undersampling does not always improve the performance of ML models (Dal Pozzolo *et al*., 2015), we decide to use this technique for computational reasons.[7]

The dataset is also divided into a training set and a test set[8] for the estimation and evaluation of the models, as follows. The years 2014 and 2023, first and last year of analysis, are designated as the test set, allowing us to test the models in a year with a higher default rate (2014) and in a year with a lower default rate (2023). Both years are marked by challenging economic conditions:[9] during 2014, the economic cycle was still affected by the aftermath of the sovereign debt crisis, while 2023 was affected by the surge of inflation in 2021-2022 and the subsequent sharp rise in interest rates. Excluding the Covid-19 pandemic, the central years (2015-2019 and 2022) are used as the training sample. This temporal split ensures that the model is tested on an out-of-sample set from distinct economic periods.

[7] Under-sampling greatly reduces the size of the dataset, speeding up the training-validation cycle of ML models.
[8] Under-sampling was only applied to the training set.
[9] See "Banca d'Italia Annual Report 2014" and "Banca d'Italia Annual Report 2023".

3.2 Random forests

Random Forests (RFs) are a combination of individual decision trees (Breiman, 2001a) which improves the overall predictive performance and robustness. RFs may handle large, high-dimensional datasets and model complex, non-linear relationships. In the context of credit risk, where the relationship between predictors and default are influenced by numerous factors, RFs mitigate overfitting by averaging multiple decision trees, thus providing more stable and accurate predictions. Additionally, RFs can be used as a feature importance ranking method, to rank the financial and behavioural indicators for credit risk, making RFs a powerful tool not only for prediction, but also for interpretation.[10]

The methodology used to estimate the RF model for predicting PDs involves several steps (see Appendix 2). The pre-processing phase involves constructing sub-datasets by selecting variables of interest for each of the eleven financial statement models and three credit performance models.[11] Then, for each sub-model, a parameter space is defined for the hyper-parameter optimization process (Table 2).

Table 2 – Random forests hyper-parameters search space

Parameter	Type	Range
n_estimators	Integer	100 ÷ 500
min_samples_leaf	Integer	1 ÷ 1000
max_features	Categorical	[sqrt, log2, None]
bootstrap	Categorical	[True, False]
criterion	Categorical	[gini, entropy]
n_iter	Integer	200
cv	Integer	10

Note: *n_estimators* is the number of trees in the forest; *min_samples_leaf*, the minimum number of samples required to be at a leaf node; *max_features*, the number of features to consider when looking for the best split; *criterion* measures the quality of a split, with 'gini' for the Gini impurity and 'entropy' for the information gain; *n_iter*, is the number of iterations for the hyper-parameter search; *cv* is the number of folds.

[10] RFs operate by constructing multiple decision trees during the training phase and then predicting the outcome by aggregating the results from all trees. In classification tasks, where the goal is to assign observations to discrete classes, the model aggregates the predictions across the trees to determine the final class. In regression tasks, where the objective is to predict a continuous value, the model combines predictions by averaging the output from all trees to generate a final, more stable prediction. Each tree is trained on a different bootstrap sample of the original data, and at each split in the tree, a random subset of features is considered for splitting. This feature generally results in better performance and reduced susceptibility to overfitting compared to individual decision trees (Louppe *et al.*, 2013).

[11] These eleven financial statement models reflect the structure and variables of S-ICAS, encompassing five macroeconomic sectors (industrial, trade, construction, services, and real estate) across two distinct types of financial statements (ordinary and simplified), plus an additional model for holding companies. Distinct sub-datasets are created by extracting relevant features for each sector and financial statement type. Similarly, three credit models are constructed to capture firms' credit behaviour performance.

The training dataset is then split into multiple subsets, known as folds, and cross-validation[12] is used for hyper-parameter optimization (Kohavi, 1995) to evaluate the model generalizability and mitigate overfitting. By evaluating the model performance across multiple folds, we enhance the model predictive accuracy and robustness. The cross-validation process not only assesses the model ability to generalize to unseen data, but it also plays a critical role in hyper-parameter tuning, that is critical to enhance the model performance.[13] Once the optimal hyper-parameters are identified, the final RF model is trained using the entire training dataset and subsequently evaluated on the test set to assess its performance.

3.3 Extreme gradient boosting

Extreme Gradient Boosting (XGB or XGBoost) is a widely recognized ML technique (Chen and Guestrin, 2016) that has been successfully applied to various predictive modelling problems, including the prediction of corporate default (Petropoulos *et al.*, 2019; Schalck and Yankol-Schalck, 2021). XGB represents an advanced implementation of Gradient Boosting (GB), a ML technique that builds predictive models by sequentially training decision trees, where each new tree corrects the errors made by the previous ones, to optimize both computational speed and model performance. XGB aggregates the outputs of multiple weak learners, typically shallow decision trees, to form a single, more accurate predictive model. The technique incorporates advanced regularization methods to reduce overfitting and employs innovative algorithms that allow for faster training and better scalability, making it a more robust and efficient version of GB. We use XGB to predict the PD by leveraging financial and credit behavioural indicators.

The optimization process is carried out using a ten-fold cross-validation strategy for each of the sub-models to ensure robust hyper-parameter selection (Table 3). This method helps in assessing the model performance across different subsets of the data, thus providing a better generalization.

[12] Cross-validation implies systematically training the model on a combination of the folds, while validating it on the remaining one.

[13] We employ Bayesian optimization to explore systematically the hyper-parameter space and identify the optimal combination of parameters that maximize the model's AuROC on the validation data.

Table 3 – Extreme Gradient Boosting hyper-parameters search space

Parameter	Type	Range
n_estimators	Integer	50 ÷ 500
learning_rate	Real	0.005 ÷ 0.3
max_depth	Integer	1 ÷ 10
reg_alpha	Real	0 ÷ 500
reg_lambda	Real	0 ÷ 500
n_iter	Integer	200
cv	Integer	10

Note: *n_estimators* is the number of trees in the forest; *learning_rate* controls the contribution of each tree to the final model, balancing model complexity and learning speed; *max_depth* determines the maximum depth of each tree and helps to control overfitting by limiting the tree's complexity; *reg_alpha* is the L1, or Lasso, regularization term on weights, which adds a penalty to the absolute values of the weights to encourage sparsity and prevents overfitting; *reg_lambda* is the L2, or Ridge, regularization term on weights, which adds a penalty to the squared values of the weights to prevent overfitting by discouraging large coefficients; *n_iter*, is the number of iterations for the hyper-parameter search; *cv* is the number of folds.

The hyper-parameters selected are those that maximize the AuROC on the validation sets. Once the optimal hyper-parameters are identified, the final XGB model is trained on the entire training set and evaluated on the test set. The model ability to rank the predicted probabilities effectively is visualized using the ROC curve, which represents the true positive rate against the false positive rate at various threshold settings.

An important feature of XGB is the robustness to overfitting, which is achieved through the combination of boosting and regularization techniques. In XGB, this robustness is further enhanced by the regularization parameters that add penalties for large coefficients, avoiding the risk that the model becomes overly complex and thus reducing the risk of overfitting. By controlling the size of the coefficients, regularization ensures that the model remains generalizable to new, unseen data, striking a balance between fitting the training data and maintaining predictive power on the test set.

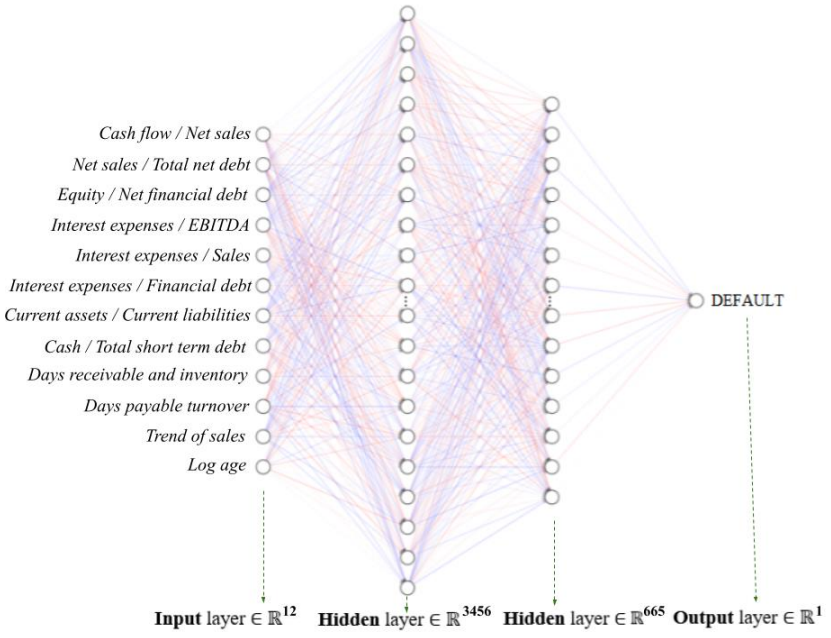
3.4 Deep learning

Deep learning uses artificial neural networks to perform complex computations on large amounts of data. This type of ML, which resembles the structure of the human brain, have emerged as a powerful tool in credit risk forecast. Neural networks consist of layers of interconnected nodes, or neurons, that process input data. Each neuron receives inputs, performs a computation, and passes the result to the next layer. These networks typically include an input layer, multiple hidden layers, and an output layer. In the hidden layers, neurons apply weights to the inputs and pass them through an activation function to introduce non-linearity, allowing the network to learn complex patterns. The output layer generates the final predictions. Deep learning arises when multiple hidden layers are employed within a neural network. As the number of layers increases, the network gains the ability to detect more intricate patterns and complex relationships within the data.

