MIS Quarterly

RESEARCH ARTICLE

DEPICTING RISK PROFILE OVER TIME: A NOVEL MULTIPERIOD LOAN DEFAULT PREDICTION APPROACH[1]

Zhao Wang
School of Management, Hefei University of Technology,
The Philosophy and Social Sciences Key Laboratory of Digital Economy and Smart Enterprise Management, Anhui Province,
Hefei, CHINA {xcwangzhao@163.com}

Cuiqing Jiang
School of Management, Hefei University of Technology,
The Philosophy and Social Sciences Laboratory of Data Science and Intelligent Society Governance, Ministry of Education,
Hefei, CHINA {jiangcuiq@163.com}

Huimin Zhao
Sheldon B. Lubar College of Business, University of Wisconsin–Milwaukee,
Milwaukee, WI, U.S.A. {hzhao@uwm.edu}

With the rapid development of fintech, the need for dynamic credit risk evaluation is becoming increasingly important. While previous studies on credit scoring have mostly focused on single-period loan default prediction, we call for a new avenue—multiperiod default prediction (MPDP)—to depict risk profiles over time. To address the challenges raised by MPDP, such as monotonic default probability prediction and complex relationship accommodation, we propose a novel approach, hybrid and collective scoring (HACS). We design a hybrid modeling strategy to predict whether and when a borrower will default separately through a default discrimination model and a default time estimation model, respectively, and synthesize them through a probabilistic framework. To accommodate various possible patterns of default time and measure the distribution of default probability over successive time intervals, we propose a joint default modeling method to train the default time estimation model. Empirical evaluations at the model (time-to-default prediction performance and discrimination performance) and mechanism (identifiability and discriminability) levels, as well as impact analyses at the application (granting performance and profitability performance) level, show that HACS outperforms the benchmarked survival analysis and multilabel learning methods on all fronts. It can more accurately predict time-to-default and provide financial institutions and investors better decision-support in granting loans and selecting loan portfolios.

Keywords: Credit risk, dynamic evaluation, multiperiod default prediction, hybrid modeling, monotonic probability, risk analysis, profit scoring

Introduction

Credit risk refers to the risk of defaulting on a debt that may arise from a borrower failing to make the required repayments (Fu et al., 2021). With the ability to depict the creditworthiness of borrowers, credit risk evaluation helps improve returns and financial stability and thus is undoubtedly of major concern for both financial institutions and individual investors. Nevertheless, due to the inevitable information asymmetry between lenders and borrowers, *credit risk evaluation* has always been a challenging problem, especially in the current fast-moving financial markets. From the market perspective, with the explosive growth of fintech, innovative business models, such as marketplace lending and crowdfunding are constantly emerging (Hendershott et al., 2021). Such financial

[1] Gediminas Adomavicius was the accepting senior editor for this paper. Nachiketa Sahoo served as the associate editor.

initiatives increase credit accessibility immensely and expand consumer groups but inevitably bring in more uncertainty (e.g., subprime borrowers) and thus greatly intensify the need to depict the dynamic[2] risk profiles of market participants over time.

From the business perspective, credit risk management runs through the entire life cycle of the loan business, and default prediction is accordingly carried out at different stages to support distinct decisions (e.g., granting loans at the pre-loan stage and risk warnings at the post-loan stage). At the pre-loan stage, deciding whether to grant credit to an application is the core goal, and it is generally supported by predicting whether a borrower will default in the full loan term. With the growth of financial markets, this goal has gradually shifted toward choosing loans or portfolios of high profitability, hence further requiring the assessment of how the credit risk of a borrower might evolve over time since the profitability of a loan depends on not only *whether* but also *when* the borrower will default. Moreover, knowing *when* a borrower will default can help lenders (e.g., banks) tap into the potential for earning growth (e.g., identifying extra profitable applications) and maintaining customer relations. For example, for a loan application with high default risk toward the end of the loan only, offering a loan with a term adjustment (e.g., extension) or an interest concession, instead of simply rejecting the application, may prevent customer attrition and also ease default concerns. It is even possible that the received loan repayments would compensate for or exceed any potential losses resulting from default if the default time is late enough. At the post-loan stage, risk monitoring and control are the core goals and are generally supported by re-predicting whether a borrower will default based on additional repayment information or passively waiting for the occurrence of delinquency. However, both of these two methods may be insufficient in practice. The former method (i.e., binary predictions) only supports indiscriminate control activities (e.g., contacting every risky customer) and may thus be costly and vulnerable to incurring insufficient or excessive interventions, as the risk level is highly time dependent. The latter is an afterthought and thus is incapable of delivering early warnings and is vulnerable to risk deterioration. A more effective method would be to differentially manage customers by mastering their risk levels at any given time (e.g., Lu et al., 2021). For example, the differential intervention may be a text reminder for customers who are risky in the long term (e.g., risky at six months) versus a call reminder for customers who are risky in the short term (e.g., risky in a month).

Regarding advanced decision support for credit risk management, c*redit scoring* is arguably the most widely used device and has drawn considerable attention in information systems (IS) research (e.g., Wang et al., 2020; Hendershott et al., 2021). Previous studies have mostly treated credit scoring as a classification problem and predicted a single default probability, indicating whether a borrower will default within a specific period. As a borrower's default involves a process that evolves over time, such single-period default prediction (SPDP) may be too impenetrable to accurately predict credit risk, especially concerning time-to-default predictions, and hence may not be sufficient to effectively support risk management decisions at both pre-loan (e.g., profit scoring) and post-loan (e.g., risk warning) stages.

As SPDP lags behind the practical needs for dynamic evaluation, we call for a new avenue, i.e., *multiperiod default prediction* (MPDP). As a more complex task, MPDP inevitably entails some challenges. We identify two essential challenges, in particular. First, borrowers always need to survive for a certain period before they can default in the next period—i.e., since the default probability of a borrower is monotonic over time, how to accommodate such monotonicity is a crucial challenge for MPDP. Second, as the information used for credit scoring is becoming more extensive and complex (Wang et al., 2020), the modeling process of credit scoring necessarily involves complex relationships (e.g., nonlinear dependencies), posing another challenge.

To address these challenges, we propose a novel MPDP approach, i.e., *hybrid and collective scoring* (HACS), in this design science research (see Gregor & Hevner, 2013). HACS consists of two components: a *default discrimination* model, predicting *whether* a borrower will default in the full observation period, and a *default time estimation* model, predicting the default probability in each observation interval (i.e., *when* a borrower will default). The two components are synthesized through a probabilistic framework, which ensures the monotonicity of the output. Such a hybrid modeling design helps to distinguish the influences of features on whether and when a borrower will default, thus allowing for the learning of a more flexible model. Further, to accommodate complex relationships, we keep base classifiers assumption free (i.e., totally data driven) and design metalearning to consolidate their predictions. HACS is distinctly different from the existing credit scoring methods that could be adapted to MPDP, including survival analysis and multi-label learning methods. Survival analysis methods can predict the monotonic default probability over time but rely on restrictive

[2] Here, "dynamic" denotes that the default risk (in terms of default probability) of a borrower changes over time and, accordingly, its colocations, such as "dynamic evaluation" and "dynamic credit scoring", refer to predicting default probability at different times (i.e., a survival/default curve), rather than that the predictions should evolve continuously as more information is added to the model. In this study, to accommodate new information generated over time, we build multiple models, each of which corresponds to a specific prediction time, pre-loan or post-loan.

assumptions (e.g., proportional hazards) and are not amenable to directly accommodating complex nonlinear dependencies. Multi-label learning methods are more flexible but cannot guarantee the monotonicity of predicted default probability over time. HACS synthesizes the monotonicity property and flexible learning ability for dynamic credit scoring.

We evaluated HACS using data from a leading online lending platform. We compared HACS with seven representative survival analysis and multi-label learning methods at three levels: *model*, *mechanism*, and *application*. From the model perspective, predicting and distinguishing bad loans from good ones is the main goal of credit scoring; thus, we evaluated the time-to-default prediction performance and discrimination performance (i.e., the ability to risk-rank borrowers accurately in different periods) at both the pre-loan and post-loan stages. From the mechanism perspective, to analyze the specific impact of monotonic default probability on each decision-making object, we carried out a case analysis. From the application perspective, considering two types of participants in financial markets and their unique goals, we carried out two impact analyses to examine the granting performance (i.e., the ability to grant loans at a low default rate) and profitability performance (i.e., the ability to select portfolios with high profits) for financial institutions and individual investors, respectively. The results show the advantages of HACS on all fronts. HACS outperformed the alternative methods in terms of time-to-default prediction performance and discrimination performance at both the pre-loan and post-loan stages. As monotonic default probability is highly desired in decision-making, HASC demonstrated superiority to the alternative methods in terms of identifiability and discriminability. Since HACS enhances granting or profitability performance, both financial institutions and individual investors could benefit from using HACS.

Literature Review

A large body of literature has explored the development and application of predictive decision support methods in the credit industry. Such methods predict the probability of default based on a set of features. The goal of such methods is pragmatic, i.e., to pursue better performance, and there are generally two ways to achieve this goal. One is mining effective features to provide additional information, such as incorporating soft information (e.g., Iyer et al., 2016; Wang et al., 2020). The other is designing a better method to accurately map features to the target variable (e.g., Abbasi et al., 2012; Wang et al., 2021). Two divergent research streams suggest that specific challenges arise in feature mining versus predictive model development. The latter is the focus of this paper.

The most prevailing approach for default prediction is classification, where each loan is classified into either creditworthy (i.e., will not default) or non-creditworthy (i.e., will default). Numerous classification methods have been used (see Lessmann et al., 2015 and Abellán & Castellano, 2017 for surveys of state-of-art classification methods for default prediction), including logistic regression (Ge et al., 2017), decision tree (Siering et al., 2016), support vector machine (Dong et al., 2018), and neural network (Siering et al., 2016). Recently, ensemble learning, with competitive and robust predictive performance, has been broadly used in default prediction (Abellán & Castellano, 2017). Although it is well-explored and continues to attract much attention, this type of default prediction method is not always sufficient, especially when dynamic credit risk prediction is desired. Thomas (2009) depicted classification-based default prediction as a connection between two snapshots (i.e., single-period): the first identifies the characteristics at a specific time (e.g., loan application time) and the second depicts the situation at a later time (e.g., six months after the loan is granted). As the pattern of borrower behavior may change over time, two snapshots may be insufficient for capturing the dynamics; thus, MPDP may be a more reasonable and practically valuable approach (Dirick et al., 2017).

When it comes to MPDP, *survival analysis* has usually been the go-to approach (see Dirick et al., 2017 for a survey of survival analysis methods in credit scoring). With the modeling of time-to-event (i.e., default) data, survival analysis can predict default probability over time, thus helping to identify not only *whether* but also *when* a borrower will default (Tong et al., 2012). Moreover, previous studies have shown the competitive predictive performance of survival analysis as compared to classification-based default prediction methods (e.g., logistic regression) and the additional benefits of survival analysis in depicting dynamic risk over time (e.g., Jiang et al., 2019), modeling censored observations (e.g., Dirick et al., 2017), and incorporating macroeconomic factors (e.g., Djeundje & Crook, 2019). Survival analysis generally implicitly assumes that all borrowers will default sooner or later, but, obviously, only a small proportion of borrowers will actually default. Consequently, a special type of survival analysis, called the mixture cure model, has been proposed to relax the assumption by modeling borrowers in terms of two distinct subpopulations (Tong et al., 2012). In one subpopulation, borrowers are cured and will never default during the lifetime of the loan, while the other subpopulation consists of borrowers who are uncured and will default at some point.

An alternative approach for MPDP is multi-label learning, which assigns a subset of labels (class) to each object. It is straightforward to adapt multi-label learning to MPDP by assigning each prediction period a label via an independent classifier, and hence some studies have used this simple

approach, i.e., binary relevance, as a benchmark for survival analysis in MPDP (e.g., Tong et al., 2012; Jiang et al., 2019). Some extended multi-label learning methods may also be applicable to MPDP along the same basic logic. The most popular way is to capture label dependencies, resulting in various methods (Baesens et al., 2005; Rivolli et al., 2020)—such as classifier chain, which learns a sequence (chain) of classifiers using other true labels on the chain, and nested stacking, which learns classifiers using other predicted labels on the chain.

In summary, research on credit scoring has identified two method families for MPDP, i.e., survival analysis and multi-label learning. Survival analysis enables stakeholders (e.g., banks and investors) to predict *dynamic* and *monotonic* default probabilities over time but has some limitations, such as linear mapping and assumption reliance. Multi-label learning enables stakeholders to predict multiperiod default probabilities that are *data-driven* (i.e., assumption-free) and *flexible* (i.e., nonlinear dependency accommodation) but also has certain limitations, such as the possibility of producing non-monotonic predictions. We strive to address these limitations.

As fintech is transforming every corner of financial services (deposits, loans, credit, fundraising, investment, and risk assessment, among others), the IS community has also started to focus on the wave of information transformation (Hendershott et al., 2021). For lending business, recent changes such as digitalization has led stakeholders (e.g., banks) to pursue more nuanced insights into business processes (e.g., *whether* and *when* a borrower will default), with the help of numerous data resources and technologies (e.g., artificial intelligence). While the IS literature has explored various types of methods for credit risk evaluation (summarized in Table 1), most existing studies have focused on SPDP, and how to accurately predict time-to-default is still an open and challenging question. We strive to extend the IS literature by bringing forth an effective approach (i.e., HACS) for predicting time-to-default.

Background

MPDP with Survival Analysis

In credit scoring, the interest of survival analysis is the failure time, T, the time of default. The survival function can be expressed as the probability of not having defaulted yet by time t:

$$S(t) = P(T > t) = \int_t^{\infty} f(u)du, \quad (1)$$

where f denotes the probability density function of T. By definition, $S(t)$ is a monotonically decreasing function of t. The scale of time t can be continuous or discrete. In credit scoring, concerns about time are generally discrete observation intervals—for example, the definition of default is generally measured in months (e.g., overdue for over one month)—thus, we focus on predicting default probabilities in discrete observation intervals (i.e., multiple periods). Note that each discrete observation interval shares one common hazard function, as repayment behaviors only occur and are observed at discrete moments. With the survival function, the default probability can be estimated under a flexible multiperiod prediction horizon. Let $\pi(t)$ be the probability function (i.e., the cumulative distribution function of failure time T) denoting the probability of having defaulted by time t. Then, $\pi(t) = 1 - S(t)$.

Methods for modelling the survival function $S(t)$ are diverse, ranging from parametric to non-parametric. Non-parametric models, such as the Kaplan-Meier estimator, usually provide only qualitative descriptions at the population level. Parametric models based on specific families of distributions may involve strict assumptions on failure times, such as Weibull. The semi-parametric models in between—e.g., the widely used Cox PH model—are more flexible since no assumptions are made on failure times. The Cox PH model consists of a non-parametric baseline survival function, describing how the survival of event per time unit changes over time at baseline levels of covariates, and a parametric part, describing how the survival probability varies in response to explanatory covariates. The survival function of the Cox PH model is given by:

$$S(t) = S_0(t)^{\exp(\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n),} \quad (2)$$

where $S_0(t)$ is the baseline survival function, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is the vector of explanatory variables, and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ is a vector of regression parameters associated with $\boldsymbol{x}$.

The formulation of the Cox PH model indicates that the hazard of an observation could change over time but the hazard *ratio* between any two observations remains constant over time (i.e., the proportional hazard assumption). This assumption can be relaxed with the accelerated failure time (AFT) model, in which explanatory variables act as acceleration factors to speed up or slow down the event process (e.g., default), but default probability at a specific prediction horizon may not always be available (i.e., the output time indicators corresponding to the predicted survival probabilities are affected by the explanatory variables and thus may differ from the desired time horizons) and the event time may be out of boundaries (e.g., default after the loan term), thus rendering the AFT model not directly applicable to MPDP.

Table 1. Representative Studies in the IS Literature on Credit Risk Evaluation

Study	Prediction object	Task	Methods
Siering et al. (2016)	Default (fraud) of founders in crowdfunding	SPDP	Support vector machine, neural network, naïve Bayes, *k*-nearest neighbors, decision tree, and majority voting ensemble
Ge et al. (2017)	Default of borrowers in P2P lending	SPDP	Logistic regression
Van et al. (2017)	Default (intentional bankruptcy) of companies	SPDP	Random logistic forests and random forests
Dong et al. (2018)	Default (fraud) of corporates	SPDP	Logistic regression, decision tree, support vector machine, and neural network
Wang et al. (2020)	Default of borrowers in P2P lending	SPDP	Logistic regression, lasso, random forests, and extreme gradient boosting
Wang et al. (2021)	Default of borrowers in P2P lending	SPDP	Hybrid strategy-based random subspace and adaptive aggregation
Fu et al. (2021)	Default of borrowers in crowd lending	SPDP	Extreme gradient boosting
Wang et al. (2022)	Default of platforms in online lending	MPDP	Logistic regression, Cox PH, mixture cure model, and random (survival) forests

There is an implicit assumption in most standard survival models, including the Cox PH model, that all borrowers will eventually default over a sufficiently long period of observation, i.e., $S(\infty) = 1$. However, in practice, most borrowers do not default over the full loan term, suggesting that some of these borrowers may be long-term survivors not susceptible to default. Hence, standard survival models have been extended to mixture cure models (Tong et al., 2012; Dirick et al., 2017), also called split hazard models (Sinha & Chandrashekaran, 1992). A mixture cure model consists of two components: an *incidence* part predicting whether a borrower will default and a *latency* part predicting the survival time of a borrower conditional on the borrower being susceptible to default. The survival function of a mixture cure model is given by:

$$S(t) = 1 - p + p * S(t|y = 1), \quad (3)$$

where y is a binary random variable defined for the default event ($y = 0$ denoting that the borrower will never default and $y = 1$ otherwise), p is referred to as *incidence* and denotes the probability that the borrower will eventually default, and $S(t|y = 1) = P(T > t|y = 1)$ is referred to as *latency* and denotes the conditional probability that the borrower survives beyond time t given that the borrower will eventually default. The latency part can be modeled by the Cox PH model. The incidence part can be modeled by logistic regression.

Both Cox PH and mixture cure models assume that covariates linearly affect the (conditional) survival probability, rendering the utilities of these models sensitive to nonlinear dependencies, i.e., interactions among covariates and nonlinear relationships between covariates and the (conditional) survival probability. Although certain strategies (e.g., adding higher-order polynomials and interactions) can help accommodate nonlinear dependencies, it is unlikely that all the nonlinear dependencies can be heuristically identified and formulated. A more reasonable and commonly used strategy would be to adaptively learn nonlinear dependencies using data-driven methods (e.g., random forests). Further, the inherent proportional hazard assumption may not strictly hold in the context of credit scoring, as explained by Dirick et al. (2017, p. 655): "For any continuous variable, e.g., age, the default hazard ratio between a 25- and a 30-year-olds is the same as the hazard ratio between an [*sic*] 70- and 75-year-olds."

To accommodate complex nonlinear dependencies, some studies have used artificial neural networks (ANN) to predict a survival curve (see Baesens et al., 2005 and Wang et al., 2019 for surveys), taking either the survival status or the hazard rate as output. Predicting the survival status using ANN essentially shares the same modeling mechanism as multi-label learning; thus, we group such methods into the family of multi-label learning, i.e., using ANN as a base learner. These methods do not guarantee that monotonic survival curves will be generated, as noted by Baesens et al. (2005, p. 1091): "The probability of a person surviving two periods could be greater than the probability to survive one period because the interdependencies of the survival probabilities over time are not properly taken into account." Predicting the hazard rate using ANN could indirectly estimate monotonic survival curves through the use of the Kaplan-Meier estimator but, at the same time, would inherit the implicit assumption that all subjects will eventually default, hence reducing the discriminative ability for long-term survivors and failing to capture the heterogeneous effects of covariates on whether and when a borrower will default (one type of important complex relationship in MPDP). In summary, a method with both flexible modeling and label dependency accommodation capabilities would be highly desirable.

MPDP with Multi-Label Learning

Multi-label learning is a classification variant for which multiple labels can be assigned to each observation. By treating the default status (in default or not in default) in each prediction period as a label to learn, multi-label learning can be naturally adapted to MPDP. Let $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ be the binary vector of default statuses and the subscripts correspond to the time vector $(t_1, t_2, \ldots, t_m)$, with $y_i = 1$ denoting that the borrower will default during a time period $(0, t_i)$ and $y_i = 0$ otherwise. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ be the vector of features. Then, the goal of MPDP with multi-label learning is to induce a model that maps the inputs $\boldsymbol{x}$ to the binary vector $\boldsymbol{y}$ and can properly predict the default statuses of a borrower in multiple periods.

The most straightforward multi-label learning method to tackle MPDP is *binary relevance* (BR), which decomposes a MPDP task with m periods into m independent binary classification tasks (as illustrated in Figure 1). BR trains m independent models and the default probabilities are predicted independently of each other. BR has several merits. First, it builds an ensemble of binary classifiers, and various classification algorithms can be intactly used as base learners. Second, nonlinear relationships between features and labels can be easily accommodated by nonlinear models automatically. Third, by decomposing the MPDP task into multiple subtasks, BR tries to depict the credit profile of a borrower synthetically from multiple views (i.e., credit profile in different periods), making it easier to achieve robust predictive performance. Based on previous studies, it seems likely that BR would yield competitive performance, as compared to survival analysis (Tong et al., 2012).

Nevertheless, BR totally ignores label dependencies. Considering the interdependencies among the labels could help enhance the performance of multi-label learning (Baesens et al., 2005). Such label dependencies may be even more salient for MPDP, since there are clear time dependencies among the different periods. For any two observation times $t_i < t_j$, if we know that the borrower defaulted in the period $(0, t_i)$, then the label for a wider time period $(0, t_j)$ will be positive too, i.e., $y_i = 1 \rightarrow y_j = 1$. To capture label dependencies, learning classifiers that *condition* the prediction of a label on not only the feature set $\boldsymbol{x}$ but also some of the other labels may be useful. The idea of conditioning can be realized in different ways, including *classifier chain* (CC) and *nested stacking* (NS).

CC trains base classifiers for labels following an order on the label set, i.e., a chain of labels (as illustrated in Figure 2). Specifically, each base classifier (except the first one) is trained using not only the feature set $\boldsymbol{x}$ but also the true label information of the previous node in the chain. Although different chains, in terms of label order, should theoretically be equivalent, the outputs of CC, are commonly sensitive to the order of the chain and performance varies accordingly; hence, in practice, multiple chain orders (e.g., forward and backward) are commonly tried and compared. In the prediction phase, when a new borrower arrives and needs to be scored, a prediction $\hat{\boldsymbol{y}} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m)$ is produced by sequentially implementing each trained base classifier. Since the true values of $(y_1, y_2, \ldots, y_{m-1})$, which are used as additional features for training base classifiers $(h_2, h_3, \ldots, h_m)$, respectively, are not available at the time of prediction, they are commonly replaced by their respective predictions. While feasible, such replacements may violate the essential assumption in prediction that the future will resemble the past. More formally, the distribution of the true labels is theoretically different from that of the predicted values; thus, training data are not representative of testing data, resulting in potential prediction bias. This problem can be prevented by expanding the feature space using the predictions $\hat{y}$ instead of the true labels (i.e., the NS method).

NS is equivalent to CC, other than using the predicted labels $\hat{\boldsymbol{y}}$ instead of the true labels $\boldsymbol{y}$ for expanding the feature set in the training phase (as illustrated in Figure 3). NS builds two layers of base classifiers (i.e., the label layer and the predicted label layer), and the feature space of the base classifier for y_i (except the first node) in the label layer consists of $(\boldsymbol{x}, \hat{y}_{i-1})$, in which $\hat{y}_{i-1}$ can be obtained from the correspondingly previous node in the predicted label layer, thus forming a nested structure. Note that although NS can avoid using the true label information, which is not available at the time of prediction, it may, however, fail to capture the true interdependencies among the labels.

Overall, the above-mentioned multi-label learning methods (i.e., BR, CC, and NS) have several merits, both common and unique. For example, they can take advantage of advanced classification algorithms to improve predictive performance and are well-positioned to fit complex nonlinear relationships. As such, they show competitive performance in several contexts. When there are high label dependencies, CC and NS may outperform BR. However, it is arguable that none of the existing multi-label learning methods can guarantee the monotonicity of the outputs, which is essential for MPDP. From a probabilistic perspective, the default probabilities in different periods follow the addition rule:

$$\pi(t_i) = P(T \le t_i) = P(T \le t_{i-1}) + P(t_{i-1} < T \le t_i) \geq \pi(t_{i-1}). \quad (4)$$

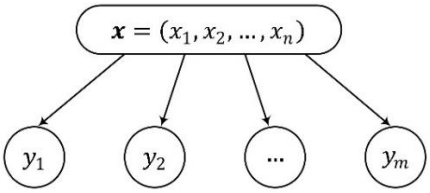

Figure 1. Binary Relevance

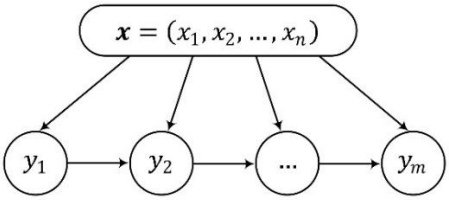

Figure 2. Classifier Chain

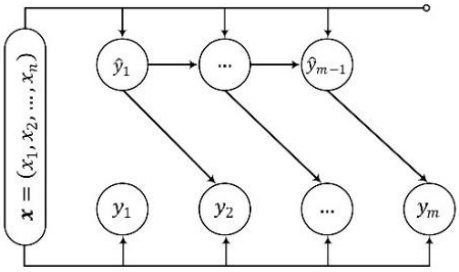

Figure 3. Nested Stacking

Thus, the default probabilities over periods are supposed to monotonically increase, but the outputs of base classifiers in multi-label learning, however, cannot strictly satisfy this property although some of the multi-label learning methods (e.g., CC and NS) do consider the label dependencies in the training phase. For example, a linear multi-label learning method may give a few features lower weights to the model at t_i than that at t_{i-1} due to its *separate modeling*, and when these features dominate the default probability of an observation, non-monotonic prediction may occur. This will greatly limit the practical value of such methods for MPDP. A non-monotonic default probability curve, on the one hand, fails to meet the basic principle of probability theory and thus weakens the reliability, and on the other hand, cannot accurately estimate the occurrence time of the default event according to a preset cut-off value (e.g., multiple points of intersection may occur) and thus will be inflexible to depict the dynamic credit risk of borrowers. Note that monotonicity may be a specific challenge for MPDP and becomes irrelevant when it comes to SPDP.

Proposed Hybrid and Collective Scoring Approach

Both survival analysis and multi-label learning inevitably have some intrinsic limitations in terms of MPDP, and a method that simultaneously has the flexibility to fit complex relationships (like in multi-label learning) and the property of monotonicity in the predicted default probability over time (like in survival analysis) would be highly desirable. Our design of the *hybrid and collective scoring* (HACS) approach to MPDP derives from this main motivation.

The basic idea behind HACS (outlined in Figure 4) is to first distinguish defaulting borrowers from non-defaulting borrowers and then further identify the observation interval that the default event will fall into. Accordingly, HACS consists of two components: a *default discrimination* model predicting the probability that the borrower will default during the entire observation period and a *default time estimation* model estimating the probability that the borrower will default in each observation interval, conditional on being susceptible to default.

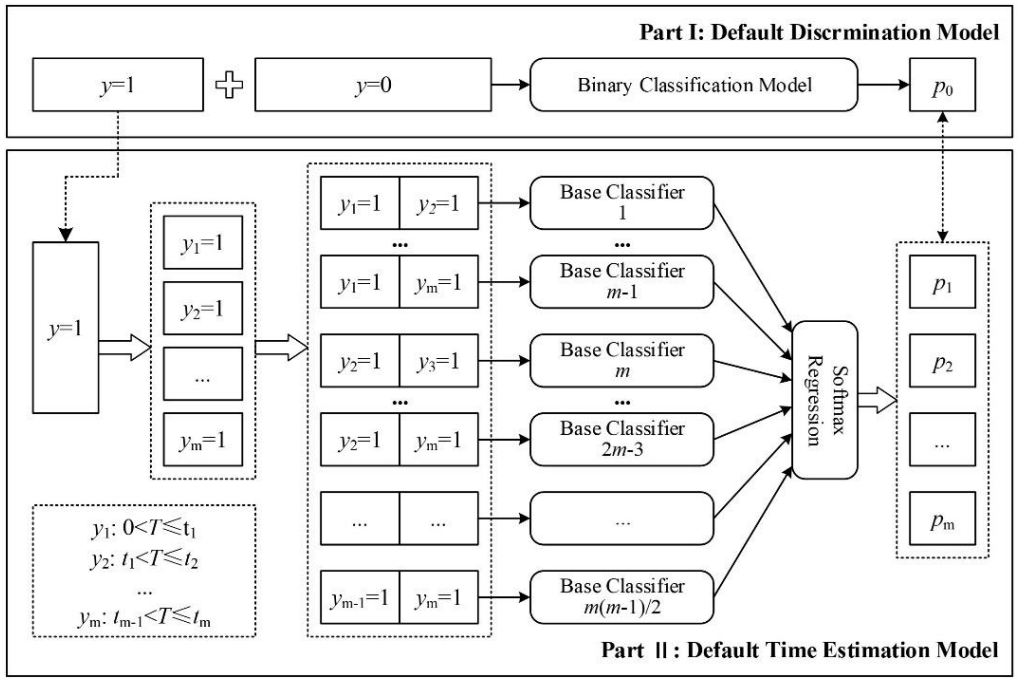

Figure 4. The Proposed HACS Approach

Let y be a binary random variable defined for the default event, with $y = 1$ denoting that the borrower will default during the entire observation period (e.g., the loan term) and $y = 0$ otherwise. Let $\boldsymbol{t} = (t_1, t_2, \ldots, t_m)$ be a time vector denoting the observation times, i.e., the prediction horizon. Then, the entire observation period can be divided into m intervals: $(0, t_1), (t_1, t_2), \ldots, (t_{m-1}, t_m)$. The goal of HACS is to induce, from training data, a probability function defined for MPDP as follows:

$$\pi(t) = p_0 \times \sum_{t_i \le t} p_{i,} \tag{5}$$

where $p_0 = P(y = 1)$ is referred to as the *full* default probability and denotes the probability that the borrower defaults during the entire observation period, and $p_i = P(t_{i-1} < T \le t_i | y = 1)$ is referred to as an *interval* default probability and denotes the probability that the borrower defaults in the i^{th} observation interval (t_{i-1}, t_i) conditional on being susceptible to default. For example, the probability that a borrower will have defaulted by time t_2 is estimated by multiplying the full default probability and the sum of the first two interval default probabilities, i.e., $p_0 \times (p_1 + p_2)$. Note that we model the default probability at all m periods, rather than m-1 periods, since stakeholders may be interested in the risk ranking of each costumer at any given time in order to support their risk management decision-making in different market and policy environments.

Accordingly, two models will be trained in HACS: a default discrimination model for predicting the full default probability p_0 and a default time estimation model for predicting the interval default probabilities p_i. Figure 5 presents the HACS procedure.

Default Discrimination Model

The default discrimination model is essentially a binary classification model (as shown in Figure 4) that tries to induce, from a training set consisting of positive ($y = 1$) and negative ($y = 0$) instances, a hypothesis $h: \boldsymbol{x} \to y$, which could properly distinguish defaulting borrowers from non-defaulting borrowers (Line 1 in Figure 5). Numerous binary classification models have been used in credit scoring, constituting a valuable repository for the default discrimination model. They can be simply divided into two families: linear and nonlinear. For linear models, the most widely model used in credit scoring in both research and practice is arguably logistic regression (e.g., scorecard). For nonlinear models, representatives include random forests and gradient-boosted trees. In light of learning theory (i.e., bias-variance decomposition), random forests grow multiple unpruned trees and mitigate the detrimental effect of variance through model averaging. Similarly, gradient-boosted trees reduce variance through model averaging and distinctively reduce bias through consecutively building incremental models.

Parameters:
L: base classifier learner;
k: number of folds for the cross validation in meta-learning.
Inputs:
$S = S_G \cup S_B$: training set, with each case including:
$\boldsymbol{t} = (t_1, t_2, \ldots, t_m)$: observation time vector;
$\boldsymbol{st} = (st_1, st_2, \ldots, st_m)$: observation status vector;
$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$: feature set.
Outputs:
f_0: default discrimination model;
f_{ij} $(i = 1,2, \ldots, m-1;\ \ j = i+1, \ldots, m)$: base classifiers;
f_{meta}: meta model.

1 $f_0 = L(st_m \sim \boldsymbol{x}, \text{data} = S)$;
2 $S_B \rightarrow (S_1, S_2, \ldots, S_m)$;
3 $\boldsymbol{t}, \boldsymbol{st} \rightarrow (y_1, y_2, \ldots, y_m)$;
4 for i in $\{1:(m-1)\}$ do
5 for j in $\{(i+1):m\}$ do
6 $S_{ij} = S_i \cup S_j$;
7 $f_{ij} = L(y_i \sim \boldsymbol{x}, \text{data} = S_{ij})$;
8 $\text{pred}, S_{\text{meta}} = \emptyset$;
9 $t_{\text{d}} = \min\{t_i | st_i == 1\}$;
10 $\boldsymbol{z} = (z_{1,2}, z_{1,3}, \ldots, z_{1,m}, z_{2,3}, z_{2,4}, \ldots, z_{2,m}, \ldots, z_{m-1,m})$;
11 $\text{ind} = \text{Sample}(k, \text{nrow}(S_B), \text{replace} = \text{True})$;
12 for d in $\{1:k\}$ do
13 $OOS_{\text{train}} = S[!\,\text{ind} == d,]$;
14 $OOS_{\text{test}} = S[\text{ind} == d,]$;
15 $OOS_{\text{train}} \rightarrow (S_1, S_2, \ldots, S_m)$;
16 for i in $\{1:(m-1)\}$ do
17 for j in $\{(i+1):m\}$ do
18 $S_{ij} = S_i \cup S_j$;
19 $\text{classifier}_{ij} = L(y_i \sim \boldsymbol{x}, \text{data} = S_{ij})$;
20 $\text{pred}_{ij} = \text{Predict}(\text{classifier}_{ij}, \text{newdata} = OOS_{\text{test}})$;
21 $\text{pred} = \text{RowBind}(\text{pred}, \text{pred}_{ij})$;
22 $S_{\text{meta}} = S_{\text{meta}} \cup \text{pred}$;
23 $f_{\text{meta}} = \text{SoftmaxRegression}(t_{\text{d}} \sim \boldsymbol{z}, \text{data} = S_{\text{meta}})$.