In classification tasks, such as credit risk assessment, neural networks categorize input data into predefined classes. The input features - such as financial ratios, payment records - are fed to the network. The network processes these features through its layers, adjusting the weights during training to minimize the error between its predictions and the actual outcomes. The final output is a probability score that indicates the likelihood of a firm defaulting. By leveraging multiple hidden layers, deep learning models can reveal hidden relationships in financial datasets, allowing for the estimation of complex, non-linear relationships (LeCun *et al.*, 2015). This capability arises from the evolution of the weights that connect the neurons in the hidden layers. As the model is trained, these weights are adjusted through back-propagation, enabling the network to learn and model the non-linear relationships present in financial data. This approach is particularly advantageous in assessing credit risk, as it allows the model to uncover patterns that traditional linear models may overlook.

In applying deep learning to firm default prediction, the architecture of the model is crucial. The designed network consists in multiple layers, each serving a specific role in transforming the input data into a meaningful output that reflects the probability of default. The sub-model are implemented using *Keras,*[14] a high-level neural networks library that simplifies the construction and training of deep learning models. The architecture includes dense and fully connected layers with varying numbers of neurons, activation functions to introduce non-linearity, and dropout layers for regularization.

Figure 2 – Industry ordinary balance sheet neural network



As an example, Figure 2 illustrates the neural network architecture used to estimate the sub-model of ordinary balance sheets within the industrial macro-sector. This architecture comprises several layers

[14] *Keras* is an open-source software library that provides a Python interface for artificial neural networks.

(input, hidden, output), depicted at the bottom with their vector dimensions, each serving a specific function in the network overall performance. The input layer is composed of multiple nodes, each corresponding to an input variable relevant to credit analysis.[15] The network ability to generalize from the training to unseen data – i.e. data not included in the training set - is crucial for achieving reliable predictions. This capability heavily depends on configuring the hidden layers effectively, as their structure and parameters enable the network to capture underlying data patterns. This generalization allows the model to accurately assess the default risk of firms that it has not encountered during training, ensuring its robustness across scenarios. For representation purposes, the number of neurons in each hidden layer is displayed, highlighting the architecture complexity and capacity for learning intricate patterns. The final layer in the network is the output layer, which consists of a single neuron that produces the prediction of default. This output is a probability score indicating the likelihood that a firm will default based on the input variables processed through the network.

Regularization techniques are essential in neural network training to mitigate overfitting, a common issue where the model performs well on the training data, but poorly on the test data (see Appendix 2). As for RFs and XGB, a thorough hyper-parameter search is conducted for this model (Table 4).

Table 4 – Neural network hyper-parameters search space

Parameter	Type	Range
dense_1_units	Integer	2048 ÷ 6144
dense_2_units	Integer	2048 ÷ 6144
dense_1_dropout	Real	0.01 ÷ 0.99
dense_2_dropout	Real	0.01 ÷ 0.99
learning_rate	Real	$1 \times 10^{-4} \div 5 \times 10^{-1}$
reg_strength	Real	$1 \times 10^{-6} \div 1 \times 10^{-3}$
activations	Categorical	relu, selu, elu, tanh, swish[16]
batch_size	Integer	32 ÷ 64

Note: the number of neurons in each layer are *dense_x_units*; the dropout rates are *dense_x_dropout*, for the hidden layers, the strength of regularization is *reg_strength*. *Batch_size* significantly affects the model performance and training stability

[15] See Narizzano *et al.* (2024) for a detailed definition of the financial and credit indicators used across the various models.
[16] Activation functions (Dubey et al., 2022) are crucial, allowing the network to learn complex patterns. The *ReLU* (Rectified Linear Unit) function helps to mitigate the vanishing gradient problem, *i.e.* when in neural network training gradients become extremely small, preventing weights from updating effectively. The *SELU* (Scaled Exponential Linear Unit) function automatically normalizes activations across layers, leading to faster and more stable training. *ELU* (Exponential Linear Unit) introduces smoothness by producing negative values when the input is below zero, helping the model to capture more nuanced patterns. The *Tanh* function maps input values into a range between -1 and 1, often leading to better convergence. *Swish*, provides smoother transitions and potentially leading to higher accuracy by allowing the network to retain more information during training.

helping to prevent overfitting, without compromising the efficiency of computation and the speed of convergence. The optimal combination of hyper-parameters is searched with Bayesian optimization, facilitated by the KerasTuner library.

Hyper-parameter optimization through cross-validation is essential in deep learning (see Appendix 2), as in all ML models. To further improve model training, *early stopping* and *reduce learning rate on plateau* techniques are used.[17] We also note that the computational demands of training multiple deep learning models necessitate the use of Graphics Processing Units (GPUs).[18]

Finally, although the AuROC of the deep learning model on the test set may be slightly lower than traditional models like RF and XGB, its inclusion in a stacked ensemble model boosts the overall performance (see Section 4). The ensemble leverages the strengths of each model, resulting in better generalization, meaning that the model performs well not only on training data, but also on new, unseen data, reducing the risk of overfitting.

3.5 Stacked model

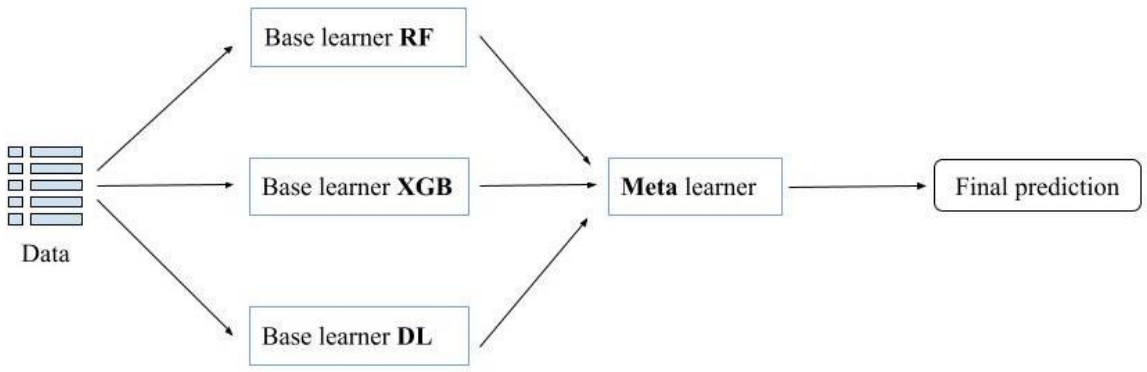
Model stacking (Wolpert, 1992) is a powerful ensemble learning technique that combines multiple machine learning models to produce a single, more accurate predictive model. By integrating various models, stacking leverages the strengths of each model, thereby mitigating their weaknesses and enhancing the overall performance. In stacking, the base models are trained on the training data, then a meta-model is trained to make the final predictions based on the outputs of these base models.[19] This approach captures a diverse range of patterns and relationships within the data, which might be missed by a single model, thus improving generalization and robustness.

The benefits of stacking are manifold (Dietterich, 2000; Alexandropoulos *et al*., 2019). First, it improves predictive accuracy by incorporating the unique strengths of different models. In this study, the final model for predicting firms' defaults is derived through the stacking of RFs, XGB, and deep learning models. While tree-based models, such as RFs and XGB, are best suited for structured data and the

[17] The first one monitors the validation loss and halts training when the loss stops to decrease, preventing overfitting. The second technique reduces the learning rate when the validation loss levels off or plateaus. This adjustment allows the model to fine-tune its weights with greater precision, adapting to smaller gradients as it approaches an optimal solution.
[18] A GPU (Graphics Processing Unit) is a specialized processor designed to accelerate the rendering of images and video, but it has become essential in deep learning due to its ability to perform parallel processing on large datasets. Unlike CPUs, which are optimized for sequential tasks, GPUs can execute thousands of operations simultaneously, making them highly efficient for training neural networks, especially when dealing with complex models and large-scale data. GPUs, such as the NVIDIA® GeForce RTX™ 4080, are designed to handle the parallel processing required for training deep learning models efficiently. The NVIDIA® GeForce RTX™ 4080 is a powerful graphics processing unit (GPU) designed for high-performance tasks like gaming, deep learning, and complex computations. Based on NVIDIA's Ampere architecture, it provides advanced features like ray tracing and AI-enhanced graphics. With its improved tensor cores and support for technologies such as DLSS (Deep Learning Super Sampling), the RTX 4080 is particularly well-suited for speeding up deep learning model training and other intensive workloads. In this project, leveraging GPU capabilities facilitated the training of sub-models across various balance sheet and credit rating models, with the entire process taking approximately 12 days.
[19] In order to train the meta-model, the output of base models is obtained through a cross-validation procedure in order to avoid overfitting (Breiman, 1996).

identification of non-linear relationships, deep learning models excel at capturing complex patterns and interactions within the data (Shwartz-Ziv and Armon, 2022). By combining these models, stacking ensures that the final prediction is well-rounded and robust. Second, stacking can help to reduce the risk of overfitting by allowing a meta-model to learn how to optimally combine the predictions from multiple base models. This is achieved by training the meta-model on a validation set, where it assesses the strengths and weaknesses of each base model predictions. By assigning appropriate weights to the base models according to their performance, the meta-model effectively balances their individual biases and variances. This process results in a more generalized and robust final prediction, minimizing the overfitting risk that any single model might introduce.