Figure 5. Procedure of HACS

Default Time Estimation Model

The default time estimation model is designed to estimate interval default probabilities, $p_i = P(t_{i-1} < T \leq t_i | y = 1)$. Since time-to-default is a continuous status rather than a dichotomous indicator, accurately estimating the default probability in each observation interval is nontrivial. The main challenges are threefold. First, the credit risk level of a borrower, specifically a default borrower in the default time estimation model, changes over successive observation intervals, forming various possible patterns (e.g., escalating and defaulting in the short run and moderately increasing and defaulting after a long time). Thus, it is difficult to discriminatively represent the characteristics of defaulters in different periods with a single model, especially a single linear model. Second, when multiple models are involved, how to train each base model (e.g., the one-against-all strategy as in multi-label learning) to effectively differentiate defaulters in different observation intervals is not straightforward. Third, after multiple models are trained, a novel mechanism still needs to be designed to integrate (e.g., adaptively) the outputs of the base models to further improve the predictive performance and simultaneously guarantee that the integrated results satisfy the addition rule (Equation 4).

We propose a novel method, *joint default modeling* (JDM), to train the default time estimation model. Along with the illustration and procedure of HACS in Figures 4 and 5, we explain JDM in detail, which consists of three phases: *default time binarization* (Lines 2-3 in Figure 5), *round-robin default modeling* (Lines 4-7 in Figure 5), and *metalevel multiperiod default modeling* (Lines 8-23 in Figure 5).

Phase 1 (*default time binarization*): A straightforward way to estimate the default time would be using regression methods with the default time as the target variable; however,

this simple strategy suffers from several deficiencies. First, a regression model cannot include both event and time aspects as the outcome. Second, direct regression on the default time cannot generate a survival curve, which is imperative in identifying whether and *how* a borrower becomes a defaulter over time. Third, as mentioned earlier, it is difficult for a single model to effectively learn multiple patterns. A more reasonable strategy (referred to as default time binarization) would be to map the entire observation period into multiple (mutually exclusive) intervals and identify whether the borrower will default in each observation interval.

A departure of default time binarization, and thus also of JDM, is that it only concerns the subset of default instances S_B from the original training set S, as the default time is conditional on the loan being susceptible to default. For m observation intervals $\{(0, t_1), (t_1, t_2), \dots, (t_{m-1}, t_m)\}$, let $\boldsymbol{y} = (y_1, y_2, \dots, y_m)$ denote whether a borrower defaults in each observation interval, separately. Then, S_B can be divided into m defaulter groups accordingly, i.e., $S_B \rightarrow (S_1, S_2, \dots, S_m)$, where group S_i includes all the instances in S_B that default in the i^{th} observation interval (t_{i-1}, t_i), i.e., $y_i = 1$. Note that the definition of $\boldsymbol{y}$ here is different from that in multi-label learning: the definition here in JDM is the label in an observation interval, whereas the definition in multi-label learning is the label over a time horizon (since the observation starting time).

Phase 2 (*round-robin default modeling*): Having obtained the collection of defaulter groups, we need to collectively train classifiers for the corresponding observation intervals. Based on previous studies, there are two possible strategies for this purpose, i.e., one-against-all and one-against-one. As illustrated in Figure 6 (in a two-dimensional reduced feature space), the one-against-all strategy treats one defaulter group (*circle* group) as positive examples and all other groups as negative examples for each base classifier, whereas the one-against-one strategy considers two defaulter groups (*circle* group vs. *plus* group) at a time for each base classifier.

In comparison, the decision boundaries in the one-against-one strategy are obviously simpler than those in the one-against-all strategy and are thus more likely to lead to robust base classifiers. Moreover, the more complex the decision boundaries, the more data would be needed to fit them. Therefore, for JDM, we adopt the one-against-one strategy for round-robin default modeling. Specifically, for each pair of (mutually exclusive) defaulter groups, S_i and S_j ($i = 1,2, \dots, m-1$; $j = i+1, \dots, m$), a joint set ($S_i \cup S_j$) is generated and accordingly a base classifier is trained, resulting in $m(m-1)/2$ base classifiers in total. The choice of the algorithm used to train each base classifier can be flexible. Since the correlation across base classifiers is relatively low due to the fact that they are trained using different pairwise defaulter groups, the variance reduction effect would be expected with the subsequent model ensembling; hence, a low-bias learner may be preferred to further improve predictive performance.

Phase 3 (*metalevel multiperiod default modeling*): Each of the base classifiers trained in Phase 2 can discriminate between two groups of defaulters (i.e., part of task); therefore, we need to combine the predictions of the multiple base classifiers and learn to address the whole task, i.e., interval default probabilities. The general method for this purpose through a separate trainable model is commonly referred to as *stacking*. Its basic idea is to use the predictions of base classifiers as intermediate predictions (called metadata), and then use them to train a model at the metalevel. In our case, the targets at the base level (a binary default indicator) and metalevel (a set of multiclass default time intervals) are heterogeneous. Thus, the output of each base classifier is more like a real feature in a specific view for training the metamodel. Such difference certainly calls for low-bias, even unbiased, metadata (e.g., out-of-sample predictions), yet testing the base classifiers directly on training data to generate metadata obviously biases the results since the base classifiers have already seen the training data (Murphy, 2012). Therefore, in JDM, we generate the metadata through a *k*-fold cross-validation. Figure 7 illustrates the procedure of metalearning. Such unbiased estimation of "features" (i.e., predictions of base classifiers) could avoid error propagation present in the classical stacking framework. Note that the base classifiers trained in metalearning (Phase 3) are only used for generating unbiased predictions of themselves, whereas the base classifiers for real predictions have been trained separately using the entire training set (Phase 2).

For the metamodel, we use softmax regression (a generalization of logistic regression) to estimate the interval default probabilities. On the one hand, the outputs of softmax regression are independent and additive and nicely sum up to one. Such properties are exactly what the interval default probabilities need to have. On the other hand, in light of previous studies on ensemble learning (Abellán & Castellano, 2017), base classifiers are often complex and diverse (i.e., low-bias and high-variance) whereas the metamodel is often simple and smooth, providing robust predictions. The metamodel through softmax regression is given by:

$$p_i = P(y_i = 1) = \frac{\exp(\alpha_{i0} + \alpha_{i1}x_1 + \alpha_{i2}x_2 + \cdots + \alpha_{in}x_n)}{\sum_{k=1}^{m} \exp(\alpha_{k0} + \alpha_{k1}x_1 + \alpha_{k2}x_2 + \cdots + \alpha_{kn}x_n)}, \quad (6)$$

where $\boldsymbol{\alpha}_i = (\alpha_{i0}, \alpha_{i1}, \dots, \alpha_{in})$ denotes a vector of coefficients for the i^{th} observation interval, which are estimated by minimizing the average of all cross-entropies over training instances.

(a) One-against-All (b) One-against-One

Figure 6. Learning Strategies for a Four-Period MPDP Problem

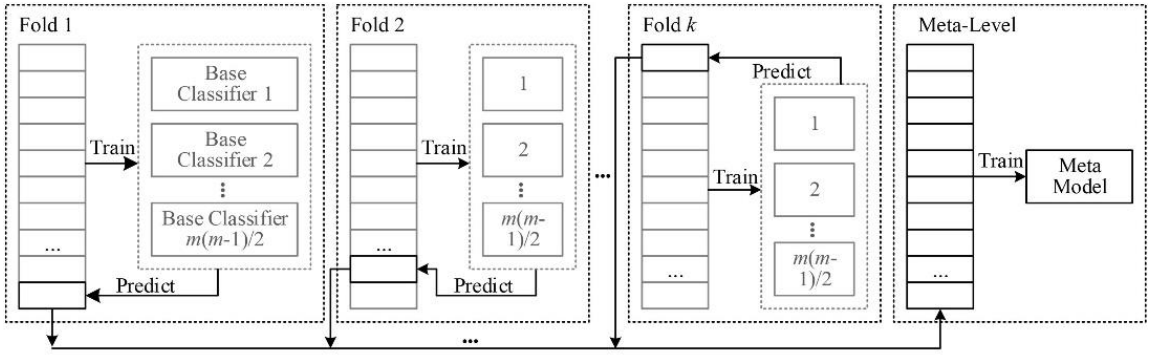

Figure 7. Procedure of Metalearning

Table 2. Comparison of HACS with Related Methods

Property	COX	MCM	RSF	MTLSA	BR	CC	NS	HACS
Direct nonlinear modeling			√		√	√	√	√
Monotonic outputs	√	√	√	√				√
Flexible base learner	NA	NA		NA	√	√	√	√
PH assumption-free			√	√	√	√	√	√
If-and-when separation		√						√
Label dependency accommodation	√	√	√	√		√	√	√
Between-group discrimination	√	√	√					√
Error propagation avoidance	NA	NA	NA	NA	√			√
Attribute noise avoidance	NA	NA	NA	NA	NA		√	√

Note: Attribute noise denotes that the attributes (e.g., metafeatures) used in modeling vary across training and testing phases, i.e., the "clean-training data vs. noisy test data" case. MCM: mixture cure model; MTLSA: multitask learning-based survival analysis (Li et al., 2016); RSF: random survival forest (Wang et al., 2022). √: yes; NA: not applicable.

Contrast with Related Methods in the Existing Literature

Table 2 contrasts HACS against related methods from existing credit scoring research, survival analysis research, and multi-label learning research. Compared with the extant credit scoring research, this study is one of the few that recognize the importance and benefits of MPDP. Most loans have a nature of multiple periods, leading to multiperiod credit risks for lenders, which should be adequately modeled. The literature on credit scoring offers few methodological tools for MPDP, other than directly borrowing survival analysis from the field of medical science, rendering the predictive performance highly dependent on the generalizability of the survival method used. We hence propose a new modeling paradigm for MPDP, which fuses statistical modeling (e.g., the design of output function) and machine learning (e.g., round-robin modeling and metalearning), thereby opening up a new avenue for credit scoring research.

Compared with survival analysis methods, HACS can be seen as an enhanced survival analysis tool and can be applied to solve discrete-time survival analysis problems. Theoretically, the output function of HACS, i.e., $\pi(t_i)$, is equivalent to the cumulative distribution function in survival analysis. With the same target function, HACS makes no assumption on the failure time or other factors, whereas standard survival analysis methods mostly rely on specific assumptions. For example, the main assumption of Cox PH is the proportionality of hazards. However, if this assumption is violated, Cox PH is less likely to yield the desired performance. Moreover, the assumption that the event of interest will eventually occur is implicitly embedded in most standard survival analysis methods, yet long-term survivors are ubiquitous in the financial market. In such cases, HACS provides a more flexible tool for survival data analysis.

Compared with multi-label learning methods, HACS can be treated as an effective multi-label learning tool in contexts that are subject to monotonicity constraints. For example, in churn prediction (e.g., Martens et al., 2016), the probability of customer churn is generally supposed to increase over time and should thus be modeled as a monotonic function of time. Too often, such constraints are ignored in both research and practice, especially when multi-label learning is used. HACS transcends classical multi-label learning methods by honoring the monotonicity property of the output while preserving several merits, such as label dependency accommodation, error propagation avoidance, and attribute noise avoidance.

Empirical Evaluation

Data

We evaluated HACS on a dataset from a major online lending platform in China. The platform provided 10% of its real business data between January 1, 2015 and February 22, 2017, including the loan application information and the monthly repayment information (i.e., term duration of loan, scheduled repayment amount, due date, repayment date, and repayment status). To obtain the entire loan status and sufficient observations, we used data on 12-month loans, which make up the majority of loan observations. After removing observations with incomplete repayment information (i.e., loans that had not been paid off or defaulted on by February 22, 2017), the dataset used in our evaluation consisted of 34,679 loan observations. A loan observation is considered to be in default when one repayment has been overdue for more than 30 days. Accordingly, there were 32,181 observations of loans not in default and 2,498 observations of loans in default in the dataset (7.2% default rate). Table 3 summarizes the attributes available in the loan application information. Categorical attributes were all transformed into dummy features, except for the platform-assigned grade. Since it is on an ordinal scale, we treated the platform-assigned grade as a continuous attribute and coded grades of A to F as 1 to 6, respectively.

Our experiment covered both pre-loan and post-loan credit risk evaluation. At the pre-loan stage, the goal of MPDP is to predict dynamic default probabilities at the time of application. As our dataset only contained approved applicants and the evaluation objects included all (approved and rejected) applicants, sample selection bias may have occurred. However, previous studies have empirically suggested that "there is only modest scope for model improvement in considering the behavior of rejected applicants" (Banasik et al., 2003, p. 831) and, "at least in this consumer credit setting, the resulting benefits from determining the true outcome values of the rejected cases are low" (Verstraeten & Van, 2005, p. 989). In light of such insights and data availability, most research and practice simply use a sample of approved applicants for training credit scoring methods (e.g., Iyer et al., 2016; Ge et al., 2017; Wang et al., 2020). We first followed this mainstream routine and then further examined the effects of sample selection bias on the predictive performance of MPDP methods using the same strategy used by Verstraeten and Van (2005), in which marginally accepted applicants were treated as rejected. Note that the effects of sample selection bias may vary across contexts and rejection inferences (e.g., Shen et al., 2020) may be necessary if such bias could significantly damage predictive performance.

At the post-loan stage, the goal of MPDP is to predict dynamic default probabilities at a specific time after loan issuance. As evaluation objects are simply approved applicants, our dataset does not suffer from sample selection bias for post-loan predictions. Besides, the scope of criteria relevant to decision-making is much larger at this stage, as empirical information about repayment behavior regularly accumulates. Specifically, based on monthly repayment information prior to the prediction point, we constructed three additional features: (1) number of times in delinquency (i.e., being late on installment repayment); (2) cumulative number of days in delinquency; and (3) maximum number of days in delinquency for a repayment. These features were then used to complement the typical features (summarized in Table 3) in MPDP methods. Note that although the new evidence presented in the post-loan stage is *dynamic*, since its utility is typically partial or even complementary and the major evidence for MPDP is still *static* information at the pre-loan stage, sequential approaches (e.g., LSTM) may not be intuitive options.

Table 3. Attributes Used in Analysis

No.	Attribute	Summary statistics			
	Continuous	***Min***	***Max***	***Mean***	***SD***
1	Amount of loan (RMB)	100	235,000	5,044.61	3,941.97
2	Interest rate (%)	10	24	21.31	1.73
3	Age (years)	18	56	29.59	6.51
4	Number of successful loan applications	0	38	1.59	2.38
5	Amount of successful loan applications (RMB)	0	334,000	6,237.92	11,477.08
6	Amount to be repaid (RMB)	0	74,360.78	2,524.84	4,112.42
7	Number of historical on-time repayments	0	182	6.53	11.63
8	Number of historical overdue repayments	0	35	0.32	1.25
	Categorical	***Distribution* (%)**			
9	Gender	Male (74.62), Female (25.38)			
10	Platform-assigned grade	A (1.77), B (4.76), C (24.97), D (57.43), E (10.59), F (0.49)			
11	Type	Flash loan (12.22), Normal (65.27), Other (22.51)			
12	Is the first loan on the platform	Yes (46.01), No (53.99)			
13	Has mobile verification	Yes (67.04), No (32.96)			
14	Has household verification	Yes (6.01), No (93.99)			
15	Has video verification	Yes (9.25), No (90.75)			
16	Has education verification	Yes (37.80), No (62.20)			
17	Has credit verification	Yes (4.18), No (95.82)			

Experiment Design and Performance Evaluation

We evaluated HACS in comparison with benchmarked methods from two families, i.e., survival analysis and multi-label learning. For survival analysis, we used Cox, MCM, MTLSA (Li et al., 2016), and RSF (Wang et al., 2022). For multi-label learning, we used BR, CC, and NS. Considering that the chain order may affect the performance of CC and NS, we set up two types of chain orders, i.e., forward (FD) (i.e., gradually extending the prediction horizon) and backward (BD) (i.e., gradually shortening the prediction horizon), resulting in four combinations (CC_FD, CC_BD, NS_FD, and NS_BD). For the base classifier learner in HACS and multi-label learning methods, we applied logistic regression (LR) as a representative of linear classifiers, and random forests (RF) and extreme gradient boosting (XGB) as representatives of nonlinear classifiers, resulting in 18 combinations—Three base classifier learners (LR, RF, and XGB) × Six MPDP methods (BR, CC_FD, CC_BD, NS_FD, NS_BD, and HACS). We kept the default parameter settings of all benchmarked methods. For HACS, we used 10-fold cross-validation in the metalearning (i.e., k =10). We also examined the effects of an additional base classifier (i.e., ANN) and parameter tuning as a robustness check.

We first compared the above-mentioned methods in terms of their ability to predict *time-to-default*. Specifically, we computed the C-index and integrated the time-dependent Brier score (IBS), which have been widely used for time-to-event outcomes (Wang et al., 2019), based on the monthly predictions of each method. The C-index evaluates whether a higher risk score is associated with a shorter survival time and is defined as the number of concordant pairs divided by the number of comparable pairs (Wang et al., 2019). The Brier score evaluates the accuracy of a predicted survival function at a given time and is defined as the average squared distance between the observed survival status and the predicted survival probability (Wang et al., 2019). IBS provides an aggregate measure of the Brier score over all available times. Typically, the larger the C-index and the smaller the IBS, the better the performance. We also estimated the default probability in each time interval and accordingly examined the performance in predicting defaulters in each month.

We then compared selected top-performing methods in terms of their *discrimination* performance (i.e., the ability to risk-rank borrowers accurately) over the one-year loan term. Specifically, we selected four time horizons, i.e., 3, 6, 9, and 12 months, respectively, to give a comprehensive comparison (we caution that these representative time horizons were selected for a pragmatic purpose only). For the pre-loan stage, the prediction time was naturally set as the loan issuance time. For the post-loan stage, we selected the third month (t_1) and sixth month (t_2) as two representative prediction times and predicted the probability of default over the remaining loan term (e.g., $t_3|t_1$). We used AUC, Kolmogorov–Smirnov (KS), and H-measure (Hand, 2009) for gauging discrimination performance.

To estimate the out-of-sample performance of each method, we performed 10 independent 10-fold cross-validations, resulting in 100 performance estimates, to get a robust result. Performance results (mean and its 95% confidence interval) reported later are all based on the 100 estimates. The confidence

interval of each mean was estimated by *t*-value times standard error (i.e., $\bar{x} \pm t_{n-1,\alpha/2} * \frac{\sigma}{\sqrt{n}}$). For a fair comparison between methods, the partitioning of folds was kept identical across all methods during each 10-fold cross-validation.

Finally, we conducted a case analysis to examine the differences between methods with (i.e., HACS and survival analysis) and without (i.e., multi-label learning) the monotonicity property of outputs in terms of the ability to accurately identify the default time (i.e., *identifiability*) for specific loans. We further examined the *discriminability* of the methods with the monotonicity property in identifying the default time by comparing the distances between their predicted probabilities before and after the default time.

Performance in Time-to-Default Prediction

We first examined: *Whether and how much HACS contributes to performance improvement over multi-label learning and survival analysis methods in the time-to-default prediction.* Tables 4 and 5 summarize the time-to-default prediction performance of HACS versus benchmarked methods in terms of the C-index and IBS, respectively. The results of vanilla HACS versus censoring-adapted HACS are available in Appendix A.

Overall, HACS outperformed all multi-label learning and survival analysis methods at both stages and in terms of both performance metrics. Across the two stages, the predictive performance of each method at the post-loan stage was better than that at the pre-loan stage, indicating that incorporating post-loan repayment information contributed to performance improvement in the time-to-default prediction. Further, at the post-loan stage, predictive performance at t_2 was better than that at t_1, further indicating that the more post-loan repayment information available, the better each method (except MTLSA) can predict the time to default. Among the three types of base classifiers, RF and XGB performed somewhat comparably, both better than LR, for all MPDP methods. For example, HACS with RF gave the best performance at the pre-loan stage in terms of both performance metrics, whereas HACS with XGB gave the best performance at the post-loan stage (t_2) in terms of both performance metrics. Interestingly, for the multi-label learning methods, although not capturing label dependencies, BR was competitive with and often better than CC and NS; further, the backward chain order (CC_BD and NS_BD) often gave the worst performance, indicating that the error propagation effect may overshadow the performance lifting effect in CC and NS. The results show that HACS indeed contributed to performance improvement in the time-to-default prediction, from the aspects of both discrimination (i.e., C-index) and calibration (i.e., IBS).

We tested the statistical significance of the comparisons between HACS and benchmarked methods using a non-parametric Friedman test. Since the means and confidence intervals of performance measures (C-index and IBS) clearly uncovered competitive methods in each family (i.e., BR with a nonlinear base classifier; MCM, RSF, and HACS with a nonlinear base classifier), we focused on these methods in the significance testing and the subsequent analyses. Tables 6 and 7 summarize the results of full pairwise comparisons of the six methods at the pre-loan and post-loan stages, respectively. Since the Friedman test is a kind of rank-sum test, the results across performance metrics (C-index and negative transformed IBS) and prediction horizons (t_1 and t_2 at the post-loan stage) were pooled together. Overall, the differences across the six MPDP methods were statistically significant at both the pre-loan (χ^2 = 744.391, p < 0.001) and post-loan (χ^2 = 976.803, p < 0.001) stages. Further pairwise comparisons show that HACS statistically significantly outperformed all benchmarked methods at both stages.

We further examined the effect size of using HACS (with RF and XGB) in lieu of each benchmarked method in terms of performance improvement in the time-to-default prediction. We performed repeated-measure ANOVA with method (two-level, i.e., HACS vs. one of the benchmarked methods, respectively) as a main factor and stage (three-level, i.e., pre-loan stage and t_1 and t_2 at the post-loan stage) as a between-subject factor. Figure 8 shows the partial η^2 of the main factor. The results show that using HACS with either RF or XGB in lieu of multi-label learning methods (BR_RF and BR_XGB) and survival analysis methods (MCM and RSF) accounted for conspicuous power (i.e., variance)—with all being over 0.4—in performance improvement in the time-to-default prediction.

In addition to commonly used performance metrics (i.e., C-index and IBS), we also formulated an evaluation task, i.e., predicting default at each time interval (month), to further evaluate the performance of each method in the time-to-default prediction. For each method, we calculated the default probability in each month, i.e., the probability that a borrower will survive over *t* months and default within the next month (e.g., 4|3M). The score function was derived as $(p_t - p_{t-1})/(1 - p_{t-1})$, where p_t denotes the default probability at time horizon *t*. Table 8 summarizes the predictive performance of each method in terms of average AUC across all months. The number of months was 12, 9, and 6 for the pre-loan, post-loan (t_1), and post-loan (t_2) stages, respectively. The results show that HACS with RF outperformed the multi-label learning (BR) and survival analysis methods (MCM and RSF). Interestingly, as the default probabilities over different time horizons are associated when predicting default in a time interval, the methods with monotonicity property (MCM, RSF, and HACS) outperformed those without the monotonicity property (BR), indicating that temporal dependency (e.g., monotonicity) may be a desirable property of time-to-default prediction.

Table 4. Performance of Time-to-Default Prediction in Terms of C-Index

Method	Base classifier	Pre-loan stage	Post-loan stage (t_1)	Post-loan stage (t_2)
BR	LR	0.654 (0.651-0.658)	0.743 (0.740-0.747)	0.781 (0.777-0.785)
CC_FD	LR	0.652 (0.649-0.656)	0.743 (0.739-0.746)	0.787 (0.782-0.791)
CC_BD	LR	0.549 (0.545-0.552)	0.562 (0.558-0.566)	0.568 (0.562-0.573)
NS_FD	LR	0.654 (0.651-0.657)	0.743 (0.740-0.747)	0.779 (0.774-0.783)
NS_BD	LR	0.654 (0.650-0.657)	0.743 (0.739-0.746)	0.779 (0.775-0.783)
HACS	LR	0.657 (0.655-0.660)	0.747 (0.743-0.750)	0.787 (0.783-0.792)
BR	RF	0.685 (0.682-0.688)	0.749 (0.745-0.752)	0.784 (0.779-0.788)
CC_FD	RF	0.655 (0.652-0.659)	0.724 (0.720-0.728)	0.763 (0.759-0.768)
CC_BD	RF	0.653 (0.650-0.656)	0.723 (0.719-0.727)	0.759 (0.754-0.765)
NS_FD	RF	0.665 (0.661-0.669)	0.724 (0.720-0.728)	0.760 (0.755-0.764)
NS_BD	RF	0.665 (0.661-0.669)	0.710 (0.706-0.713)	0.737 (0.732-0.742)
HACS	RF	**0.702** (0.699-0.705)	**0.762** (0.759-0.766)	0.793 (0.789-0.798)
BR	XGB	0.676 (0.672-0.679)	0.752 (0.748-0.755)	0.792 (0.787-0.796)
CC_FD	XGB	0.634 (0.631-0.638)	0.727 (0.723-0.732)	0.769 (0.764-0.774)
CC_BD	XGB	0.524 (0.523-0.526)	0.535 (0.533-0.537)	0.544 (0.541-0.546)
NS_FD	XGB	0.655 (0.651-0.658)	0.722 (0.718-0.726)	0.758 (0.753-0.762)
NS_BD	XGB	0.666 (0.662-0.670)	0.704 (0.700-0.709)	0.747 (0.742-0.752)
HACS	XGB	0.691 (0.688-0.695)	0.761 (0.758-0.764)	**0.802** (0.798-0.807)
MCM	-	0.664 (0.661-0.667)	0.749 (0.745-0.752)	0.789 (0.785-0.793)
COX	-	0.658 (0.655-0.661)	0.746 (0.742-0.749)	0.781 (0.776-0.785)
MTLSA	-	0.595 (0.591-0.599)	0.588 (0.583-0.592)	0.583 (0.578-0.588)
RSF	-	0.684 (0.681-0.687)	0.751 (0.747-0.754)	0.784 (0.779-0.789)

Table 5. Performance of Time-to-Default Prediction in Terms of IBS

Method	Base classifier	Pre-loan stage	Post-loan stage (t_1)	Post-loan stage (t_2)
BR	LR	0.441 (0.436-0.446)	0.260 (0.257-0.264)	0.130 (0.129-0.132)
CC_FD	LR	0.513 (0.507-0.519)	0.465 (0.460-0.470)	0.288 (0.285-0.292)
CC_BD	LR	0.472 (0.466-0.477)	0.285 (0.281-0.289)	0.132 (0.130-0.134)
NS_FD	LR	0.441 (0.436-0.446)	0.260 (0.257-0.263)	0.130 (0.128-0.132)
NS_BD	LR	0.441 (0.436-0.446)	0.260 (0.257-0.263)	0.130 (0.128-0.132)
HACS	LR	0.440 (0.435-0.445)	0.260 (0.257-0.263)	0.130 (0.128-0.132)
BR	RF	0.408 (0.403-0.413)	0.255 (0.251-0.258)	0.127 (0.126-0.129)
CC_FD	RF	0.451 (0.445-0.457)	0.274 (0.270-0.277)	0.127 (0.125-0.129)
CC_BD	RF	0.433 (0.427-0.438)	0.282 (0.278-0.285)	0.136 (0.134-0.138)
NS_FD	RF	0.447 (0.442-0.452)	0.304 (0.301-0.308)	0.158 (0.156-0.161)
NS_BD	RF	0.425 (0.420-0.430)	0.292 (0.289-0.296)	0.146 (0.144-0.148)
HACS	RF	**0.405** (0.400-0.410)	0.251 (0.248-0.254)	0.125 (0.123-0.126)
BR	XGB	0.414 (0.409-0.419)	0.254 (0.251-0.257)	0.125 (0.123-0.127)
CC_FD	XGB	0.462 (0.457-0.468)	0.281 (0.277-0.285)	0.130 (0.128-0.132)
CC_BD	XGB	0.695 (0.673-0.717)	0.461 (0.444-0.479)	0.211 (0.199-0.223)
NS_FD	XGB	0.541 (0.532-0.550)	0.347 (0.341-0.353)	0.192 (0.188-0.196)
NS_BD	XGB	0.488 (0.479-0.497)	0.307 (0.301-0.313)	0.156 (0.152-0.159)
HACS	XGB	0.410 (0.405-0.415)	**0.250** (0.247-0.253)	**0.123** (0.121-0.125)
MCM	-	0.436 (0.431-0.441)	0.258 (0.255-0.261)	0.129 (0.127-0.131)
COX	-	0.441 (0.435-0.446)	0.261 (0.258-0.264)	0.133 (0.131-0.135)
MTLSA	-	0.541 (0.432-0.651)	0.546 (0.334-0.758)	0.290 (0.163-0.418)
RSF	-	0.411 (0.406-0.416)	0.254 (0.251-0.257)	0.127 (0.125-0.129)

Table 6. Results of Full Pairwise Comparisons at the Pre-Loan Stage

Method	Average rank	*p*-value of pairwise comparison adjusted by Bonferroni correction				
		BR_RF	BR_XGB	HACS_RF	HACS_XGB	MCM
BR_RF	2.84					
BR_XGB	4.63	<0.001				
HACS_RF	**1.20**	<0.001	<0.001			
HACS_XGB	2.84	1.000	<0.001	<0.001		
MCM	5.86	<0.001	<0.001	<0.001	<0.001	
RSF	3.62	<0.001	<0.001	<0.001	0.001	<0.001
Friedman χ^2	744.391 (*p*<0.001)					

Table 7. Results of Full Pairwise Comparisons at the Post-Loan Stage

Method	Average rank	*p*-value of pairwise comparison adjusted by Bonferroni correction				
		BR_RF	BR_XGB	HACS_RF	HACS_XGB	MCM
BR_RF	4.68					
BR_XGB	3.42	<0.001				
HACS_RF	2.24	<0.001	<0.001			
HACS_XGB	**1.67**	<0.001	<0.001	<0.001		
MCM	4.83	1.000	<0.001	<0.001	<0.001	
RSF	4.16	0.001	<0.001	<0.001	<0.001	<0.001
Friedman χ^2	976.803 (*p*<0.001)					

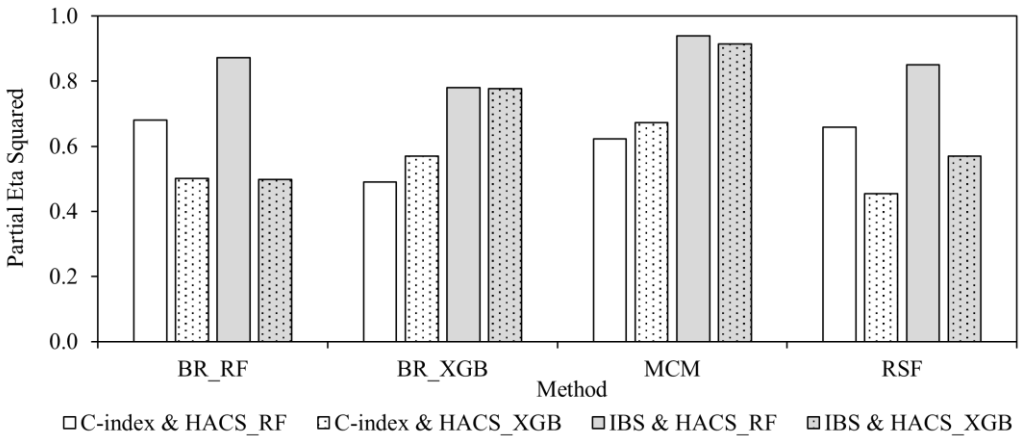

Figure 8. Partial η^2 of Repeated-Measure ANONA

Table 8. Performance (Average AUC) in Predicting Default in Each Month

Method	Pre-loan stage	Post-loan stage (t_1)	Post-loan stage (t_2)
BR_RF	0.609 (0.605-0.614)	0.640 (0.634-0.645)	0.659 (0.652-0.667)
BR_XGB	0.596 (0.591-0.600)	0.623 (0.617-0.628)	0.635 (0.628-0.642)
HACS_RF	**0.676** (0.672-0.680)	**0.734** (0.730-0.739)	**0.770** (0.765-0.776)
HACS_XGB	0.664 (0.660-0.668)	0.728 (0.723-0.733)	0.768 (0.762-0.774)
MCM	0.662 (0.658-0.666)	0.721 (0.716-0.726)	0.746 (0.740-0.753)
RSF	0.646 (0.642-0.650)	0.702 (0.696-0.707)	0.733 (0.727-0.740)

Discrimination Performance over Time Horizons

We then examined: *Whether and how much HACS contributes to discrimination performance improvement over multi-label learning and survival analysis methods for specific time horizons.* The results of discrimination performance of the HACS versus benchmarked method for different time horizons at the pre-loan and post-loan stages are available in Appendix B, and the results of a robustness check regarding base classifier and parameter tuning are available in Appendix C. Overall, HACS with a nonlinear base classifier (either RF or XGB) always gave the best discrimination performance in terms of all performance metrics at both stages and for all prediction horizons, showing that HACS may be a superior alternative to multi-label learning and survival analysis for MPDP.

Effect of Sample Selection Bias

At the pre-loan stage, each MPDP method was trained using approved applicants only but needed to evaluate all applicants, resulting in sample selection bias. To ensure the robustness of our findings, we examined the effect of sample selection bias on the discrimination performance of MPDP methods using the same strategy used by Verstraeten and Van (2005). Specifically, we treated our dataset as a "full" sample and marginally accepted applicants (i.e., applicants with a "platform-assigned grade" of E or F) as rejected applicants. Note that the feature "platform-assigned grade" was hereupon excluded from each method. We then trained each method using "approved" observations (i.e., removing observations with grades of E or F from the original training set) and tested it using two testing sets: (1) the original testing set, including observations with grades of A to F (i.e., with sample selection bias); (2) a reduced testing set excluding observations with grades of E or F (i.e., without sample selection bias). Table 9 summarizes the discrimination performance of each method on these two testing sets. The results show that sample selection bias had little effect on discrimination performance, echoing previous studies, such as Verstraeten and Van (2005).

Ablation Study

To examine whether and how much each of our design artifacts contributes to performance improvement, we carried out an ablation study. Specifically, we built three methods: (1) HACS with RF (M_0); (2) M_0 without the artifact of joint default modeling (M_1) (i.e., RF for default discrimination model and softmax regression directly for the default time estimation model); and (3) M_1 without the artifact of hybrid modeling (M_2) (i.e., softmax regression directly for all statuses, including in default and not in default at each time). Table 10 summarizes the discrimination performance of the three designed methods (M_0, M_1, and M_2) at different stages and prediction horizons (t_1 to t_4 at the pre-loan stage and $t_2|t_1$ to $t_4|t_2$ at the post-loan stage). Overall, the discrimination performance showed a weakening trend with the removal of the design artifacts ($M_0 > M_1 > M_2$). Between the two design artifacts, the damage to performance induced by dropping hybrid modeling was more severe than that induced by dropping joint default modeling. The results show that both design artifacts contribute to an improvement in discrimination performance, albeit to different degrees, and jointly constitute the utility of HACS.