Figure 3 – ICAS stacked machine learning model for predicting firms' PDs



The stacked model process begins with the training of the base models - RFs, XGB, and the deep learning - on the dataset. Each model generates its predictions, which serve as input features for the meta-model. The meta-model, trained with logistic regression on these predictions, is responsible for making the final decision. By learning the optimal way to combine the base model predictions, the meta-model effectively enhances the overall performance. The inclusion of deep learning in the stacking ensemble is noteworthy. While the AuROC of the deep learning model on the test set may slightly lag behind the RF and XGB models individually, its contribution to the stacked model is significant. The deep learning model unveils complex dependencies and interactions that the other models might overlook, thus enriching the feature set for the meta-model and leading to more accurate predictions.

4. Models' results

The S-ICAS currently used by Banca d'Italia has been trained on a different dataset than the one used in this work. To properly compare the performance of the models, we decided to train not only the ML models, but also to re-train S-ICAS, using the same dataset described in the previous section. The

performance of the models is assessed with AuROC. In the following sub-sections, we report the results for each of the three main components of S-ICAS.

4.1 Financial component

We show the aggregated[20] AuROC of ML models and that of S-ICAS financial component (Table 5).

Table 5 – AuROC for the financial component

Year	S-ICAS	Random forests	XGBoost	Deep learning	Stacked
2014	0.737	0.760	0.761	0.757	0.764
2023	0.755	0.786	0.788	0.782	0.791

We first note that, in general, the models perform better in 2023. This may depend on the difference in the percentage of ordinary balance sheets available in those two years. While for 2014 the number of available ordinary balance sheets was less than half the number of simplified financial statements, for 2023 the situation was reversed. This is important because models estimated on ordinary balance sheets are superior, in terms of AuROC, to corresponding models estimated on simplified financial statements (see Appendix 3). This was expected since ordinary financial statements contain more information than simplified financial statements.

In comparing the discriminatory power of the methods, we observe that the performances of RF and XGBoost are very similar, and superior to that of the S-ICAS financial component, with an approximate 2.3 to 3.3 percentage point increase in AuROC, depending on the year. Deep learning also performs better than S-ICAS, but the AuROC increments are slightly less pronounced.

To further enhance the discriminatory power, we integrate the predictions obtained from all three ML methods using the stacking method. The discriminatory power of the meta-model is slightly superior to that of RF and XGBoost (almost 0.5 percentage points), showing the potential of combining different ML approaches to achieve improved performance, and illustrating the value of ensemble and stacking techniques in ML. [21]

[20] The performance of ML models on each one of the eleven sub-component of S-ICAS financial module are reported in Appendix 3.

[21] We conducted a bootstrap analysis to verify the statistical significance of these results with 95 per cent confidence intervals for the difference between the AuROC of ML models and of S-ICAS (Appendix 4, Table A.7).

4.2 Credit behaviour component

We show the aggregated[22] AuROC of ML models and that of S-ICAS credit behaviour component (Table 6).

Table 6 – AuROC for the credit behaviour component

Year	S-ICAS	Random forests	XGBoost	Deep learning	Stacked
2014	0.864	0.868	0.870	0.869	0.870
2023	0.823	0.835	0.838	0.829	0.839

First, we observe that AuROC values are far greater than the ones observed in Table 5. This was expected since the credit behaviour component is estimated using high quality NCR data, that are also available with a shorter delay than it is the case for financial statement data. Second, we note that ML models still outperform S-ICAS, but the increments in AuROC, between 0.5 and 1.5 percentage points, are less pronounced than the ones obtained for the financial component, in line with the findings of Moscatelli *et al*. (2020). Third, the disruptions caused by the Covid-19 pandemic blurred the information conveyed by firms' credit and financial indicators, also due to the measures which supported firms' liquidity, denting models' accuracy after the pandemic. Lastly, we observe that the model built using the stacking method has the same performance or slightly outperforms each one of the base models[23], but the increments in AuROC are less noticeable than the ones obtained in the financial component.

4.3 Complete model

In this section, we present the results of the integrated models (the financial component merged with the credit behaviour component) in terms of discriminatory power. Since the best ML model is the one obtained using stacking, both for the financial and the credit behaviour component, we show the comparison between the aggregated AuROC of the meta-model and that of S-ICAS (Table 7).

Table 7 – Comparison between AuROC of complete models

Year	S-ICAS	Stacked
2014	0.874	0.880
2023	0.854	0.873

As expected, the two complete models outperform both the corresponding financial and credit behaviour component in terms of AuROC. We also note that the meta-model outperforms S-ICAS in 2014 (by 0.6

[22] The performance of ML models on each one of three sub-components of S-ICAS credit behaviour module are reported in Appendix 3.

[23] We conducted a bootstrap analysis to verify the statistical significance of these results with 95 per cent confidence intervals for the difference between the AuROC of ML models and of S-ICAS (Appendix 4, Table A.8).

percentage points), and especially in 2023 (by 1.9 percentage points). We believe that this finding depends on a greater robustness of the meta-model. The aftermath of the Covid pandemic, as noted in the previous sub-section, suggests that the meta-model may be more resilient to the challenges posed to models by the decrease in the precision of credit quality information caused by the disruption to the economic activity and the associated mitigating measures.

5. Robustness

To understand if the last finding derives from the greater stability of the meta-model, we report the performance of the two models during the Covid period, 2020-2021 (Table 8).

Table 8 – Comparison between AuROC of complete models with Covid period

Year	S-ICAS	Stacked
2014	0.874	0.880
2020	0.872	0.885
2021	0.838	0.852
2023	0.854	0.873

The performance gap (in terms of AuROC) between the meta-model and S-ICAS widens over the years: in 2014, the difference is 0.6 percentage points; in 2020, at the onset of the Covid crisis, the gap increases to 1.3 percentage points; and in 2021, it reaches 1.4 percentage points. Moreover, in 2023, in the aftermath of the pandemic, even if there are still effects of the Covid-19 crisis on the variables used by the models[24], the performance of both models improves. However, the difference between the two AuROC grows further, and it is equal to 1.9 percentage points.[25]

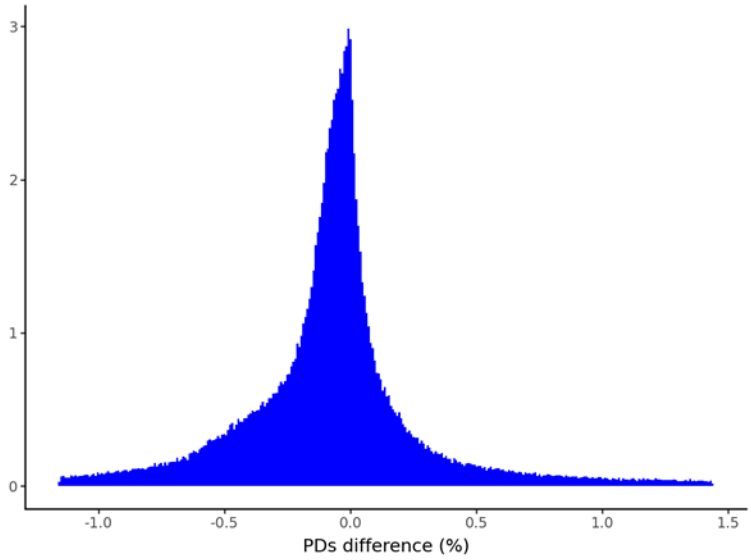
Thus, we conclude that: i) the meta-model is less affected by the disruptions that occurred in 2020 and 2021 than S-ICAS; ii) the meta-model performance appears to have recovered, after the pandemic, more than S-ICAS. We believe that the greater robustness of the meta-model depends on: i) the ability of ML models to capture non-linear and non-monotonic effects, as noted in Moscatelli *et al*. (2020); ii) the use of stacking, which improves robustness as it aggregates different base models.

[24] To predict 2023 PDs, 2021 financial statements data are used.
[25] We conducted a bootstrap analysis to verify the statistical significance of these results with 95 per cent confidence intervals for the difference between the AuROC of the stacked model and of S-ICAS (Appendix 4, Table A.9).

6. Forecasts explanations

The default probabilities generated by ML models can significantly differ from those produced by S-ICAS. For instance, Figure 4 shows the normalized histogram of the statistical distribution of the difference between the PDs produced by the meta-model and by S-ICAS for 2023.

Figure 4 – Statistical distribution of the difference between meta-model and S-ICAS PDs



PDs for year 2023.

These differences can be substantial, given that the average default probabilities produced by both models are around 1.5 percentage points. A comparison between the performance of both models in the tails of the distribution of the difference between meta-model and S-ICAS PDs by year[26] shows (Table 9) that the performance gap is greater than the one reported in the previous section.

Table 9 – Delta AuROC of complete models on the tails of the distribution of the difference between meta-model and S-ICAS PDs by year

Year	Delta AuROC
2014	0.021
2020	0.032
2021	0.045
2023	0.053

Also in this case the performance gap widens over the years: in 2014, the difference is 2.1 percentage points; in 2023 it reaches 5.3 percentage points. Moreover, in the tails the distribution of firms by size is slightly different (Table 10).

[26] We define the tails of the distribution as the observations lying above the 90th percentile and below the 10th percentile.

Table 10 – Distribution of firms by size

Size	Dataset (%)	Tails (%)
Micro	56.5	61.7
Small	33.2	30.2
Medium	8.4	6.7
Large	1.9	1.4

In the tails of the distribution of PDs difference, the percentage of micro-firms is greater while the percentage of medium-large firms is lower. This is an indication of the fact that the meta-model performs particularly well for micro-firms. This is consistent with the fact that the difference in the performance of credit behaviour ML models and credit behaviour logistic regression is greater for micro firms than for medium-large firms (see Appendix 3, Table A.6). This may be attributed to the circumstance that less information is available for smaller firms and that ML techniques help bridge the information gap, similarly to what highlighted by their contribution in periods, as the COVID-19 pandemic, where less information was available.