Case Analysis

We also examined: *Whether the methods (multi-label learning, survival analysis, and HACS) provide monotonic predictions over multiple periods as well as the identifiability and discriminability of the methods.* Due to the inclusion relationship in terms of time, the default probability during a longer period will certainly be higher than that of a shorter period. Such semantically meaningful relationships may not always be captured by prediction methods. As discussed earlier, HACS and survival analysis methods embed the monotonicity constraint into their modeling processes. However, when using a multi-label learning method, it may behave unexpectedly, i.e., producing non-monotonic predictions.

We drilled down to the individual case level to examine how the monotonicity of predictions affects the effectiveness of credit risk evaluations (e.g., in identifying the default time). We selected two cases, each corresponding to a non-monotonic prediction of BR_RF or BR_XGB, respectively, as illustrative examples from 10 independent 10-fold cross-validations. We also examined the predictions of HACS (with the same base classifier) and MCM (the best survival analysis method in time-interval-level default prediction) for these cases. Both cases defaulted between the sixth month and the ninth month (specifically, case #1 defaulted at the eighth month and case #2 defaulted at the seventh month). We caution that the main purpose of this case analysis is to examine whether the MPDP methods are capable of accurately identifying the default event, but not to simulate the real decision process, which usually involves more nuances beyond the individual level of default risk. Figure 9 illustrates the multiperiod default probability predictions of each method for each case.

Table 9. Results on the Effect of Sample Selection Bias

Method	Time	With sample selection bias			Without sample selection bias		
		AUC	KS	H-measure	AUC	KS	H-measure
BR_RF	t_1	0.689 (0.682-0.695)	0.317 (0.306-0.328)	0.158 (0.151-0.166)	0.685 (0.677-0.692)	0.318 (0.306-0.331)	0.183 (0.174-0.192)
BR_XGB		0.683 (0.677-0.689)	0.312 (0.302-0.322)	0.155 (0.148-0.163)	0.681 (0.673-0.689)	0.316 (0.303-0.328)	0.177 (0.168-0.186)
HACS_RF		0.713 (0.707-0.719)	0.358 (0.348-0.367)	0.184 (0.176-0.191)	0.718 (0.711-0.726)	0.367 (0.355-0.379)	0.225 (0.215-0.235)
HACS_XGB		0.692 (0.687-0.698)	0.328 (0.318-0.338)	0.169 (0.162-0.176)	0.705 (0.697-0.712)	0.350 (0.338-0.362)	0.207 (0.197-0.216)
MCM		0.649 (0.642-0.656)	0.258 (0.248-0.269)	0.135 (0.127-0.142)	0.657 (0.648-0.666)	0.275 (0.262-0.287)	0.156 (0.146-0.165)
RSF		0.685 (0.679-0.691)	0.318 (0.308-0.327)	0.155 (0.148-0.162)	0.687 (0.680-0.695)	0.324 (0.312-0.336)	0.187 (0.178-0.195)
BR_RF	t_2	0.682 (0.678-0.686)	0.294 (0.286-0.301)	0.138 (0.133-0.143)	0.675 (0.669-0.680)	0.275 (0.267-0.284)	0.164 (0.158-0.171)
BR_XGB		0.662 (0.657-0.667)	0.260 (0.252-0.268)	0.120 (0.115-0.125)	0.661 (0.655-0.667)	0.261 (0.252-0.270)	0.139 (0.133-0.145)
HACS_RF		0.687 (0.683-0.692)	0.301 (0.294-0.308)	0.143 (0.138-0.149)	0.687 (0.682-0.693)	0.298 (0.289-0.307)	0.173 (0.167-0.179)
HACS_XGB		0.671 (0.667-0.676)	0.273 (0.266-0.280)	0.130 (0.125-0.135)	0.679 (0.673-0.684)	0.283 (0.274-0.292)	0.151 (0.145-0.158)
MCM		0.634 (0.629-0.639)	0.217 (0.210-0.225)	0.099 (0.095-0.103)	0.644 (0.638-0.649)	0.236 (0.227-0.244)	0.116 (0.110-0.121)
RSF		0.679 (0.675-0.684)	0.287 (0.280-0.294)	0.139 (0.133-0.144)	0.673 (0.668-0.679)	0.276 (0.268-0.285)	0.162 (0.156-0.169)
BR_RF	t_3	0.677 (0.673-0.681)	0.285 (0.279-0.291)	0.127 (0.123-0.132)	0.678 (0.673-0.683)	0.282 (0.275-0.289)	0.156 (0.151-0.162)
BR_XGB		0.659 (0.655-0.662)	0.245 (0.239-0.251)	0.113 (0.109-0.117)	0.663 (0.659-0.668)	0.250 (0.243-0.257)	0.130 (0.125-0.134)
HACS_RF		0.679 (0.675-0.683)	0.284 (0.277-0.290)	0.129 (0.125-0.133)	0.683 (0.678-0.688)	0.286 (0.279-0.294)	0.160 (0.155-0.165)
HACS_XGB		0.665 (0.661-0.668)	0.252 (0.246-0.258)	0.118 (0.114-0.122)	0.672 (0.668-0.677)	0.262 (0.255-0.269)	0.136 (0.131-0.141)
MCM		0.634 (0.631-0.638)	0.207 (0.201-0.213)	0.093 (0.089-0.096)	0.646 (0.642-0.650)	0.229 (0.222-0.236)	0.108 (0.104-0.112)
RSF		0.674 (0.670-0.677)	0.274 (0.268-0.280)	0.130 (0.126-0.135)	0.674 (0.670-9.679)	0.273 (0.266-0.280)	0.154 (0.149-0.159)
BR_RF	t_4	0.673 (0.670-0.677)	0.270 (0.264-0.275)	0.122 (0.118-0.126)	0.678 (0.674-0.682)	0.274 (0.268-0.281)	0.149 (0.145-0.154)
BR_XGB		0.661 (0.658-0.665)	0.242 (0.236-0.247)	0.112 (0.108-0.115)	0.670 (0.666-0.674)	0.253 (0.246-0.260)	0.128 (0.124-0.133)
HACS_RF		0.673 (0.670-0.677)	0.269 (0.263-0.275)	0.121 (0.118-0.125)	0.678 (0.674-0.682)	0.273 (0.266-0.280)	0.149 (0.145-0.153)
HACS_XGB		0.662 (0.659-0.665)	0.244 (0.238-0.249)	0.113 (0.109-0.116)	0.671 (0.667-0.675)	0.255 (0.248-0.262)	0.130 (0.126-0.134)
MCM		0.637 (0.633-0.640)	0.204 (0.199-0.209)	0.089 (0.086-0.092)	0.650 (0.646-0.653)	0.229 (0.223-0.236)	0.104 (0.100-0.108)
RSF		0.672 (0.668-0.675)	0.262 (0.256-0.267)	0.128 (0.124-0.131)	0.673 (0.669-0.677)	0.264 (0.258-0.270)	0.147 (0.143-0.152)

Table 10. Results of Ablation Study

Method	Time	AUC	KS	H-measure
M_0	t_1	0.753 (0.747-0.759)	0.404 (0.394-0.414)	0.278 (0.270-0.287)
M_1		0.729 (0.723-0.735)	0.366 (0.355-0.377)	0.239 (0.230-0.248)
M_2		0.687 (0.681-0.694)	0.319 (0.309-0.330)	0.174 (0.166-0.182)
M_0	t_2	0.722 (0.717-0.726)	0.339 (0.332-0.346)	0.230 (0.224-0.236)
M_1		0.703 (0.699-0.708)	0.313 (0.306-0.320)	0.193 (0.188-0.199)
M_2		0.668 (0.663-0.672)	0.275 (0.267-0.283)	0.135 (0.130-0.140)
M_0	t_3	0.711 (0.707-0.715)	0.318 (0.312-0.324)	0.208 (0.203-0.213)
M_1		0.697 (0.694-0.701)	0.297 (0.291-0.303)	0.180 (0.175-0.184)
M_2		0.661 (0.657-0.664)	0.256 (0.250-0.262)	0.118 (0.114-0.121)
M_0	t_4	0.707 (0.704-0.711)	0.307 (0.302-0.313)	0.200 (0.196-0.205)
M_1		0.698 (0.694-0.701)	0.295 (0.289-0.300)	0.176 (0.172-0.180)
M_2		0.663 (0.660-0.666)	0.256 (0.250-0.261)	0.112 (0.109-0.115)
M_0	$t_2 \mid t_1$	0.810 (0.804-0.815)	0.523 (0.514-0.532)	0.389 (0.381-0.397)
M_1		0.789 (0.783-0.795)	0.491 (0.482-0.501)	0.370 (0.361-0.379)
M_2		0.773 (0.768-0.779)	0.451 (0.442-0.460)	0.323 (0.314-0.331)
M_0	$t_3 \mid t_1$	0.781 (0.777-0.786)	0.447 (0.440-0.455)	0.319 (0.312-0.326)
M_1		0.766 (0.761-0.771)	0.435 (0.427-0.443)	0.310 (0.303-0.317)
M_2		0.753 (0.749-0.756)	0.406 (0.399-0.412)	0.263 (0.257-0.270)
M_0	$t_4 \mid t_1$	0.767 (0.764-0.771)	0.411 (0.404-0.417)	0.285 (0.280-0.290)
M_1		0.753 (0.749-0.757)	0.400 (0.393-0.407)	0.275 (0.269-0.281)
M_2		0.739 (0.735-0.742)	0.371 (0.365-0.377)	0.231 (0.225-0.236)
M_0	$t_3 \mid t_2$	0.823 (0.817-0.829)	0.542 (0.532-0.553)	0.403 (0.393-0.414)
M_1		0.815 (0.809-0.820)	0.531 (0.522-0.541)	0.400 (0.390-0.410)
M_2		0.796 (0.790-0.802)	0.488 (0.477-0.499)	0.350 (0.338-0.361)
M_0	$t_4 \mid t_2$	0.796 (0.792-0.801)	0.476 (0.468-0.484)	0.339 (0.330-0.347)
M_1		0.792 (0.787-0.797)	0.474 (0.466-0.483)	0.335 (0.327-0.343)
M_2		0.777 (0.773-0.781)	0.441 (0.433-0.448)	0.293 (0.285-0.301)

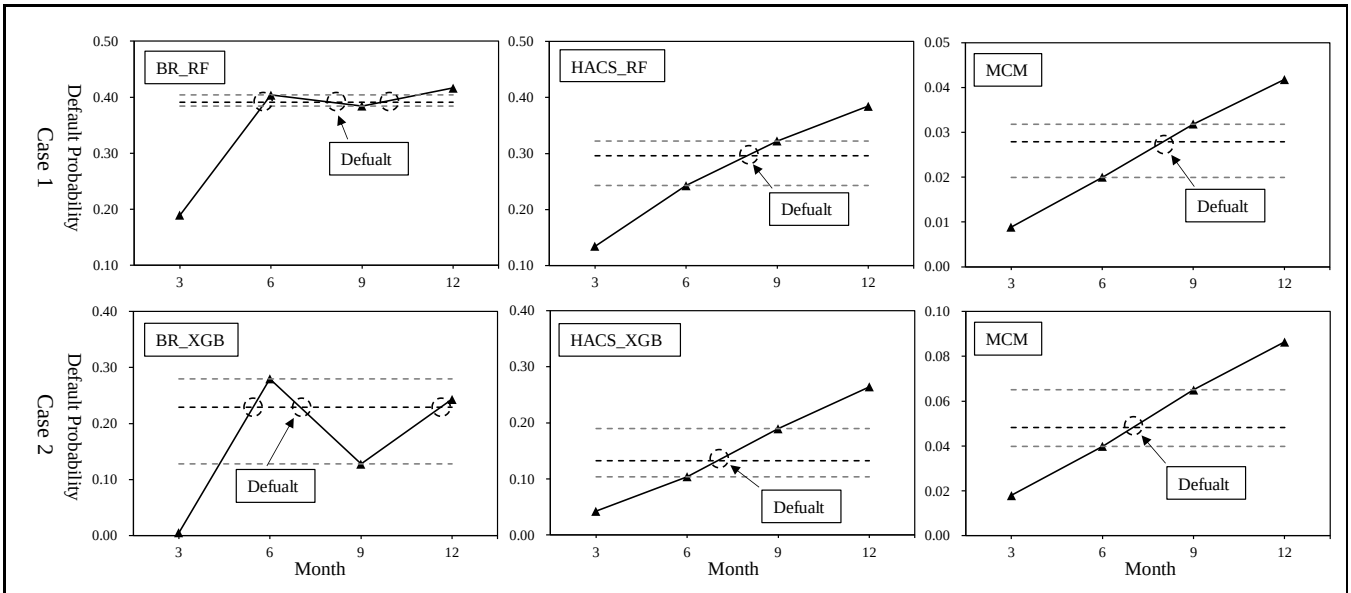

Figure 9. Illustration of Multiperiod Default Probability Predictions

The case analysis results show the *identifiability*, i.e., the ability to precisely infer the occurrence time of default, and *discriminability*, i.e., difference in default probability predictions before and after the occurrence of default of each method. For identifiability, with a proper threshold (cut-off line), HACS and MCM were able to precisely identify when the borrower defaulted in each case, whereas the curves related to BR had multiple intersections with the cut-off line, providing confusing results. These spurious results of BR failed to accurately depict the evolution process of credit risk and could easily cause stakeholders to misjudge the risk. In case #1, BR_RF triggered three risk signals; the first signal occurred prior to the sixth month but the borrower actually defaulted in the eighth month. In case #2, BR_XGB also triggered three risk signals; the first signal occurred prior to the sixth month but the borrower actually defaulted in the seventh month. The results provide clear evidence that, especially in practical use, monotonicity is essential for MPDP to accurately identify the time to default.

For discriminability, the grey lines accompanying each curve highlighted the default probability predictions before and after the occurrence of a loan default, respectively. The wider the band between the two grey lines, the greater the difference between the predictions before and after default, indicating that the method is better able to identify default. Although HACS and MCM both possess identifiability, they revealed differences in discriminability. Specifically, the width between the grey lines of HACS was higher than that of MCM (0.079 vs. 0.012 and 0.086 vs. 0.025 for the two cases, respectively), indicating that HACS predicted the changes of credit risk more accurately and effectively than MCM. Overall, the case analysis shows that the survival analysis (MCM) and HACS were superior to multi-label learning (BR) in terms of identifiability and that HACS was further superior to survival analysis (MCM) in terms of discriminability.

Impact Analysis through Simulation Study

We also designed impact analyses to examine the *granting* performance and *profitability* performance of each method from a practical perspective. For financial institutions such as banks, loan granting is arguably their most important credit decision, and methods supporting this decision are expected to realize as low a default rate as possible under diverse granting proportions (i.e., granting performance). As banks may also consider the loan term when granting loans, varying from short-term (e.g., three months) to long-term (e.g., one year or longer), we examined the granting performance of each method at four representative times—3, 6, 9, and 12 months. For individual investors, their goals generally focus on achieving profits while avoiding defaults. We simulated real-case investment scenarios and selected multiple loan applications using profit-ranking recommendations of the MPDP and classification-based default prediction methods or the simple proxy index for profit, i.e., interest rate.

Practical Use in Terms of Granting Performance

We examined the granting performance from a practical perspective: *Whether and how much HACS helps financial institutions such as banks grant loans more effectively than other methods.* We simulated real-case loan granting scenarios and selected multiple loan applications in our dataset using the risk-ranking results of credit scoring methods. We then counted the number of loans in default under different granting ratios (i.e., granting performance). For example, assuming that we decided to lend money to 20% of the loan applications in our dataset, we chose the top 20% of loan applications based on MPDP results. We adopted a large range of granting ratios to more realistically reflect various possible real-case scenarios, such as tight and lax credit policies.

Table 11 summarizes the granting performance (i.e., number of loans in default) of each method at different horizons (t_1 to t_4) and granting proportions (from 20% to 50%). Compared to using BR or MCM, granting loans using HACS, with either RF or XGB, always resulted in fewer loans in default under any granting proportion and at any horizon (except t_4, at which BR and HACS are equivalent). Compared to using RSF, granting loans using HACS with RF always resulted in fewer loans in default. The decrease in loan defaults could reduce losses for financial institutions when granting loans. The results show that HACS improved the risk-ranking ability and decreased the number of loan defaults, compared to other methods.

Practical Use in Terms of Profitability Performance

We also examined the profitability performance from a practical perspective: *Whether and how much HACS helps investors select loan portfolios with higher profits than other methods.* In the actual financial market, given the possible risk of losing some or even all the principal due to loan defaults, investors often mitigate their total risk by diversifying their investments across multiple loans and investing in a portfolio of multiple loans rather than a single loan.

Table 11. Granting Performance

Time	%	BR_RF	BR_XGB	HACS_RF	HACS_XGB	MCM	RSF
t_1	20	3.740 (3.375-4.105)	4.080 (3.716-4.444)	3.520 (3.178-3.862)	**3.270** (2.915-3.625)	3.570 (3.215-3.925)	4.190 (3.807-4.573)
	25	4.800 (4.377-5.223)	5.270 (4.812-5.728)	4.570 (4.189-4.951)	**4.450** (4.021-4.879)	5.290 (4.848-5.732)	5.370 (4.962-5.778)
	30	6.360 (5.891-6.829)	6.670 (6.156-7.184)	**5.540** (5.105-5.975)	5.810 (5.274-6.346)	7.410 (6.882-7.938)	6.610 (6.171-7.049)
	35	8.040 (7.504-8.576)	8.160 (7.636-8.684)	**6.750** (6.283-7.217)	7.290 (6.754-7.826)	9.220 (8.657-9.783)	7.850 (7.372-8.328)
	40	9.800 (9.222-10.378)	9.970 (9.407-10.533)	**8.060** (7.587-8.533)	8.890 (8.281-9.499)	11.420 (10.820-12.020)	9.520 (8.997-10.043)
	45	11.670 (11.029-12.311)	11.890 (11.303-12.477)	**9.460** (8.926-9.994)	10.620 (10.000-11.240)	13.430 (12.754-14.106)	11.360 (10.773-11.947)
	50	13.900 (13.140-14.660)	14.030 (13.361-14.699)	**10.970** (10.398-11.542)	12.580 (11.936-13.224)	15.740 (14.980-16.500)	13.150 (12.465-13.835)
t_2	20	9.370 (8.738-10.002)	9.830 (9.235-10.425)	8.970 (8.365-9.575)	**8.290** (7.711-8.869)	10.290 (9.679-10.901)	9.740 (9.215-10.265)
	25	12.210 (11.588-12.832)	13.130 (12.427-13.833)	11.820 (11.123-12.517)	**11.270** (10.553-11.987)	14.280 (13.590-14.970)	12.850 (12.155-13.545)
	30	16.030 (15.301-16.759)	16.780 (16.017-17.543)	14.880 (14.070-15.690)	**14.440** (13.633-15.247)	18.380 (17.560-19.200)	15.960 (15.209-16.711)
	35	19.740 (18.971-20.509)	20.610 (19.774-21.446)	**17.870** (17.021-18.719)	18.150 (17.187-19.113)	22.550 (21.627-23.473)	19.680 (18.837-20.523)
	40	23.860 (23.046-24.674)	24.950 (24.052-25.848)	**21.500** (20.631-22.369)	22.290 (21.301-23.279)	26.710 (25.732-27.688)	23.560 (22.670-24.450)
	45	27.790 (26.868-28.712)	29.520 (28.503-30.537)	**25.580** (24.607-26.553)	26.740 (25.744-27.736)	30.920 (29.871-31.969)	27.200 (26.246-28.154)
	50	32.080 (31.081-33.079)	34.420 (33.307-35.533)	**29.870** (28.917-30.823)	31.440 (30.396-32.484)	35.980 (34.829-37.131)	31.430 (30.374-32.486)
t_3	20	15.290 (14.498-16.082)	14.160 (13.443-14.877)	14.590 (13.780-15.400)	**13.340** (12.626-14.054)	15.240 (14.535-15.945)	15.580 (14.757-16.403)
	25	20.570 (19.649-21.491)	19.400 (18.603-20.197)	19.770 (18.774-20.766)	**18.260** (17.407-19.113)	21.600 (20.745-22.455)	20.600 (19.630-21.570)
	30	25.860 (24.887-26.833)	25.590 (24.576-26.604)	24.480 (23.347-25.613)	**23.880** (22.900-24.860)	29.010 (28.003-30.017)	26.080 (25.021-27.139)
	35	31.970 (30.861-33.079)	31.850 (30.717-32.983)	**29.820** (28.624-31.016)	30.100 (28.956-31.244)	35.520 (34.420-36.620)	31.820 (30.636-33.004)
	40	38.200 (37.000-39.400)	38.870 (37.634-40.106)	**36.060** (34.787-37.333)	36.910 (35.723-38.097)	42.520 (41.354-43.686)	38.140 (36.831-39.449)
	45	43.900 (42.641-45.159)	46.290 (44.970-47.610)	**42.250** (40.964-43.536)	44.380 (43.103-45.657)	49.560 (48.310-50.810)	44.140 (42.730-45.550)
	50	49.830 (48.573-51.087)	53.750 (52.458-55.042)	**48.340** (46.965-49.715)	52.570 (51.241-53.899)	56.420 (55.091-57.749)	50.780 (49.366-52.194)
t_4	20	-	-	18.440 (17.537-19.343)	**16.930** (16.111-17.749)	19.090 (18.227-19.953)	19.140 (18.214-20.066)
	25	-	-	24.990 (23.902-26.078)	**23.180** (22.139-24.221)	27.360 (26.299-28.421)	25.440 (24.378-26.502)
	30	-	-	30.820 (29.588-32.052)	**30.710** (29.508-31.912)	36.600 (35.457-37.743)	31.800 (30.561-33.039)
	35	-	-	**37.650** (36.345-38.955)	38.680 (37.363-39.997)	45.180 (43.891-46.469)	39.200 (37.799-40.601)
	40	-	-	**45.600** (44.244-46.956)	47.530 (46.123-48.937)	52.990 (51.666-54.314)	47.240 (45.784-48.696)
	45	-	-	**53.900** (52.417-55.383)	57.900 (56.394-59.406)	62.150 (60.749-63.551)	55.700 (54.235-57.165)
	50	-	-	**63.050** (61.551-64.549)	68.640 (67.005-70.275)	71.210 (69.675-72.745)	64.530 (63.005-66.055)

Note: BR and HACS are equivalent at the full prediction horizon (t_4).

When selecting portfolios, two types of strategies are generally used: (1) selecting loan applications with high interest rates, assuming that risks have already been considered in pricing; (2) selecting loan applications with low default probabilities predicted by credit scoring methods (widely adopted). In practice, these strategies may both suffer from deficiencies in that they may be either too speculative (the first strategy) or incompatible with profit maximization (the second strategy). Hence, we propose a new portfolio selection strategy based on HACS. Given the multiperiod default probabilities predicted by HACS, the proposed strategy ranks loan applications by estimating the *expected return rate* (ERR):

$$ERR = (1 - p_m) * IR - p_m * \left(\sum_{i=1}^{m} \frac{m-i+1}{m} \left(\frac{p_m - p_{m-1}}{1 - p_{m-1}} \right) \right), \quad (7)$$

where *IR* denotes the *interest rate* and p_i is predicted by HACS (with p_0 identically equal to zero). ERR measures both *return*, i.e., interest conditional on not defaulting, and *loss*, i.e., default probability in each time interval and its corresponding loss rate, and thus is expected to give a more accurate profit ranking.

We simulated real-case investment scenarios and selected multiple loan applications in our dataset using the profit-ranking results of the three strategies: (1) IR, (2) probability of default (PD), and (3) ERR. Profitability performance was measured using the average return rate of the selected portfolio. To remove the influence of the loan amount, we assumed that the total capital would be evenly allocated to each loan in the portfolio. For the IR ranking, we used the interest rate of each loan application. For the PD ranking, we used classification methods (RF and XGB) to estimate the default probability of each loan during the full loan term. For the ERR ranking, we used HACS (with RF and XGB), MCM, and RSF to estimate the ERR of each loan application. Since monotonicity cannot be strictly satisfied, multi-label learning methods cannot accurately estimate ERR and thus were left out of this analysis.

Table 12 summarizes the profitability performance (i.e., average return rate) of each strategy at different portfolio sizes. First, with any portfolio size, the *IR* strategy always led to the worst average return rates, providing clear evidence that high returns are also accompanied by high risks and simply pursuing high interest rates may even lead to losses. Second, the PD strategy always led to modest average return rates, and its profitability performance was worse than that of the ERR strategy using HACS, indicating the superiority of the proposed ERR strategy combined with HACS. Third, with the ERR strategy, using HACS with RF always led to the best profitability performance (i.e., highest average return rate), with any portfolio size, indicating that HACS may depict the default risk more effectively, compared to MCM and RSF, thus helping to estimate ERR more accurately. Overall, the results show that compared to the widely used IR and PD strategies, the proposed ERR strategy combined with HACS was able to identify loan portfolios with higher profits.

We further drilled down to the individual portfolio level to examine how the proposed ERR strategy combined with HACS changed the selected portfolio. We selected two folds of portfolio selection results as illustrative examples from 10 independent 10-fold cross-validations. Figure 10 illustrates the portfolio selection results of IR, PD_RF (the best PD strategy), and ERR_HACS_RF in these two cases. Overall, the average return rate of the selected portfolio using ERR_HACS_RF was higher than that using IR or PD_RF. As a straightforward strategy, IR simply pursued loans with the highest interest rates while totally ignoring risk, leading to a considerable number of loans in default. The PD strategy using RF effectively selected loans with lower risk (i.e., probability of default), leading to only one loan in default in both cases. However, the return rates of the selected loans were mostly moderate, and when a selected loan defaulted, its loss might become erratic (e.g., -49% in case #1 and -51% in case #2) as time-to-default was not considered. The ERR strategy was designed to find an appropriate tradeoff between return and loss; thus, the ERR strategy using HACS with RF-selected loans with relatively high return rates while absorbing a few loans in default. As HACS with RF effectively injected a suppression effect on the time to default (i.e., the earlier the default, the higher the risk score), the selected loans in default tended to default relatively late with lower losses. Some of the selected loans in default (e.g., the loan in default on the left in case #2 of Figure 10) might even be profitable if they defaulted sufficiently late.

Discussion and Conclusion

In response to the growing demand for MPDP and the deficiencies of existing methods—both survival analysis and multi-label learning—we have initiated a new way for MPDP by proposing the HACS approach. We synthesize statistical modeling and machine learning in the proposed approach to achieve desired and valuable properties, such as monotonic prediction and complex relationship accommodation. Our empirical evaluation shows the advantages of the proposed approach. It outperformed the benchmarked survival analysis and multi-label learning methods in terms of time-to-default prediction performance and discrimination performance. Case analysis further verified its identifiability and discriminability. Financial institutions and individual investors could both benefit from it by improving granting performance when granting loans and profitability performance when selecting loan portfolios.

Table 12. Profitability Performance

Size	IR	PD_RF	PD_XGB	ERR_HACS_RF	ERR_HACS_XGB	ERR_MCM	ERR_RSF
10	0.051 (0.030-0.072)	0.194 (0.191-0.197)	0.183 (0.179-0.186)	**0.217** (0.208-0.225)	0.196 (0.185-0.208)	0.156 (0.142-0.170)	0.209 (0.199-0.218)
20	0.080 (0.064-0.096)	0.194 (0.192-0.196)	0.185 (0.182-0.188)	**0.211** (0.204-0.218)	0.193 (0.185-0.201)	0.178 (0.170-0.186)	0.207 (0.200-0.214)
30	0.065 (0.053-0.078)	0.194 (0.192-0.196)	0.184 (0.182-0.187)	**0.208** (0.202-0.214)	0.194 (0.187-0.201)	0.189 (0.182-0.195)	0.207 (0.201-0.212)
40	0.075 (0.064-0.086)	0.195 (0.193-0.196)	0.185 (0.183-0.187)	**0.207** (0.202-0.212)	0.194 (0.188-0.200)	0.195 (0.190-0.200)	0.206 (0.200-0.211)
50	0.084 (0.074-0.093)	0.194 (0.192-0.196)	0.185 (0.182-0.187)	**0.207** (0.203-0.212)	0.194 (0.189-0.200)	0.197 (0.193-0.201)	0.204 (0.199-0.209)
60	0.091 (0.082-0.099)	0.195 (0.193-0.196)	0.184 (0.182-0.187)	**0.208** (0.203-0.212)	0.196 (0.191-0.200)	0.197 (0.194-0.201)	0.205 (0.201-0.210)
70	0.087 (0.079-0.094)	0.194 (0.193-0.196)	0.184 (0.182-0.186)	**0.206** (0.203-0.210)	0.196 (0.192-0.201)	0.198 (0.194-0.201)	0.205 (0.201-0.209)
80	0.083 (0.076-0.091)	0.194 (0.193-0.196)	0.185 (0.183-0.187)	**0.205** (0.201-0.208)	0.198 (0.194-0.202)	0.198 (0.195-0.202)	0.203 (0.199-0.207)
90	0.082 (0.075-0.089)	0.194 (0.192-0.196)	0.186 (0.184-0.188)	**0.204** (0.200-0.207)	0.198 (0.194-0.201)	0.199 (0.196-0.202)	0.203 (0.199-0.206)
100	0.083 (0.076-0.089)	0.193 (0.192-0.195)	0.186 (0.184-0.188)	**0.203** (0.200-0.207)	0.198 (0.195-0.202)	0.198 (0.195-0.201)	0.202 (0.198-0.205)

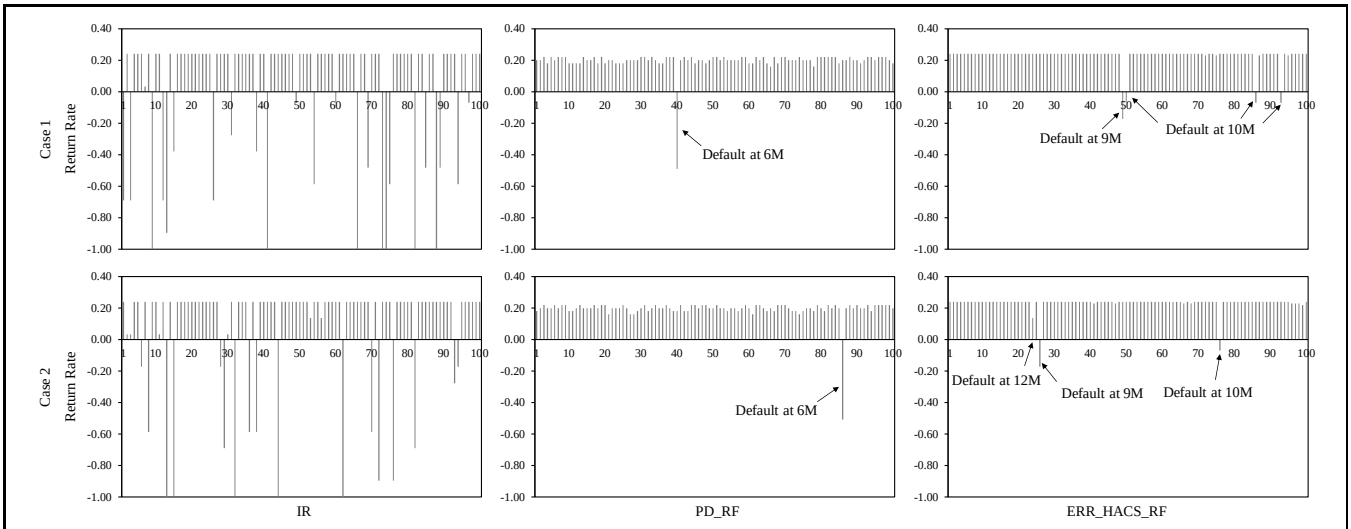

Figure 10. Illustration of Portfolio Selection Results

Contributions to the IS Knowledge Base

We contribute a new MPDP method for credit scoring to the literature. The MPDP research stream follows two avenues: one adopts the statistical modeling paradigm through survival analysis and the other adopts the machine learning paradigm through multi-label learning. We reveal a new avenue of MPDP in the middle of the two existing avenues by synthesizing their advantages. HACS is data driven and free from assumptions on survival time, which hinder the performance of survival analysis methods in financial risk analytics. HACS allows for the prediction of default probability *monotonically* over time, thus alleviating the downsides of multi-label learning methods, which focus on local fitting and ignore the global relevance and monotonicity constraint, rendering them effective for theoretical verification (i.e., discrimination performance) but problematic for practical verification (i.e., they are unable to accurately identify the default time). HACS models default status and default time separately and synthesizes them through a probabilistic framework, allowing for the differentiation of the feature influences regarding *whether* and *when* a borrower will default, thereby helping to better split and fit the complex relationships embedded in credit data.

Design science research offers different types of contributions to the IS knowledge base, including strong theory, partial theory, incomplete theory, and even some particularly interesting and perhaps surprising empirical generalizations in the form of a new design artifact (Gregor & Hevner, 2013). Our theoretical contribution is to motivate, examine, and establish two design principles in credit scoring and data analytics: (1) the hybrid modeling framework and (2) joint default modeling for default time estimations. These design principles prescribe how to model monotonic default probability separately and jointly to attain improved predictive performance. Through empirical evaluation and impact analysis, this study offers proof of value added by demonstrating the viability and utility of these design principles in credit risk analytics. While we designed and evaluated HACS in the context of default prediction, the prescriptive knowledge advanced in this study may be generalizable as a "nascent design theory" (Gregor & Hevner, 2013) to other predictive analytics contexts that satisfy the applicability conditions of HACS: (1) the primary objective is to predict *whether* and *when* a particular event of interest occurs; (2) the event of interest occurs at discrete (truly discrete or recorded on discrete units) values of time within a certain period; (3) some observations may never experience the event of interest in the entire period, i.e., existence of a subgroup of long-term survivors; (4) the features available for prediction tend to be multidimensional or multiview and have complex relationships. For example, in customer relationship management, successful customer retention requires insights into *whether* and *when* a customer will churn within a specific time period (e.g., one year), which can help the organization target retention efforts on the right customer at the right time (Backiel et al., 2016). In this case, HACS may also serve as an effective tool for predicting the time to churn. In crowdfunding, due to the "all-or-nothing" policy (i.e., creators obtain funding only if the fundraising reaches its goal within the fundraising period), both creators and backers are eager to know *when* a project will reach its fundraising target and HACS may accordingly be applicable to time-to-success predictions as well (Wang et al., 2019).