We have shown that the meta-model consistently outperforms and is more robust than S-ICAS. Consequently, if the probability of default produced by the meta-model significantly differs from that generated by S-ICAS for the same firm, it could indicate that S-ICAS faces a challenge in the assessment of the riskiness of the company. This discrepancy could serve as the basis for an alert system that flags such cases to the financial analysts in charge of the expert assessment stage. Hence, we are interested in a methodology capable of offering insight into the contribution of each input variable to the difference between the two PDs. Such information would allow financial analysts to concentrate on the most problematic areas, to understand if S-ICAS PD has indeed appropriately captured the company's credit risk. It is possible to associate each variable with its contribution to the prediction of any model using a specific XAI technique, the Shapley values, as discussed below.

6.1 Shapley values

Shapley values, originating from game theory (Shapley, 1953), provide a method to fairly distribute the total expected payout in a cooperative game among its participants, ensuring that certain desirable properties are satisfied (see Appendix 5).[27] In recent years, this technique has been applied to machine learning (Štrumbelj and Kononenko, 2014).

[27] In this context, a player's Shapley value corresponds to the payout he deserves, i.e. it quantifies the player's contribution to the team's win. Moreover, the sum of all the Shapley values is equal to the difference between the total expected payout and the fixed payout of the game (if the latter is different from 0).

The application is straightforward; let's assume we have a model that maps variables $\boldsymbol{x} = (x_1, x_2, \dots, x_K) \in \mathbb{R}^K$ to a prediction $f(\boldsymbol{x}) \in \mathbb{R}$. To determine the contribution of each variable to a specific prediction $f(\boldsymbol{x}^*)$ we can find its Shapley values ϕ_k (for $k = 1, 2, \dots K$), where: i) the values of the model's variables $\boldsymbol{x}^*$ take on the role of players; ii) the value of the prediction $f(\boldsymbol{x}^*)$ takes on the role of the total expected payout; iii) the average prediction $E[f(x)]$ takes on the role of the fixed payout of the game. In this case, the sum of all Shapley values is equal to the difference between $f(\boldsymbol{x}^*)$ and $E[f(x)]$.[28] Formally:

$$\sum_{k=1}^{K} \phi_k = f(\boldsymbol{x}^*) - E[f(x)]. \quad (1)$$

In order to calculate Shapley values, it is necessary to know (or to approximate) the joint distributions of the K variables $\boldsymbol{x}$. Besides, due to the computational burden of the exact Shapley value calculation, several approximation algorithms have been developed, such as the 'Shapley sampling value' (Štrumbelj and Kononenko, 2014)[29] and the 'Kernel SHAP' (Lundberg and Lee 2017b). While the latter is more computationally efficient, both methods assume that the variables are independent (see Appendix 5). This type of approximation is not accurate in many cases, such as in S-ICAS, where there is a non-negligible correlation among variables. To address this problem, Aas *et al.* (2021) develop some approximation algorithms that compute more accurate Shapley values, taking into consideration the variables' correlation. In this work, we use their methods to compute the Shapley values.[30]

6.2 Shapley values for ICAS expert assessment

We build an application to obtain, for a given firm, the Shapley values related to the difference between the PD forecast with the meta-model and the one produced with S-ICAS. This application is built to explain the difference in the financial PDs or the difference in the credit behaviour PDs, since financial analysts look at the two components separately during their analysis.

It is possible to show (see Appendix 5) that, by calculating the Shapley values related to the difference in the predictions between the two models, for a specific observation $\boldsymbol{x}^*$ we approximately get:

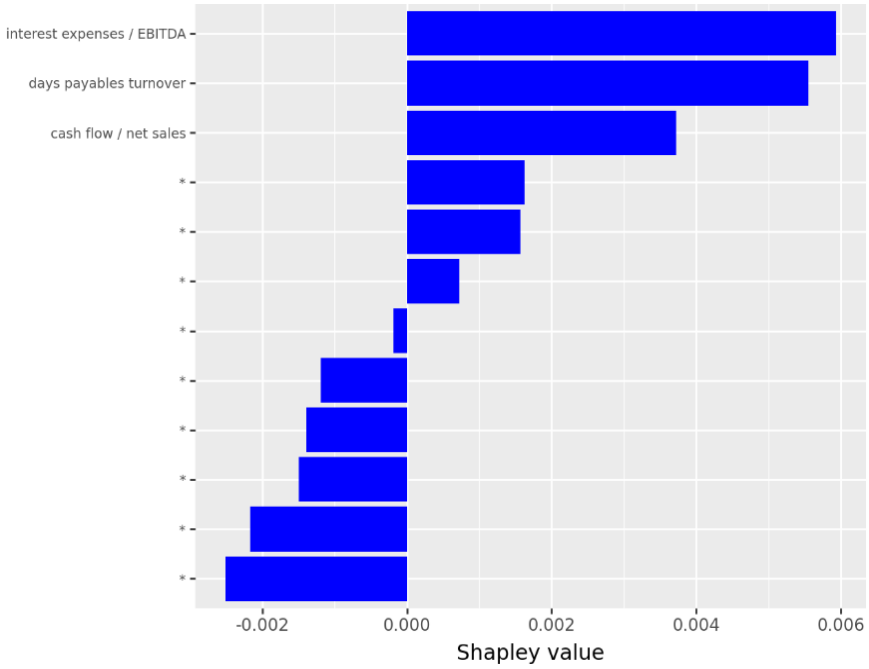
$$\sum_{k=1}^{K} \phi_k \approx f_m(\boldsymbol{x}^*) - f_s(\boldsymbol{x}^*), \quad (2)$$

[28] If we do not know the value of any variable, the best possible prediction is the expected value of the model output.
[29] This method has been applied for explaining ML predictions in Cascarino *et al.* (2022).
[30] The algorithms developed in Aas *et al.* (2021) are available in an R package, *shapr*, and in a Python package, *shaprpy*, which is a wrapper of the R package.

where ϕ_k, for $k = 1, 2, \ldots K$, are the Shapley values, $f_m(\boldsymbol{x}^*)$ is the PD produced by the meta-model and $f_s(\boldsymbol{x}^*)$ is the one produced by S-ICAS.[31] This approximation holds for the most problematic cases, the ones where the two PDs are very different from each other. For example, it holds when the two PDs belong to two not adjacent Credit Quality Steps (CQS).[32] In those cases, the sum of Shapley values is approximately equal to the difference between the two PDs.

We now present an example from the application of Shapley values.[33] Consider an industrial firm where the S-ICAS financial component predicts a default probability of 0.51 per cent, while the meta-model predicts a PD of 1.57 per cent. The meta-model deems the firm as much riskier. The Shapley values explaining the difference of the two PDs provide insights into the variables which mostly influence the higher risk signalled by the meta-model.

Figure 5 - Shapley values for the industrial company



Note: for simplicity, only variables with the greatest Shapley values are reported.

[31] The expected value in Equation 1 has vanished. Since each S-ICAS sub-model uses specific indicators, Equation 2 needs to be applied separately for each sub-model.
[32] The Credit Quality Steps are the core of the Eurosystem's harmonized rating scale. For further details, see European Central Bank (2015).
[33] We present a second example in Appendix 6.

In this example, the variables '*interest expenses divided by EBITDA*', '*days payables turnover*'[34] and '*cash flow divided by net sales*' have the largest Shapley value. Consequently, these indicators are the ones that contribute more to the positive difference in PDs.[35]

In the example we show how Shapley values can clarify aspects that may have represented a challenge for S-ICAS. We believe that the issue often lies in the presence of interaction effects among variables. In fact, S-ICAS is a multivariate logistic model that is additive in its nature. In some cases, a combination of some variables could have a strong positive or negative effect on a firm's conditions, and this effect might not be reflected accurately in S-ICAS PD. While it is true that Shapley values are not always straightforward to interpret,[36] in cases where there is a dominance of highly positive (or negative) Shapley values, the use of such values may be an effective support for ICAS financial analysts.

7. Conclusions

In this study, we use three machine-learning models - random forests, extreme gradient boosting and deep learning –, as well as a meta-model that integrates predictions from the three models, to replicate the Banca d'Italia's ICAS statistical model, which is estimated with logistic regressions. The dataset includes financial and credit behaviour variables for hundreds of thousands of Italian non-financial companies, from 2014 to 2023.

Within the financial component of S-ICAS, we observe that RF and XGBoost yield an improvement of the discriminatory power, as measured by the AuROC, of 2.3 to 3.3 percentage points over logistic regressions, while deep learning yields slightly less pronounced increments. ML models provide an increase in discriminatory power, though smaller, also in the credit behaviour component (between 0.5 and 1.5 percentage points). We also find that the meta-model that integrates the predictions from the three ML models outperforms the individual models. When looking at the complete model, the meta-model consistently outperforms the discriminatory power of S-ICAS by 0.6 to 1.9 percentage points depending

[34] The average number of days it takes a company to pay back its accounts payable.
[35] To better understand this point, we consulted a financial analyst to assess the firm's financial condition. The analyst, paying attention to the three variables reported above, said that the firm had low margins and a good part of the company liquidity was absorbed by suppliers. This analysis reveals that the company has limited available liquidity and a low capacity for cash generation, which could become problematic, especially during periods of financial distress. For these reasons, a PD of 0.51 per cent seems to be too low when compared to the true riskiness of the firm, while a PD of 1.57 per cent is more appropriate. There are also variables with a negative Shapley value. These variables contribute negatively to the difference between the two PDs, i.e. they indicate that, for the meta-model, there are some aspects of the firm's financial situation that S-ICAS has judged too negatively. However, in this case these variables have a small weight, so they provide a small contribution.
[36] It is possible to have at the same time both high positive and negative Shapley values. These situations can be hard to interpret.

on the year. Furthermore, the meta-model performance is less affected by the disruptions that occurred in 2020 and 2021 and shows a stronger performance recovery than S-ICAS in the post-pandemic period.