Practical Implications

With the continuous development of fintech and credit markets, credit accessibility is increasing and consumer groups of credit products are becoming larger and more complex. Consequently, credit evaluation tools (e.g., credit scoring) are desired to develop more intelligence to capture the uncertainty and depict the dynamics of credit risk. Financial institutions and individual investors can all benefit from a more comprehensive and accurate credit evaluation tool like HACS. We discuss key practical implications for these stakeholders below.

Financial institutions: Granting loans is the fundamental profitable business of financial institutions such as banks, and credit scoring tools like HACS may facilitate financial institutions in granting loans more effectively. First, depicting the risk profile over time enables financial institutions to implement fine-grained risk stratification (e.g., identifying long-term and short-term risky loans), beyond simply *whether* a borrower will default; consequently, they may benefit from more elaborate information to optimize their credit allocation. Specifically, financial institutions can grant loans in a novel way by engaging in a homogeneous level of risk over time (i.e., risk equalization), thereby achieving sustainable and stable future cash flow and funding turnover. They can also set a risk time, i.e., the intersection of the predicted default probability curve and the preset risk-threshold line (as in the case analysis), and accordingly offer optimal loan terms for new customers. Second, as the industry advances, financial institutions are gradually shifting their granting targets from loans with lower default risk to those with higher profits, i.e., profit scoring, which depends on not only *whether* but also *when* a borrower will default. HACS, as a competitive MPDP method, may help financial institutions acquire and maintain potentially profitable customers to a greater extent to meet such emerging industry demand. Third, compliance guidelines, such as Basel II and Basel III, require financial institutions to base their loss provisions upon the expected losses over the entire lifetime of each credit portfolio. Dynamic evaluation tools like HACS may provide a more accurate decision basis for estimating expected losses, as their components, such as exposure at default, may be highly correlated with default time (as shown in the impact analysis). For post-loan management, knowing *whether* and *when* a customer will default can help financial institutions carry out post-loan risk control in a timely matter and retain the customer by adjusting the line of credit (Agarwal et al., 2021). Further, HACS may also help financial institutions adjust loan asset structure and transfer post-loan risk—for example, by bundling risky short-term loans and selling them to secondary markets (e.g., collectors).

Individual investors: Compared to financial institutions, individual investors generally have fewer funds; therefore, the problem they often face is selecting a loan portfolio rather than approving massive numbers of loan applications. Whether they are risk-averse or risk-seeking, given the same portfolio risk, investors always prefer portfolios that offer higher returns (Fu et al., 2021). This study provides individual investors with a straightforward and effective index, i.e., the *expected return rate*, to select loan portfolios with maximum expected return rates, considering both default losses and default-risk-free returns (i.e., interest rate). In regard to the broadly used index, i.e., *probability of default*, we provide clear evidence that such an index is effective for selecting "safe" loans (i.e., not vulnerable to default) but not necessarily in selecting "good" loans (i.e., yielding high returns). In light of the inferior results of selecting portfolios based simply on interest rates, we also caution investors to rationally consider default risks against

benefits (i.e., interest rate) rather than simply pursuing benefits and ignoring risks. We recommend that investors adopt our proposed method, i.e., the ERR strategy using HACS, to select loan portfolios and that they further diversify their selected portfolios with the proposed optimization model based on modern portfolio theory (see analysis in Appendix D). As new evidence (i.e., repayment information) accumulates in the post-loan stage, investors can then reevaluate the risk of the loans they hold at different time horizons and consider selling loans that exceed their risk appetite (e.g., risky long-term loans).

Limitations and Future Research

Our work has several limitations that could be addressed in future research. First, as discussed earlier, we did not specify the base classifier for each subtask and used homogeneous classifiers. Although we validated the effectiveness of such homogeneous ensembles, the optimal classifier may vary across different subtasks. Future research could consider designing artifacts for selecting the optimal classifier for each subtask to further improve the performance of HACS. Second, as our empirical evaluations were based on one dataset from an online lending platform, whether and to what extent the HACS approach could enhance risk prediction performance in other contexts was not explored. Future research could experiment with HACS on more datasets collected from other contexts to validate the generalizability of our findings. Third, since our goal is to predict rather than to explain, the increased predictive performance comes at a price in model interpretability. In a scenario where interpretability is required, HACS alone may not be sufficient, and interpretable alternatives or post hoc explanation methods (e.g., LIME and SHAP) may need to be adopted to complement it. Future research could explore possibilities along these lines. Fourth, while we broached the topic of how the predictions of HACS can benefit the credit risk management decisions (e.g., post-loan interventions) of financial institutions and individual investors in the discussion of practical implications, the actual effects of these post hoc actions on specific loans cannot be faithfully measured using merely historical data and will still need to be rigorously tested in the future, e.g., through field experiments.

Acknowledgments

Cuiqing Jiang is the corresponding author. The authors would like to express their sincere gratitude to the senior editor, associate editor, and three anonymous reviewers for their constructive feedback, which helped to improve the quality of this paper significantly. This work was supported by the National Natural Science Foundation of China [Grants 71731005 and 72101073], the Anhui Provincial Natural Science Foundation [Grant 2108085MG234], and the Fundamental Research Funds for the Central Universities [Grant JZ2021HGTA0130].

References

Abbasi, A., Albrecht, C., Vance, A., & Hansen, J. (2012). Metafraud: a meta-learning framework for detecting financial fraud. *MIS Quarterly*, *36*(4), 1293-1327. https://doi.org/10.2307/41703508

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

Agarwal, S., Bubna, A., & Lipscomb, M. (2021). Timing to the statement: Understanding fluctuations in consumer credit use. *Management Science*, *67*(8), 5124-5144. https://doi.org/10.1287/mnsc.2020.3720

Backiel, A., Baesens, B., & Claeskens, G. (2016). Predicting time-to-churn of prepaid mobile telephone customers using social network analysis. *Journal of the Operational Research Society*, *67*(9), 1135-1145. https://doi.org/10.1057/jors.2016.8

Baesens, B., Van Gestel, T., Stepanova, M., Van den Poel, D., & Vanthienen, J. (2005). Neural network survival analysis for personal loan data. *Journal of the Operational Research Society*, *56*(9), 1089-1098. https://doi.org/10.1057/palgrave.jors.2601990

Ban, G. Y., El Karoui, N., & Lim, A. E. (2018). Machine learning and portfolio optimization. *Management Science*, *64*(3), 1136-1154. https://doi.org/10.1287/mnsc.2016.2644

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

Dirick, L., Claeskens, G., & Baesens, B. (2017). Time to default in credit scoring using survival analysis: A benchmark study. *Journal of the Operational Research Society*, *68*(6), 652-665. https://doi.org/10.1057/s41274-016-0128-9

Djeundje, V. B., & Crook, J. (2019). Identifying hidden patterns in credit risk survival data using generalised additive models. *European Journal of Operational Research*, *277*(1), 366-376. https://doi.org/10.1016/j.ejor.2019.02.006

Dong, W., Liao, S., & Zhang, Z. (2018). Leveraging financial social media data for corporate fraud detection. *Journal of Management Information Systems*, *35*(2), 461-487. https://doi.org/10.1080/07421222.2018.1451954

Fu, R., Huang, Y., & Singh, P. V. (2021). Crowds, lending, machine, and bias. *Information Systems Research*, *32*(1), 72-92. https://doi.org/10.1287/isre.2020.0990

Ge, R., Feng, J., Gu, B., & Zhang, P. (2017). Predicting and deterring default with social media information in peer-to-peer lending. *Journal of Management Information Systems*, *34*(2), 401-424. https://doi.org/10.1080/07421222.2017.1334472

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, *37*(2), 337-355.

Hand, D. J. (2009). Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine Learning*, *77*(1), 103-123. https://doi.org/10.1007/s10994-009-5119-5

Hendershott, T., Zhang, X., Zhao, J. L., & Zheng, Z. (2021). FinTech as a game changer: Overview of research frontiers. *Information Systems Research*, *32*(1), 1-17. https://doi.org/10.1287/isre.2021.0997

Iyer, R., Khwaja, A. I., Luttmer, E. F., & Shue, K. (2016). Screening peers softly: Inferring the quality of small borrowers.

Management Science, *62*(6), 1554-1577. https://doi.org/10.1287/mnsc.2015.2181

Jiang, C., Wang, Z., & Zhao, H. (2019). A prediction-driven mixture cure model and its application in credit scoring. *European Journal of Operational Research*, *277*(1), 20-31. https://doi.org/10.1016/j.ejor.2019.01.072

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

Li, Y., Wang, J., Ye, J., & Reddy, C. K. (2016). A multi-task learning formulation for survival analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1715-1724). https://doi.org/10.1145/2939672.2939857

Lu, X., Lu, T., Wang, C., & Wu, R. (2021). Can social notifications help to mitigate payment delinquency in online peer-to-peer lending? *POM*, *30*(8), 2564-2585. https://doi.org/10.1111/poms.13395

Martens, D., Provost, F., Clark, J., & de Fortuny, E. J. (2016). Mining massive fine-grained behavior data to improve predictive analytics. *MIS Quarterly*, *40*(4), 869-888.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.

Rivolli, A., Read, J., Soares, C., Pfahringer, B., & de Carvalho, A. C. (2020). An empirical analysis of binary transformation strategies and base algorithms for multi-label learning. *Machine Learning*, *109*(8), 1509-1563. https://doi.org/10.1007/s10994-020-05879-3

Siering, M., Koch, J. A., & Deokar, A. V. (2016). Detecting fraudulent behavior on crowdfunding platforms: The role of linguistic and content-based cues in static and dynamic contexts. *Journal of Management Information Systems*, *33*(2), 421-455. https://doi.org/10.1080/07421222.2016.1205930

Shen, F., Zhao, X., & Kou, G. (2020). Three-stage reject inference learning framework for credit scoring using unsupervised transfer learning and three-way decision theory. *Decision Support Systems*, *137*, Article 113366. https://doi.org/10.1016/j.dss.2020.113366

Sinha, R. K., & Chandrashekaran, M. (1992). A split hazard model for analyzing the diffusion of innovations. *Journal of Marketing Research*, *29*(1), 116-127. https://doi.org/10.1177/002224379202900110

Thomas, L. C. (2009). *Consumer credit models: Pricing, profit and portfolios: pricing, profit and portfolios*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199232130.001.1

Tong, E. N., Mues, C., & Thomas, L. C. (2012). Mixture cure models in credit scoring: If and when borrowers default. *European Journal of Operational Research*, *218*(1), 132-139. https://doi.org/10.1016/j.ejor.2011.10.007

Van Vlasselaer, V., Eliassi-Rad, T., Akoglu, L., Snoeck, M., & Baesens, B. (2017). Gotcha! Network-based fraud detection for social security fraud. *Management Science*, *63*(9), 3090-3110. https://doi.org/10.1287/mnsc.2016.2489

Verstraeten, G., & Van den Poel, D. (2005). The impact of sample bias on consumer credit scoring performance and profitability. *Journal of the Operational Research Society*, *56*(8), 981-992. https://doi.org/10.1057/palgrave.jors.2601920

Vock, D. M., Wolfson, J., Bandyopadhyay, S., Adomavicius, G., Johnson, P. E., Vazquez-Benitez, G., & O'Connor, P. J. (2016). Adapting machine learning techniques to censored time-to-event health record data: A general-purpose approach using inverse probability of censoring weighting. *Journal of Biomedical Informatics*, *61*, 119-131. https://doi.org/10.1016/j.jbi.2016.03.009

Wang, G., Chen, G., Zhao, H., Zhang, F., Yang, S., & Lu, T. (2021). Leveraging multisource heterogeneous data for financial risk prediction: A novel hybrid-strategy-based self-adaptive method. *MIS Quarterly*, *45*(4), 1949-1998. https://doi.org/10.25300/MISQ/2021/16118

Wang, P., Li, Y., & Reddy, C. K. (2019). Machine learning for survival analysis: A survey. *ACM Computing Surveys*, *51*(6), 1-36. https://doi.org/10.1145/3214306

Wang, Z., Jiang, C., Zhao, H., & Ding, Y. (2020). Mining semantic soft factors for credit risk evaluation in Peer-to-Peer lending. *J Management Information Systems*, *37*(1), 282-308. https://doi.org/10.1080/07421222.2019.1705513

Wang, Z., Jiang, C., & Zhao, H. (2022). Know where to invest: Platform risk evaluation in online lending. *Information Systems Research*, *33*(3), 765-783. https://doi.org/10.1287/isre.2021.1083

Author Biographies

Zhao Wang is an assistant professor in the School of Management at Hefei University of Technology, where he received his Ph.D. in management science and engineering. His research interests include data mining and credit evaluation theory and methodology. He has published in such journals as *Information Systems Research*, *Journal of Management Information Systems*, *European Journal of Operational Research*, *Decision Support Systems*, and many others.

Cuiqing Jiang is a professor in the School of Management at Hefei University of Technology, where he received his Ph.D. in management science and engineering. His research interests include big data analytics and business intelligence, data mining and knowledge discovery, and financial technology and information systems. He has published in such journals as *Information Systems Research*, *Journal of Management Information Systems*, *Journal of the Association for Information Systems*, *European Journal of Operational Research*, *Information Sciences*, and many others.

Huimin Zhao is a professor of information technology management in the Lubar College of Business at the University of Wisconsin-Milwaukee. He received B.E. and M.E. degrees in automation from Tsinghua University, China and a Ph.D. in management information systems from the University of Arizona, USA. His current research interests include data mining and healthcare informatics. He has published in such journals as *Information Systems Research*, *MIS Quarterly*, *Communications of the ACM*, *ACM Transactions on MIS*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Systems, Man, and Cybernetics*, *Information Systems*, *Journal of Management Information Systems*, and *Journal of the Association for Information Systems*. He currently serves as an associate editor for *Information Systems Research* and *Journal of Business Analytics*, he has also served as a senior editor for *Decision Support Systems* and an associate editor for *MIS Quarterly*.

Appendix A

Censoring-Adapted HACS

In the vanilla HACS, we managed censored observations using a common simple strategy, i.e., discarding those observations. We also proposed a censoring-adapted version (HACS_IPCW) by training the discrimination model with weighted observations using inverse probability of censoring weighting (IPCW) (Vock et al., 2016). Tables A1 and A2 summarize the performance of vanilla HACS versus censoring-adapted HACS in terms of the C-index and IBS, respectively. The results show that HACS slightly outperformed HACS_IPCW at the pre-loan stage, whereas it slightly underperformed HACS_IPCW, in terms of IBS, at the post-loan stage (t_2). At the post-loan stage (t_1), HACS and HACS_IPCW were somewhat comparable.

Table A1. Performance of Vanilla HACS versus Censoring-Adapted HACS in Terms of C-Index

Method	Base classifier	Pre-loan	Post-loan (t_1)	Post-loan (t_2)
HACS	LR	0.657 (0.655-0.660)	0.747 (0.743-0.750)	0.787 (0.783-0.792)
HACS_IPCW	LR	0.656 (0.653-0.659)	0.746 (0.743-0.750)	0.791 (0.787-0.795)
HACS	RF	0.702 (0.699-0.705)	0.762 (0.759-0.766)	0.793 (0.789-0.798)
HACS_IPCW	RF	0.695 (0.691-0.698)	0.758 (0.755-0.762)	0.792 (0.787-0.796)
HACS	XGB	0.691 (0.688-0.695)	0.761 (0.758-0.764)	0.802 (0.798-0.807)
HACS_IPCW	XGB	0.686 (0.682-0.689)	0.756 (0.753-0.760)	0.801 (0.796-0.806)

Table A2. Performance of Vanilla HACS versus Censoring-Adapted HACS in Terms of IBS

Method	Base classifier	Pre-loan	Post-loan (t_1)	Post-loan (t_2)
HACS	LR	0.440 (0.435-0.445)	0.260 (0.257-0.263)	0.130 (0.128-0.132)
HACS_IPCW	LR	0.449 (0.444-0.455)	0.260 (0.257-0.263)	0.123 (0.121-0.125)
HACS	RF	0.405 (0.400-0.410)	0.251 (0.248-0.254)	0.125 (0.123-0.126)
HACS_IPCW	RF	0.415 (0.409-0.420)	0.251 (0.247-0.254)	0.119 (0.117-0.120)
HACS	XGB	0.410 (0.405-0.415)	0.250 (0.247-0.253)	0.123 (0.121-0.125)
HACS_IPCW	XGB	0.416 (0.411-0.422)	0.252 (0.249-0.256)	0.119 (0.117-0.121)

Appendix B

Discrimination Performance over Time Horizons

Tables B1 and B2 summarize the discrimination performance of HACS versus benchmarked methods for different time horizons (t_1, t_2, and t_3 at the pre-loan stage and $t_2|t_1$, $t_3|t_1$, and $t_3|t_2$ at the post-loan stage). The results across the three performance metrics show similar patterns, indicating that the results are quite robust. We tested the statistical significance of the comparisons between HACS and the benchmarked methods using a non-parametric Friedman test. Overall, the differences across the six MPDP methods were statistically significant at both the pre-loan ($\chi^2 = 2391.336$, $p < 0.001$) and post-loan ($\chi^2 = 650.050$, $p < 0.001$) stages. Further pairwise comparisons show that, statistically, HACS significantly ($p < 0.001$) outperformed all multi-label learning and survival analysis methods at both stages.

Table B1. Discrimination Performance at the Pre-Loan Stage

Method	Time	AUC	KS	H-measure
BR_RF	t_1	0.719 (0.713-0.726)	0.349 (0.339-0.358)	0.233 (0.224-0.241)
BR_XGB		0.715 (0.709-0.721)	0.344 (0.335-0.354)	0.221 (0.212-0.229)
HACS_RF		**0.753** (0.747-0.759)	**0.404** (0.394-0.414)	**0.278** (0.270-0.287)
HACS_XGB		0.737 (0.731-0.744)	0.379 (0.368-0.389)	0.253 (0.244-0.262)
MCM		0.694 (0.688-0.701)	0.314 (0.304-0.324)	0.192 (0.184-0.200)
RSF		0.723 (0.717-0.729)	0.361 (0.351-0.371)	0.238 (0.229-0.246)
BR_RF	t_2	0.711 (0.707-0.715)	0.326 (0.319-0.333)	0.220 (0.214-0.225)
BR_XGB		0.696 (0.692-0.700)	0.304 (0.296-0.311)	0.189 (0.184-0.195)
HACS_RF		**0.722** (0.717-0.726)	**0.339** (0.332-0.346)	**0.230** (0.224-0.236)
HACS_XGB		0.712 (0.707-0.716)	0.323 (0.316-0.331)	0.206 (0.200-0.211)
MCM		0.677 (0.673-0.682)	0.276 (0.269-0.283)	0.154 (0.149-0.159)
RSF		0.710 (0.706-0.714)	0.325 (0.318-0.331)	0.214 (0.208-0.220)
BR_RF	t_3	0.705 (0.702-0.709)	0.315 (0.309-0.320)	0.203 (0.198-0.208)
BR_XGB		0.693 (0.689-0.696)	0.288 (0.282-0.294)	0.177 (0.172-0.181)
HACS_RF		**0.711** (0.707-0.715)	**0.318** (0.312-0.324)	**0.208** (0.203-0.213)
HACS_XGB		0.700 (0.696-0.704)	0.299 (0.293-0.305)	0.184 (0.180-0.189)
MCM		0.671 (0.668-0.674)	0.258 (0.252-0.263)	0.141 (0.136-0.145)
RSF		0.700 (0.697-0.704)	0.304 (0.298-0.310)	0.194 (0.189-0.199)

Table B2. Discrimination Performance at the Post-Loan Stage

Method	Time	AUC	KS	H-measure
BR_RF	$t_2\|t_1$	0.801 (0.795-0.806)	0.513 (0.504-0.521)	0.375 (0.367-0.384)
BR_XGB		0.799 (0.794-0.805)	0.505 (0.495-0.514)	0.372 (0.363-0.381)
HACS_RF		**0.810** (0.804-0.815)	**0.523** (0.514-0.532)	**0.389** (0.381-0.397)
HACS_XGB		0.806 (0.800-0.811)	0.516 (0.507-0.525)	0.389 (0.380-0.398)
MCM		0.798 (0.792-0.803)	0.492 (0.482-0.501)	0.363 (0.354-0.371)
RSF		0.804 (0.799-0.810)	0.521 (0.512-0.531)	0.383 (0.375-0.392)
BR_RF	$t_3\|t_1$	0.777 (0.773-0.781)	0.447 (0.440-0.455)	0.314 (0.307-0.321)
BR_XGB		0.780 (0.776-0.784)	0.445 (0.438-0.452)	0.313 (0.306-0.320)
HACS_RF		**0.781** (0.777-0.786)	0.447 (0.440-0.455)	**0.319** (0.312-0.326)
HACS_XGB		0.781 (0.777-0.785)	**0.448** (0.440-0.456)	0.318 (0.311-0.325)
MCM		0.768 (0.764-0.772)	0.426 (0.419-0.434)	0.290 (0.283-0.296)
RSF		0.776 (0.772-0.781)	0.443 (0.435-0.451)	0.314 (0.307-0.321)
HACS_RF	$t_3\|t_2$	0.815 (0.809-0.821)	0.528 (0.518-0.539)	0.390 (0.379-0.401)
HACS_XGB		0.828 (0.822-0.833)	0.552 (0.541-0.562)	0.409 (0.398-0.420)
HACS_RF		0.823 (0.817-0.829)	0.542 (0.532-0.553)	0.403 (0.393-0.414)
HACS_XGB		**0.834** (0.828-0.839)	**0.560** (0.550-0.570)	**0.417** (0.406-0.428)
MCM		0.822 (0.816-0.828)	0.530 (0.520-0.540)	0.389 (0.379-0.400)
RSF		0.816 (0.810-0.822)	0.532 (0.522-0.542)	0.394 (0.383-0.404)

Tables B3 and B4 summarize the discrimination performance of HACS versus the survival analysis methods (MCM, COX, and RSF) at the full prediction horizon at the pre-loan and post-loan stages, respectively. BR and HACS are equivalent at the full prediction horizon (i.e., all reduced to a single default prediction problem for 12 months).

Table B3. Discrimination Performance at the Pre-Loan Stage

Method	Time	AUC	KS	H-measure
HACS_RF	t_4	**0.707** (0.704-0.711)	**0.307** (0.302-0.313)	**0.200** (0.196-0.205)
HACS_XGB		0.697 (0.693-0.700)	0.288 (0.282-0.294)	0.176 (0.172-0.181)
MCM		0.672 (0.669-0.675)	0.256 (0.250-0.261)	0.136 (0.133-0.140)
RSF		0.701 (0.698-0.705)	0.298 (0.293-0.304)	0.190 (0.186-0.194)

Table B4. Discrimination Performance at the Post-Loan Stage

Method	Time	AUC	KS	H-measure
HACS_RF	$t_4 \mid t_1$	**0.767** (0.764-0.771)	**0.411** (0.404-0.417)	**0.285** (0.280-0.290)
HACS_XGB		0.766 (0.763-0.769)	0.411 (0.405-0.417)	0.285 (0.279-0.290)
MCM		0.753 (0.749-0.757)	0.394 (0.387-0.400)	0.255 (0.249-0.261)
RSF		0.765 (0.761-0.768)	0.409 (0.403-0.415)	0.283 (0.278-0.288)
HACS_RF	$t_4 \mid t_2$	0.796 (0.792-0.801)	0.476 (0.468-0.484)	0.339 (0.330-0.347)
HACS_XGB		**0.805** (0.800-0.809)	**0.487** (0.480-0.495)	**0.346** (0.338-0.354)
MCM		0.792 (0.788-0.796)	0.459 (0.451-0.467)	0.317 (0.309-0.325)
RSF		0.795 (0.790-0.799)	0.479 (0.471-0.486)	0.337 (0.330-0.345)

Appendix C

Robustness Check

To create a fair comparison between HACS and multi-label learning methods without other confounding factors that may influence performance, we selected two representative base classifiers (RF and XGB) and kept the default parameter settings. As a robustness check, we also examined some of these factors by (1) adding a new base classifier ANN and (2) tuning parameters for each base classifier. We used nested cross-validation (10-fold split in both inner and outer loops) with grid search for parameter tuning. For RF, we selected the parameter "mtry" (number of features randomly sampled as candidates at each split) from 2 to 6 in increments of 1. For XGB, we selected the parameters "eta" (learning rate) from 0.2 to 0.4 in increments of 0.05 and "max_depth" (maximum depth of a tree) from 4 to 8 in increments of 1. For ANN, we set up two hidden layers with the ReLU activation function and selected the number of neurons in each hidden layer from 8 to 128 in double increments (i.e., $5 \times 5 = 25$-dimensional parameter space).

Tables C1 and C2 summarize the discrimination performance with parameter tuning at pre-loan and post-loan stages, respectively. The results show that HACS outperformed BR, with all base classifiers (RF, XGB, and ANN), in terms of all performance metrics (AUC, KS, and H-measure), at all prediction horizons (from t_1 to t_3 at the pre-loan stage and $t_2|t_1$, $t_3|t_1$, and $t_3|t_2$ at the post-loan stage).

Table C1. Discrimination Performance with Parameter Tuning at the Pre-Loan Stage

Method	Base classifier	Time	AUC	KS	H-measure
BR	RF	t_1	0.716 (0.710-0.722)	0.348 (0.338-0.357)	0.230 (0.221-0.238)
HACS	RF		**0.752** (0.746-0.758)	**0.404** (0.394-0.414)	**0.279** (0.269-0.288)
BR	XGB		0.719 (0.712-0.725)	0.352 (0.343-0.362)	0.221 (0.213-0.229)
HACS	XGB		0.738 (0.731-0.744)	0.380 (0.371-0.389)	0.256 (0.247-0.265)
BR	ANN		0.696 (0.690-0.703)	0.328 (0.318-0.338)	0.194 (0.186-0.202)
HACS	ANN		0.710 (0.703-0.717)	0.339 (0.329-0.350)	0.217 (0.209-0.225)
BR	RF	t_2	0.710 (0.706-0.714)	0.326 (0.319-0.332)	0.213 (0.208-0.219)
HACS	RF		**0.721** (0.717-0.725)	**0.337** (0.330-0.344)	**0.229** (0.223-0.235)
BR	XGB		0.697 (0.693-0.701)	0.306 (0.299-0.313)	0.190 (0.185-0.196)
HACS	XGB		0.711 (0.707-0.715)	0.321 (0.314-0.328)	0.208 (0.203-0.214)
BR	ANN		0.684 (0.679-0.689)	0.289 (0.282-0.297)	0.166 (0.160-0.172)
HACS	ANN		0.690 (0.685-0.695)	0.291 (0.283-0.299)	0.174 (0.169-0.180)
BR	RF	t_3	0.705 (0.701-0.708)	0.312 (0.307-0.318)	0.201 (0.195-0.206)
HACS	RF		**0.711** (0.707-0.714)	**0.317** (0.311-0.323)	**0.208** (0.203-0.213)
BR	XGB		0.694 (0.690-0.697)	0.290 (0.284-0.297)	0.176 (0.171-0.181)
HACS	XGB		0.699 (0.695-0.703)	0.294 (0.288-0.301)	0.187 (0.182-0.192)
BR	ANN		0.682 (0.679-0.686)	0.273 (0.267-0.279)	0.158 (0.154-0.162)
HACS	ANN		0.685 (0.682-0.689)	0.276 (0.270-0.282)	0.161 (0.156-0.165)

Table C2. Discrimination Performance with Parameter Tuning at the Post-Loan Stage

Method	Base classifier	Time	AUC	KS	H-measure
BR	RF	$t_2\|t_1$	0.806 (0.800-0.812)	0.519 (0.509-0.528)	0.383 (0.375-0.392)
HACS	RF		**0.810** (0.805-0.816)	**0.522** (0.513-0.532)	**0.391** (0.382-0.399)
BR	XGB		0.800 (0.794-0.805)	0.506 (0.496-0.515)	0.376 (0.367-0.385)
HACS	XGB		0.807 (0.802-0.813)	0.513 (0.503-0.523)	0.388 (0.378-0.397)
BR	ANN		0.797 (0.792-0.803)	0.505 (0.494-0.515)	0.374 (0.365-0.382)
HACS	ANN		0.805 (0.800-0.810)	0.512 (0.503-0.522)	0.385 (0.377-0.394)
BR	RF	$t_3\|t_1$	0.778 (0.774-0.783)	0.448 (0.440-0.455)	0.318 (0.311-0.325)
HACS	RF		**0.782** (0.777-0.786)	0.449 (0.441-0.456)	**0.319** (0.312-0.325)
BR	XGB		0.782 (0.778-0.786)	0.448 (0.441-0.456)	0.318 (0.311-0.325)
HACS	XGB		0.782 (0.778-0.786)	0.448 (0.440-0.455)	0.318 (0.311-0.326)
BR	ANN		0.780 (0.776-0.784)	0.450 (0.443-0.457)	0.315 (0.309-0.322)
HACS	ANN		0.782 (0.779-0.786)	**0.451** (0.444-0.459)	0.319 (0.312-0.326)
BR	RF	$t_3\|t_2$	0.824 (0.818-0.830)	0.553 (0.543-0.563)	0.411 (0.400-0.422)
HACS	RF		0.823 (0.817-0.829)	0.546 (0.535-0.556)	0.404 (0.393-0.415)
BR	XGB		0.828 (0.823-0.834)	0.551 (0.541-0.562)	0.410 (0.399-0.422)
HACS	XGB		**0.834** (0.828-0.840)	**0.559** (0.548-0.569)	**0.419** (0.408-0.430)
BR	ANN		0.830 (0.824-0.835)	0.551 (0.541-0.561)	0.409 (0.398-0.419)
HACS	ANN		0.833 (0.828-0.839)	0.556 (0.546-0.567)	0.415 (0.404-0.426)

Appendix D

Modern Portfolio Theory

To examine the profitability performance in uneven allocation scenarios (i.e., the weight of each loan could be different), we also simulated selecting loan portfolios based on modern portfolio theory, mathematically represented as the Markowitz model (Ban et al., 2018). Specifically, we first selected portfolios using the profit-ranking results of the PD and ERR strategies and then assigned a weight to each loan in the selected portfolios by solving the optimization model based on the Markowitz model and characteristics of credit scenario:

$$\max_{w} \sum_{i=1}^{n} w_i \cdot \frac{r_i}{\|\boldsymbol{r}\|} - {w_i}^2 \cdot \frac{p_i}{\|\boldsymbol{p}\|}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} w_i = 1$$

$$0 \le w_i \le 2/n \quad i = 1,2,\dots,n$$

where $\boldsymbol{w} = (w_1, w_2, \dots, w_n)$ is the weight vector of a portfolio with the size of n; $\boldsymbol{r} = (r_1, r_2, \dots, r_n)$ is the reward (i.e., interest rate) vector of the portfolio; $\boldsymbol{p} = (p_1, p_2, \dots, p_n)$ is the risk (i.e., default probability in the full horizon) vector of the portfolio. The second constraint ensures the diversification effect of a loan portfolio.

Table D1 summarizes the profitability performance (i.e., average return rate) of each strategy after weight optimization at different portfolio sizes. The ERR strategy using HACS_RF always gave the best profitability performance (i.e., highest average return rate) with a small portfolio size (less than 50), and the PD strategy using RF and the ERR strategy using RSF became comparable when the portfolio size increases. As *portfolio variance* is another important risk measure in modern portfolio theory, especially when average return rates are similar, we further compared the portfolio variance of these two strategies in terms of the average variance of weights (summarized in Table D2). Although the PD strategy using RF and the ERR strategy using RSF could give an average return rate comparable to that of the ERR strategy using HACS_RF, their portfolio variances were strikingly higher than that of the ERR strategy using HACS_RF, with any portfolio size, indicating a higher level of risk as the total capital was mostly allocated to few loans. Overall, the results show that the proposed ERR strategy combined with HACS_RF was able to help investors select loan portfolios with higher profits while preserving the dispersion of loan portfolios.

Table D1. Results of Profitability Performance Using the Modern Portfolio Theory

Size	PD_RF	PD_XGB	ERR_HACS_RF	ERR_HACS_XGB	ERR_MCM	ERR_RSF
10	0.198 (0.195-0.202)	0.190 (0.186-0.194)	**0.218** (0.210-0.226)	0.195 (0.183-0.208)	0.156 (0.141-0.172)	0.209 (0.199-0.218)
20	0.203 (0.200-0.206)	0.199 (0.196-0.202)	**0.212** (0.205-0.219)	0.193 (0.185-0.202)	0.172 (0.162-0.181)	0.207 (0.200-0.214)
30	0.204 (0.201-0.207)	0.200 (0.197-0.203)	**0.209** (0.203-0.215)	0.194 (0.187-0.201)	0.180 (0.172-0.187)	0.207 (0.201-0.213)
40	0.205 (0.203-0.208)	0.200 (0.197-0.204)	**0.208** (0.203-0.213)	0.193 (0.187-0.199)	0.184 (0.177-0.191)	0.206 (0.201-0.211)
50	0.205 (0.203-0.208)	0.201 (0.197-0.204)	**0.208** (0.203-0.212)	0.193 (0.188-0.199)	0.187 (0.181-0.194)	0.204 (0.199-0.209)
60	0.205 (0.203-0.208)	0.201 (0.197-0.204)	**0.207** (0.203-0.212)	0.194 (0.189-0.199)	0.188 (0.183-0.194)	0.205 (0.201-0.210)
70	0.205 (0.203-0.208)	0.200 (0.197-0.203)	**0.206** (0.202-0.210)	0.195 (0.190-0.199)	0.189 (0.184-0.194)	0.205 (0.201-0.209)
80	0.205 (0.203-0.208)	0.201 (0.198-0.204)	**0.205** (0.201-0.208)	0.195 (0.191-0.200)	0.189 (0.184-0.195)	0.204 (0.200-0.208)
90	**0.205** (0.203-0.207)	0.201 (0.199-0.204)	0.204 (0.200-0.207)	0.195 (0.191-0.200)	0.190 (0.185-0.195)	0.204 (0.200-0.208)
100	0.204 (0.202-0.207)	0.202 (0.199-0.204)	**0.204** (0.200-0.207)	0.195 (0.191-0.199)	0.189 (0.184-0.194)	0.204 (0.200-0.208)

Table D2. Results of Portfolio Variance (in Percentage)

	10	20	30	40	50	60	70	80	90	100
PD_RF	50.507	15.100	7.322	4.317	2.885	2.076	1.550	1.196	0.950	0.777
ERR_RSF	10.953	2.583	1.064	0.641	0.502	0.397	0.323	0.297	0.308	0.335
ERR_HACS_RF	6.963	1.810	0.822	0.475	0.341	0.267	0.229	0.205	0.196	0.209