Meta-model PDs could be used by financial analysts in the expert assessment stage to improve ICAS ratings, by focusing on aspects where ML models and S-ICAS provide different signals. To make this information explainable, we use Shapley values that can provide financial analysts with the contribution of each variable of the model to the PD difference. Even though we have not carried out a fully-fledged evaluation[37] of the effectiveness of Shapley values in capturing S-ICAS biases, from the examples we have analysed we verified that these values can highlight aspects that may have not been fully grasped by S-ICAS. This would enable analysts to focus on the most problematic areas of the analysis. Nevertheless, caution is needed, since the interpretation of Shapley values may be challenging, especially when both highly positive and highly negative values are present. These conditions also show that the credit assessment process cannot rely only on ML models, reflecting also regulatory concerns, due to their complexity which makes them difficult to be interpreted and understood by all the relevant stakeholders. Overall, our results indicate that ML models have the potential to enhance both the efficiency and quality of the ICAS process, refining and reinforcing it through a comprehensive and effective analysis of firm-related information. We plan to further investigate the benefits of integrating ML models in ICAS, including by evaluating ex-post their contribution to S-ICAS and analysts' evaluations.

[37] A fully-fledged evaluation, which we plan for future work, would require providing Shapley values to analysts for a significant number of prospective assessments and then verifying ex-post if the Shapley values improved the performance of ICAS full evaluations and that of S-ICAS.

References

Aas, K., Jullum, M., Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence, 298*, 103502.

Alexandropoulos, S.-A. N., Aridas, C. K., Kotsiantis, S. B., Vrahatis, M. N. (2019). Stacking strong ensembles of classifiers. *Artificial Intelligence Applications and Innovations*, Springer, Cham, pp. 545–556.

Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Asari, V. K. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics, 8* (3), 292.

Altmann, A., Toloşi, L., Sander, O., Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. *Bioinformatics, 26*(10), 1340-1347.

Angelini, P. (2025). Data Science in Central Banking. Remarks at the 4th workshop on "Data science in Central Banking", Banca d'Italia, Rome, 20 February 2025.

Apley, D. W., Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B: Statistical Methodology, 82*(4), 1059-1086.

Barboza, F., Kimura, H., Altman, E. (2017). Machine learning models and bankruptcy prediction. *Expert Systems with Applications, 83*, 405-417.

Bitetto, A., Cerchiello, P., Filomeni, S., Tanda, A., Tarantino, B. (2023). Machine learning and credit risk: Empirical evidence from small-and mid-sized businesses. *Socio-Economic Planning Sciences, 90*, 101746.

Breiman, L. (1996). Stacked regressions. *Machine Learning, 24*, 49–64.

Breiman, L. (2001a). Random forests. *Machine learning, 45*, 5-32.

Breiman, L. (2001b). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science, 16*(3), 199-231.

Bussmann, N., Giudici, P., Marinelli, D., Papenbrock, J. (2021). Explainable machine learning in credit risk management. *Computational Economics, 57*(1), 203-216.

Cascarino, G., Moscatelli, M., Parlapiano, F. (2022). Explainable artificial intelligence: interpreting default forecasting models based on machine learning. *Bank of Italy Occasional Paper*, (674).

Chen, T., Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Ciampi, F., Gordini, N. (2013). Small enterprise default prediction modeling through artificial neural networks: an empirical analysis of italian small enterprises. *Journal of Small Business Management*, *51*(1), 23-45.

Dal Pozzolo, A., Caelen, O., Bontempi, G. (2015). When is undersampling effective in unbalanced classification tasks?. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part I 15* (pp. 200-215). Springer International Publishing.

Dell, M. (2024). Deep Learning for Economists. *NBER Working Papers* No. 32768.

Dietterich, T. G. (2000). Ensemble methods in machine learning. *Multiple classifier systems*, Springer, Berlin, Heidelberg, pp. 1–15.

Dubey, S. R., Singh, S. K., Chaudhuri, B. B. (2022). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503, 92-108.

European Banking Authority (2020). EBA Report on Big Data and Advanced Analytics.

European Banking Authority (2023). Machine learning for IRB models. *Follow-up Report from the Consultation on the Discussion Paper on Machine Learning for IRB Models*.

European Central Bank (2015). The financial risk management of the Eurosystem′s monetary policy operations.

Giovannelli F., Iannamorelli A., Levy A., Orlandi M., 2023, The Bank of Italy's In-House Credit Assessment System for Non-financial Firms, in: Scalia, A. (eds) *Financial Risk Management and Climate Change Risk. Contributions to Finance and Accounting*, Springer, 107-137.

Gregova, E., Valaskova, K., Adamko, P., Tumpach, M., Jaros, J. (2020). Predicting financial distress of slovak enterprises: Comparison of selected traditional and learning algorithms methods. *Sustainability*, *12*(10), 3954.

He, H., Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, *21*(9), 1263-1284.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2* (IJCAI'95), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 1137-1143.

LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Li, Y., Wang, Y. (2017). Machine learning methods of bankruptcy prediction using accounting ratios. *Open Journal of Business and Management*, *6*(1), 1-20.

Louppe, G., Wehenkel, L., Sutera, A., Geurts, P. (2013). Understanding Variable Importances in Forests of Randomized Trees. *Advances in Neural Information Processing Systems, 26*, 431-439.

Loyola-Gonzalez, O. (2019). Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view. *IEEE access, 7*, 154096-154113.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Lee, S. I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature machine intelligence*, *2*(1), 56-67.

Lundberg, S. M., Lee, S. I. (2017a). Consistent feature attribution for tree ensembles. *arXiv preprint arXiv:1706.06060*.

Lundberg, S. M., Lee, S. I. (2017b). A unified approach to interpreting model predictions. *Advances in neural information processing systems, 30*, 4765 – 4774.

Moscatelli, M., Parlapiano, F., Narizzano, S., Viggiano, G. (2020). Corporate default forecasting with machine learning. *Expert Systems with Applications, 161*, 113567.

Narayanan A., Kapoor S. (2024), *AI Snake Oil: What Artificial Intelligence Can Do, What It Can't, and How to Tell the Difference*, Princeton University Press, Princeton.

Narizzano, S., Orlandi, M., Scalia, A. (2024). The Bank of Italy's statistical model for the credit assessment of non-financial firms. *Markets, Infrastructures, Payment Systems* (53), 1-61.

Petropoulos, A., Siakoulis, V., Stavroulakis, E., Klamargias, A. (2019). A robust machine learning approach for credit risk analysis of large loan level datasets using deep learning and extreme gradient boosting. *IFC Bulletins chapters, 49*.

Ribeiro, M. T., Singh, S., Guestrin, C. (2016). " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135-1144).

Schalck, C., Yankol-Schalck, M. (2021). Predicting French SME failures: new evidence from machine learning techniques. *Applied Economics, 53*(51), 5948-5963.

Shapley, L. (1953) A Value for n-Person Games. In: Kuhn, H. and Tucker, A., Eds., Contributions to the Theory of Games II, Princeton University Press, Princeton, 307-317.

Shi, S., Tse, R., Luo, W., D'Addona, S., Pau, G. (2022). Machine learning-driven credit risk: a systemic review. *Neural Computing and Applications, 34*(17), 14327-14339.

Shwartz-Ziv, R., Armon, A. (2022). Tabular Data: Deep Learning is Not All You Need. *Information Fusion, 81*, 84-92.

Štrumbelj, E., Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems, 41*, 647-665.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5* (2), 241–259.

Yu, T., Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689.*

Zhang, Y., Song, K., Sun, Y., Tan, S., Udell, M. (2019). " Why should you trust my explanation?" understanding uncertainty in LIME explanations. *arXiv preprint arXiv:1904.12991*.

Zhang, W., Yan, S., Li, J., Tian, X., Yoshida, T. (2022). Credit risk prediction of SMEs in supply chain finance by fusing demographic and behavioral data. *Transportation Research Part E: Logistics and Transportation Review, 158,* 102611.

Appendix 1 – ICAS

Components

The statistical model has two components. A logistic regression is estimated for each component, providing a credit score. The components are as follows:

- the credit behaviour component uses individual firm data from the NCR. Different sub-models exist for each firm class according to size;[38] companies are grouped into micro, small, and medium/large-sized classes. Each of the three sub-models uses specific regressors;
- the financial component employs financial data on individual firms provided by Cerved Group.[39] This component involves 11 sub-models based on the macro-sector and the type of financial statement (ordinary or simplified). The sectors include industry, trade, construction, services, real estate, and holdings. Two models have been estimated for each of the first five sectors: one for companies with an ordinary financial statement and one for companies with a simplified financial statement. For the holding sector, a single model has been estimated. Each of the 11 sub-models uses specific regressors.

The results of these two components are then merged through an integration module that provides the S-ICAS final score. The integration module is specialised into four sub-models according to firm size (micro, small, medium, and large). The score produced by S-ICAS is mapped into the statistical PD via the inverse logit function.[40] A higher score implies a higher probability of default. For more details, see Narizzano *et al*. (2024).

Default definition

The dependent variable used in the estimation of each S-ICAS sub-models is binary and represents the status of the company at the end of the year following the initial moment of observation. It is valued at 1 if the firm has defaulted in at least one of the twelve months and it is valued at 0 otherwise. More in detail, a company is considered in default in a given month if both the following conditions occur (Giovannelli *et al.*, 2023):

[38] As defined by the European Commission 2003/361/EC, according to article 2 of the Annex, the category of micro, small and medium-sized enterprises (SMEs) is made up of enterprises that employ fewer than 250 persons and achieve an annual turnover not exceeding EUR 50 million, and/or total assets not exceeding EUR 43 million. Within the SME category, a small enterprise is defined as one that employs fewer than 50 persons and whose annual turnover and/or total assets do not exceed EUR 10 million. Within the SME category, a micro enterprise is defined as an enterprise that employs fewer than 10 persons and whose annual turnover and/or total assets do not exceed EUR 2 million.

[39] Cerved Group maintains an extensive dataset covering the Italian corporate sector, which includes nearly all small and micro limited-liability firms. This dataset is sourced from the National Official Business Register.

[40] The score is defined as the natural logarithm of odds. Mathematically, $score = \ln(\frac{PD}{1-PD})$.

1. The exposure reported as bad debt, unlikely to pay or past-due (over 90 days) exceeds 5 per cent of the total exposure of the company to the financial system and is greater than 500 euros;
2. The previous condition has occurred for at least three consecutive months.

However, if a bank reports a loss, the previous conditions are not applied and the company is automatically considered in default. Data on exposures and defaults are taken from the NCR.

Appendix 2 – Methods and variable selection

Cross-validation and optimization

As for cross-validation, in stratified K-fold cross-validation, the data is divided into K folds, ensuring a proportional representation of each class in every fold. The model is iteratively trained on K-1 folds and validated on the remaining fold, with this process repeated K times so that each fold is used as a validation set exactly once. This approach provides a comprehensive evaluation of the model performance across different data splits, reducing the variance associated with a single train-test split. We set the number of folds to 10, as specified by the *cv* parameter, ensuring a thorough and reliable evaluation.

Cross-validation is used to maximize the validation AuROC, guiding the search process toward configurations that enhance the model ability to discriminate between default and non-default cases. The search inside the hyper-parameter space is conducted using Bayesian optimization, which is a probabilistic, model-based approach that constructs a surrogate model to approximate the objective function. It uses this surrogate to select the most promising hyper-parameters to evaluate, balancing exploration and exploitation. This method is particularly effective for complex search spaces, as it converges to optimal solutions more efficiently than grid or random search methods.[41]

In the context of hyper-parameter optimization, there are several specialized software libraries designed to efficiently search for the best combination of parameters. Among these, *scikit-optimize* stands out as a particularly effective tool. This library provides advanced algorithms for Bayesian optimization. The decision to use scikit-optimize is based on its flexibility in handling various types of hyper-parameters (including both continuous and categorical variables). Compared to other libraries, *scikit-optimize* offers a balance between ease of use and powerful optimization techniques, making it one of the best choices for hyper-parameter tuning in machine learning models.

Regularization and training in deep learning

Regularization in deep learning involves adding penalties to the loss function for larger weights, preventing the model from becoming overly complex and overfitting the training data. By discouraging the network from assigning excessive importance to any single feature, regularization ensures that the model generalizes better to new, unseen data, ultimately leading to more robust predictions. Dropout, a different form of regularization, operates by randomly deactivating a fraction of neurons during each training iteration. This operation avoids that the network becomes too dependent on any single neuron or small group of neurons, which might otherwise capture patterns specific only to the training data. As a result, the network is forced to distribute the learning process across multiple neurons, leading to the development of a more robust and diverse feature representation.

[41] For a survey on optimization algorithms see, for example, Yu and Zhu (2020).

Stochastic Gradient Descent (SGD) is a fundamental optimization method employed in training deep learning, especially in the backpropagation process. In the context of backpropagation, SGD iteratively adjusts the model weights by computing the gradient of the loss function with respect to each weight. Instead of calculating the gradient over the entire dataset, SGD uses a randomly selected subset of the data (a mini-batch) to update the weights, making the process faster and more scalable for large datasets. This iterative adjustment helps the model minimize the loss function and gradually converge toward the optimal set of weights, improving accuracy. For all the sub-models estimated, the optimizer chosen is *Adam*[42], renowned for its adaptive learning rate capabilities, which combines the benefits of two other extensions of stochastic gradient descent, namely *AdaGrad*[43] and *RMSProp*[44]. The loss function selected is the *binary cross entropy*, which is well-suited for binary classification tasks as it measures the performance of the model by comparing the predicted probabilities to the actual binary labels, thereby guiding the model to improve its predictions through gradient descent.

[42] *Adam* is an adaptive optimizer widely used in deep learning, known for efficiently handling sparse gradients and noisy data. It adjusts the learning rate dynamically for each parameter during training, which helps to improve convergence and model performance in a variety of tasks.
[43] *AdaGrad* (Adaptive Gradient Algorithm) is an optimization method that adapts the learning rate for each parameter, improving convergence on sparse data.
[44] *RMSProp* (Root Mean Square Propagation) is an optimization algorithm that adjusts the learning rate for each parameter based on the moving average of recent gradient magnitudes.

Variable selection

Table A.1 and Table A.2 present the input variables for the ordinary and simplified financial sub-models, while Table A.3 lists the variables used in the credit behaviour sub-models. Information on the sign of each regression coefficient is also reported (+/-). Regarding the financial component of S-ICAS, raw data are initially provided by Cerved Group, from which we compute the indicators used across the eleven financial sub-models, with each sub-model utilizing its own set of specific indicators. For the credit behaviour component, the variables used in the three sub-models are derived from NCR data.

Table A.1 – Input variables for the ordinary financial sub-models

Risk area	Variable	Industrial sector				
		Industrial	Trade	Construction	Services	Real Estate
Profitability	Cash flow / Net sales	-	-		-	
	Cash flow / Total assets			-		
	Value added / Total assets					-
Leverage & financial structure	Equity / Net financial debt	-	-	-	-	-
	Fixed assets / Total assets					-
Debt sustainability	Interest expenses / EBITDA	+	+	+	+	
	Interest expenses / Cash flow					+
	Interest expenses / Net sales	+	+		+	
	Net sales / Total net debt	-	-		-	
	ROD (return on debt) = Interest expenses / Average financial debt	+	+	+	+	+
Liquidity	Current ratio = current assets / current liabilities	-		-		
	Financial mismatch = (current liabilities – current assets) / total assets		+		+	
	Cash / Total short term debt	-	-	-	-	-
Activity	Days receivables turnover + Days inventory turnover (discretize)	+				
	Days receivables turnover (discretize)			+	+	+
	Days payables turnover (discretize)	+	+	+		
Business development	Net sales negative variation (discretize)	+	+		+	
Size & Age	Log(age)	-	-	-	-	

Table A.2 – Input variables for the simplified financial sub-models

Risk area	Variable	Industrial sector					
		Industrial	Trade	Construction	Services	Real Estate	Holding
Profitability	Cash flow / Net sales	-	-		-		
	Cash flow / Total assets			-			
	Value added / Total assets					-	
	Ebit / Total assets						-
	Delta returns						-
Leverage & financial structure	Equity / Total net debt	-	-	-	-	-	
	Equity / Total assets						-
	Fixed assets / Total assets					-	
	Financial assets / Total assets						-
Debt sustainability	Interest expenses / EBITDA	+	+	+	+		
	Interest expenses / Cash flow					+	
	Interest expenses / Net sales		+		+	+	
	Net sales / Total net debt	-	-		-		
Liquidity	Financial mismatch = (current liabilities – current assets) / total assets	+	+	+	+		
	Cash / Total short term debt	-	-	-	-	-	
	Cash / Total debt						-
Activity	Days receivables turnover + Days inventory turnover (discretize)	+	+		+		
Business development	Net sales negative variation (discretize)	+	+		+		
Age	Log(age)	-	-	-	-		

Table A.3 – Input variables for the three credit behaviour models

Risk area	Variable	Credit Size		
		Micro	Small	Medium-Large
Average utilization rate	Drawn amount/granted amount - current account revolving credit lines - average last three-months	+	+	+
	Dummy no-current account revolving credit lines - last three-months	+	+	+
	Drawn amount/granted amount – short term credit lines - average last six-months			+
	Dummy no-short term credit lines - last six-months			+
	Drawn amount/granted amount – account receivables revolving credit lines - average last six-months	+	+	
	Dummy no-account receivables revolving credit lines - last six-months	+	+	
Debt composition	Dummy no medium/long term credit – last six-months	+	+	

Financial distress	N. of months (last six-months) with overdraft on current account revolving credit lines	+	+	+
	N. of months (last six-months) with overdraft on term loans	+	+	+
	Overdraft % on term loans, 6 month average (discretize)	+	+	+
	Dummy default status – last six-months	+	+	+
Quality of credit receivables	Unpaid/Expired amount on account receivables credit lines - average last six-months (discretize)	+	+	
	Unpaid amount on account receivables credit lines - average last six-months / net sales (discretize)			+
Trend	Dummy reduction in the number of reporting banks – last six-months	+	+	+
	Number of first information requests - last six-months	+	+	
Size	Net sales bucket (discretize)			-

Appendix 3 – S-ICAS and ML sub-models performance

In Table A.4 and A.5 we report the discriminatory power, calculated in terms of AuROC on the entire test set, of S-ICAS and ML models, for each financial sub-component.

Table A.4 – AuROC for the financial component - ordinary financial statement

Sector	S-ICAS	Random forests	XGBoost	Deep learning
Industry	0.832	0.854	0.852	0.849
Trade	0.781	0.809	0.806	0.804
Construction	0.783	0.807	0.812	0.803
Services	0.782	0.801	0.801	0.796
Real estate	0.820	0.837	0.830	0.834

Table A.5 – AuROC for the financial component – simplified financial statement

Sector	S-ICAS	Random forests	XGBoost	Deep learning
Industry	0.774	0.789	0.795	0.792
Trade	0.738	0.771	0.770	0.769
Construction	0.713	0.739	0.741	0.738
Services	0.741	0.759	0.761	0.757
Real estate	0.759	0.774	0.776	0.774
Holding	0.750	0.773	0.785	0.757

In Table A.6 we report the discriminatory power, calculated on the entire test set, of S-ICAS and ML models, for each credit behaviour sub-component.

Table A.6 – AuROC for the credit behaviour component

Size	S-ICAS	Random forests	XGBoost	Deep learning
Micro	0.856	0.862	0.864	0.861
Small	0.884	0.891	0.892	0.888
Medium-large	0.871	0.873	0.873	0.871

Appendix 4 – AuROC confidence intervals

The superior discriminatory power of ML models showed in Section 4 and 5 is statistically significant. Using bootstrap, we construct 95 per cent confidence intervals for the difference in the AuROC of ML models and the AuROC of S-ICAS. As we can see from Table A.7, A.8, and A.9, the lower limit of all the confidence intervals is always greater than zero.

Table A.7 – 95 per cent confidence intervals for AuROC difference - ML vs S-ICAS financial component

Year	Random forests	XGBoost	Deep learning	Stacked
2014	(0.020, 0.025)	(0.022, 0.027)	(0.018, 0.022)	(0.025, 0.029)
2023	(0.027, 0.035)	(0.030, 0.038)	(0.023, 0.031)	(0.033, 0.040)

Table A.8 – 95 per cent confidence intervals for AuROC difference - ML vs S-ICAS credit component

Year	Random forests	XGBoost	Deep learning	Stacked
2014	(0.002, 0.004)	(0.004, 0.006)	(0.004, 0.005)	(0.005, 0.007)
2023	(0.010, 0.015)	(0.013, 0.018)	(0.005, 0.008)	(0.014, 0.018)

Table A.9 – 95 per cent confidence intervals for AuROC difference - ML vs S-ICAS complete models

Year	Stacked
2014	(0.005, 0.007)
2020	(0.012, 0.015)
2021	(0.011,0.015)
2023	(0.017, 0.021)

Appendix 5 – Mathematical theory behind Shapley values

In a cooperative game with K players the goal is to maximise a payout v, with the function $v: S \rightarrow v(S)$ associating each subset $S \subseteq M = \{1, 2, \dots, K\}$ of players with the expected payout they can win together. Shapley values distribute the total expected payout $v(M)$ among the K players, ensuring that some desirable properties are met.

For a player k, with $k = 1, 2, \dots K$, the payout portion (his contribution to the team's win) is called Shapley value, denoted as ϕ_k. These values uniquely satisfy the following desirable properties (Shapley, 1953):

1) (*Additivity*) The total payout is distributed additively among players:

$$\sum_{k=0}^{K} \phi_k = v(M), \qquad (A.1)$$

where ϕ_0, if it differs from 0, represents a fixed payout not dependent on the players' contributions.

2) If two players i, j contribute equally to the payout regardless of the other players participating, i.e.:

$$v(S \cup \{i\}) = v(S \cup \{j\}), \qquad (A.2)$$

for every $S \subseteq M \backslash \{i, j\}$, then their Shapley values ϕ_i and ϕ_j are equal.

3) If a player's inclusion does not increase the expected payout for any group, i.e.:

$$v(S \cup \{i\}) = v(S), \qquad (A.3)$$

for every $S \subseteq M \backslash \{i\}$, then its Shapley values ϕ_i is equal a 0.

4) (*Linearity*) For two games with payout functions v and w, a player's Shapley value for the combined game equals the sum of his Shapley values for each game:

$$\phi_k(v + w) = \phi_k(v) + \phi_k(w), \qquad (A.4)$$

for each $k = 1, 2, \dots, K$. Additionally, for any real constant α: $\phi_k(\alpha v) = \alpha \phi_k(v)$.

Now, let's assume that we have a model that maps variables $\boldsymbol{x} = (x_1, x_2, \dots, x_K) \in \mathbb{R}^K$ to a prediction $f(\boldsymbol{x}) \in \mathbb{R}$. If we want to explain a specific prediction $f(\boldsymbol{x}^*)$ we can find its Shapley values, where:

1) The values of the K variables $\boldsymbol{x}^* = ({x_1}^*, {x_2}^*, \dots, {x_K}^*)$ take on the role of players.

2) The prediction $f(\boldsymbol{x}^*)$ takes on the role of the payout $v(M)$ and the expected value of the prediction of the model $E[f(\boldsymbol{x})]$ represents the fixed payout that does not depend on the contribution of the variables/players (ϕ_0). The property of additivity (Equation A.1) can then be rewritten:

$$\sum_{k=1}^{K} \phi_k = f(\boldsymbol{x}^*) - E[f(\boldsymbol{x})]. \quad (A.5)$$

To compute Shapley values, the reward function $v: S \rightarrow v(S)$ must be known for every subset S of the model's variables. Consistently with the definition provided earlier, the reward function for each subset S can be defined (Štrumbelj and Kononenko, 2014) as the expected value of the model's prediction knowing only the values of the variables $\boldsymbol{x}^*$ present in S (denoted as $\boldsymbol{x}^*{}_S$):

$$v(S) = E[f(\boldsymbol{x})|\boldsymbol{x}_S = \boldsymbol{x}^*{}_S]. \quad (A.6)$$

In other words, $v(S)$ represents the contribution that the knowledge of the value of the variables in S gives to the prediction.

Therefore, in order to calculate Shapley values, it is necessary to know (or to approximate) the joint distributions of the variables $\boldsymbol{x}$. Besides, due to the computational burden of the exact Shapley value calculation, several approximation algorithms have been developed. The algorithms that are most used in the literature have been defined in Štrumbelj and Kononenko (2014) and in Lundberg and Lee (2017b). However, those methods assume that the variables are independent, i.e. the conditional distribution of each subset of variables $\bar{S} = M \backslash S$ does not depend on the variables in S:

$$p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}^*{}_S) = p(\boldsymbol{x}_{\bar{S}}). \quad (A.7)$$

Instead, algorithms developed more recently in Aas *et al.* (2021) approximate $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}^*{}_S)$ without using the independence assumption.

When applying Shapley values to the difference in the predictions between two models, using the additivity property (Equations A.1 and A.5), for a specific observation $\boldsymbol{x}^*$ we get:

$$\sum_{k=1}^{K} \phi_k = f_m(\boldsymbol{x}^*) - f_s(\boldsymbol{x}^*) - (E[f_m(\boldsymbol{x})] - E[f_s(\boldsymbol{x})]), \quad (A.8)$$

where $f_m(\boldsymbol{x}^*)$ is the PD produced by the first model and $f_s(\boldsymbol{x}^*)$ is the PD produced by the second model.

The two expected values in Equation A.8, $E[f_m(\boldsymbol{x})]$ and $E[f_s(\boldsymbol{x})]$, can both be approximated with the average prediction of the corresponding model on the training set. If the two models had been trained on the same dataset, this difference is generally small but in some cases it can be not negligible. However, if we concentrate on explaining the most problematic cases, where the PDs produced by the two models are very different from each other, we can simplify Equation A.8 and write:

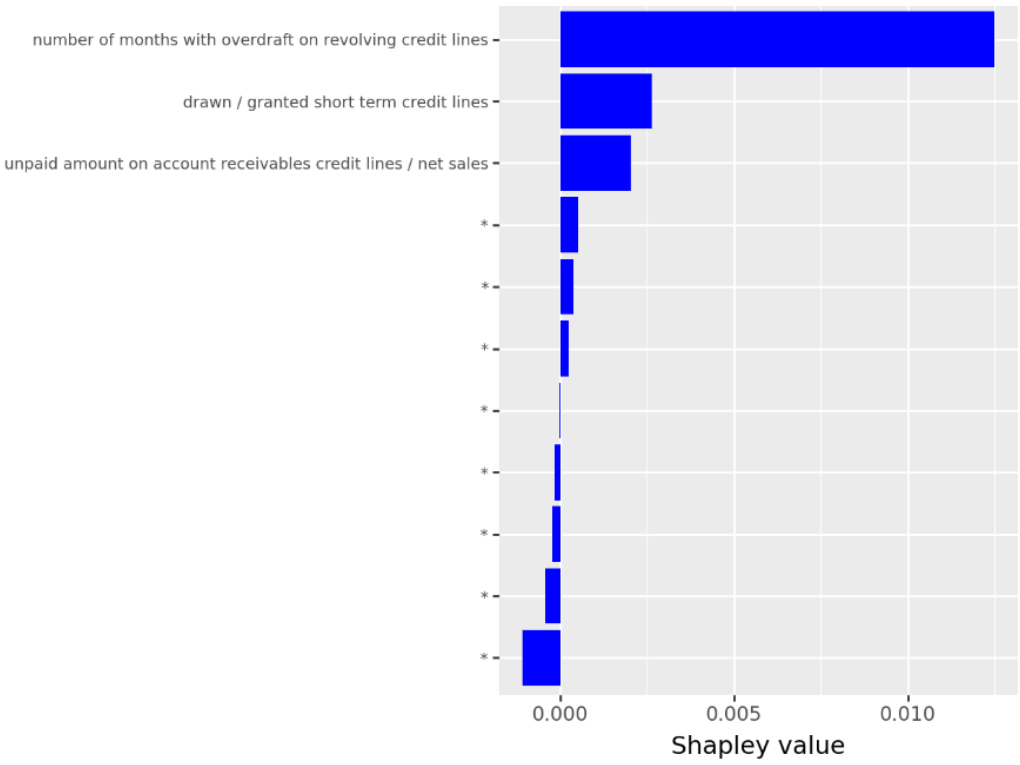
$$\sum_{k=1}^{K} \phi_k \approx f_m(\boldsymbol{x}^*) - f_s(\boldsymbol{x}^*), \qquad (A.9)$$

since in those cases $f_m(\boldsymbol{x}^*) - f_s(\boldsymbol{x}^*) \gg E[f_m(\boldsymbol{x})] - E[f_s(\boldsymbol{x})]$. This means that the sum of Shapley values is approximately equal to the difference between the two PDs.

Appendix 6 – Example on the explainability of the credit behaviour component

We consider a medium-large firm where S-ICAS credit behaviour component predicts a default probability of 0.99 per cent, while the meta-model predicts a credit behaviour PD of 2.42 per cent. The meta-model judges the company much riskier. Below we plot the Shapley values relative to the difference of the two PDs.

Figure A.1 - Shapley values for the medium-large company[45]



As we can see, the variable "*number of months with overdraft on current account revolving credit lines*"[46] has by far the greatest Shapley value and contributes the most to the difference in PDs. The financial analyst, examining the company's situation, observed that not only the number of months with overdraft is high, but the company has also a discrete amount of "*unpaid amount on account receivables credit lines compared to net sales*" and of the variable "*drawn divided by granted on short term credit lines*" (in the last six months). According to the analyst's opinion, the combination of these aspects is enough to raise suspicions on the credit soundness of the company and on the value of S-ICAS PD, which has a very low value.

[45] For simplicity, only variables with the greatest Shapley values are reported.
[46] The value is calculated looking at the past six months.

Recently published papers in the 'Markets, Infrastructures, Payment Systems' series

n. 37 Smart Derivative Contracts in DatalogMTL, *by Andrea Colombo, Luigi Bellomarini, Stefano Ceri and Eleonora Laurenza*

n. 38 Making it through the (crypto) winter: facts, figures and policy issues, *by Guerino Ardizzi, Marco Bevilacqua, Emanuela Cerrato and Alberto Di Iorio*

n. 39 The Emissions Trading System of the European Union (EU ETS), *by Mauro Bufano, Fabio Capasso, Johnny Di Giampaolo and Nicola Pellegrini* (in Italian)

n. 40 Banknote migration and the estimation of circulation in euro area countries: the italian case, *by Claudio Doria, Gianluca Maddaloni, Giuseppina Marocchi, Ferdinando Sasso, Luca Serrai and Simonetta Zappa* (in Italian)

n. 41 Assessing credit risk sensitivity to climate and energy shocks, *by Stefano Di Virgilio, Ivan Faiella, Alessandro Mistretta and Simone Narizzano*

n. 42 Report on the payment attitudes of consumers in Italy: results from the ECB SPACE 2022 survey, *by Gabriele Coletti, Alberto Di Iorio, Emanuele Pimpini and Giorgia Rocco*

n. 43 A service architecture for an enhanced Cyber Threat Intelligence capability and its value for the cyber resilience of Financial Market Infrastructures, *by Giuseppe Amato, Simone Ciccarone, Pasquale Digregorio and Giuseppe Natalucci*

n. 44 Fine-tuning large language models for financial markets via ontological reasoning, *by Teodoro Baldazzi, Luigi Bellomarini, Stefano Ceri, Andrea Colombo, Andrea Gentili and Emanuel Sallinger*

n. 45 Sustainability at shareholder meetings in France, Germany and Italy, *by Tiziana De Stefano, Giuseppe Buscemi and Marco Fanari* (in Italian)

n. 46 Money market rate stabilization systems over the last 20 years: the role of the minimum reserve requirement, *by Patrizia Ceccacci, Barbara Mazzetta, Stefano Nobili, Filippo Perazzoli and Mattia Persico*

n. 47 Technology providers in the payment sector: market and regulatory developments, *by Emanuela Cerrato, Enrica Detto, Daniele Natalizi, Federico Semorile and Fabio Zuffranieri*

n. 48 The fundamental role of the repo market and central clearing, *by Cristina Di Luigi, Antonio Perrella and Alessio Ruggieri*

n. 49 From Public to Internal Capital Markets: The Effects of Affiliated IPOs on Group Firms, *by Luana Zaccaria, Simone Narizzano, Francesco Savino and Antonio Scalia*

n. 50 Byzantine Fault Tolerant consensus with confidential quorum certificate for a Central Bank DLT, *by Marco Benedetti, Francesco De Sclavis, Marco Favorito, Giuseppe Galano, Sara Giammusso, Antonio Muci and Matteo Nardelli*

n. 51 Environmental data and scores: lost in translation, *by Enrico Bernardini, Marco Fanari, Enrico Foscolo and Francesco Ruggiero*

n. 52 How important are ESG factors for banks' cost of debt? An empirical investigation, *by Stefano Nobili, Mattia Persico and Rosario Romeo*

n. 53 The Bank of Italy's statistical model for the credit assessment of non-financial firms, *by Simone Narizzano, Marco Orlandi and Antonio Scalia*

n. 54 The revision of PSD2 and the interplay with MiCAR in the rules governing payment services: evolution or revolution?, *by Mattia Suardi*

n. 55 Rating the Raters. A Central Bank Perspective, *by Francesco Columba, Federica Orsini and Stefano Tranquillo*

n. 56 A general framework to assess the smooth implementation of monetary policy: an application to the introduction of the digital euro, *by Annalisa De Nicola and Michelina Lo Russo*

n. 57 The German and Italian Government Bond Markets: The Role of Banks versus Non-Banks. A joint study by Banca d'Italia and Bundesbank, *by Puriya Abbassi, Michele Leonardo Bianchi, Daniela Della Gatta, Raffaele Gallo, Hanna Gohlke, Daniel Krause, Arianna Miglietta, Luca Moller, Jens Orben, Onofrio Panzarino, Dario Ruzzi, Willy Scherrieble and Michael Schmidt*

n. 58 Chat Bankman-Fried? An Exploration of LLM Alignment in Finance, *by Claudia Biancotti, Carolina Camassa, Andrea Coletta, Oliver Giudice and Aldo Glielmo*

n. 59 Modelling transition risk-adjusted probability of default, *by Manuel Cugliari, Alessandra Iannamorelli and Federica Vassalli*

n. 60 The use of Banca d'Italia's credit assessment system for Italian non-financial firms within the Eurosystem's collateral framework, *by Stefano Di Virgilio, Alessandra Iannamorelli, Francesco Monterisi and Simone Narizzano*

n. 61 Fintech Classification Methodology, *by Alessandro Lentini, Daniela Elena Munteanu and Fabrizio Zennaro*

n. 62 The Rise of Climate Risks: Evidence from Expected Default Frequencies for Firms, *by Matilde Faralli and Francesco Ruggiero*

n. 63 Exploratory survey of the Italian market for cybersecurity testing services, *by Anna Barcheri, Luca Bastianelli, Tommaso Curcio, Luca De Angelis, Paolo De Joannon, Gianluca Ralli and Diego Ruggeri*

n. 64 A practical implementation of a quantum-safe PKI in a payment systems environment, *by Luca Buccella and Stefano Massi*

n. 65 Stewardship Policies. A Survey of the Main Issues, *by Marco Fanari, Enrico Bernardini, Elisabetta Cecchet, Francesco Columba, Johnny Di Giampaolo, Gabriele Fraboni, Donatella La Licata, Simone Letta, Gianluca Mango and Roberta Occhilupo*

n. 66 Is there an equity greenium in the euro area?, *by Marco Fanari, Marianna Caccavaio, Davide Di Zio, Simone Letta and Ciriaco Milano*

n. 67 Open Banking in Italy: A Comprehensive Report, *by Carlo Cafarotti and Ravenio Parrini*

n. 68 Report on the payment attitudes of consumers in Italy: results from ECB SPACE 2024 survey, *by Gabriele Coletti, Marialucia Longo, Laura Painelli, Emanuele Pimpini and Giorgia Rocco*

n. 69 A solution for cross-border and cross-currency interoperability of instant payment systems, *by Domenico Di Giulio, Vitangelo Lasorella, Pietro Tiberi*

n. 70 Do firms care about climate change risks? Survey evidence from Italy, *by Francesca Colletti, Francesco Columba, Manuel Cugliari, Alessandra Iannamorelli, Paolo Parlamento and Laura Tozzi*

n. 71 Demand and supply of Italian government bonds during the exit from expansionary monetary policy, *by Fabio Capasso, Francesco Musto, Michele Pagano, Onofrio Panzarino, Alfonso Puorro e Vittorio Siracusa*

n. 72 Statistics on tokenized financial instruments: A challenge for central banks, *by Riccardo Colantonio, Massimo Coletta, Riccardo Renzi*