The Deep Promotion Time Cure Model

Victor Medina-Olivares, Stefan Lessmann, and Nadja Klein

***Abstract*— We propose a novel method for predicting time-to-event data in the presence of cure fractions based on flexible survival models integrated into a deep neural network (DNN) framework. Our approach allows for nonlinear relationships and high-dimensional interactions between covariates and survival and is suitable for large-scale applications. To ensure the identifiability of the overall predictor formed of an additive decomposition of interpretable linear and nonlinear effects and potential higher-dimensional interactions captured through a DNN, we employ an orthogonalization layer. We demonstrate the usefulness and computational efficiency of our method via simulations and apply it to a large portfolio of U.S. mortgage loans. Here, we find not only a better predictive performance of our framework but also a more realistic picture of covariate effects.**

***Index Terms*— Credit risk, cure models, deep learning (DL), interpretability, survival analysis.**

I. Introduction

LENDERS employ mathematical models to assist decision-making by estimating each customer's probability of a credit event. These models, known as credit scoring systems, were initially developed to predict the probability of default for specific products. Over time, the use and purpose of these systems have become more diverse and aligned with the lender's strategic goals. Moreover, new computational advancements and the pursuit of better models have urged research on deep-learning (DL) approaches in this field. Gunnarsson et al. [1], conducting a study comparing DL algorithms and their practicality in credit scoring, find that while tree-based ensemble methods are still favored, DL approaches have potential, for example, in handling less traditional data sources. Meanwhile, Stevenson et al. [2] reveal the merit of DL for predicting default in small businesses using text data. In another study by Korangi et al. [3], transformer models were employed to process time-varying covariates, such as accounting metrics from the balance sheet, to predict the bankruptcy of middle-capitalization companies, showing better performance than traditional models. These findings are part of the recent evidence suggesting that DL techniques are promising to improve credit scoring systems and expand the range of data types that can be leveraged in this field.

Until now, most applied DL models to credit risk have focused on classification tasks, where a predefined performance period of a binary decision is established. A different route is that of survival analysis for building scoring systems but is less explored in the DL context [4]. Here, the outcome of interest is the time until an event occurs. One challenge in survival analysis is to reliably describe the distribution of survival times, trying to convey, for example, if all subjects are prone to the event of interest. In credit risk modeling, it is, however, natural to expect that some borrowers will never experience the event, resulting in heavy censoring at the end of the study [5]. In this situation, cure rate models are preferred [6], which extend survival models by including a latent cure fraction. The advantage is that these models allow us to separate the factors that influence the probability of the event occurrence from those that affect its timing.

Another challenge is understanding how the subject-specific features (or covariate effects) relate to survival times. To this end, two main classes of cure models exist: the mixture cure model (MCM) [7] and the promotion time cure model (PTCM) [8], [9]. Although the MCM has been extensively studied in the credit risk literature (see Table I), the PTCM, introduced in the late 1990s, has gone practically under the radar and is the focus of this article.

The MCM assumes a binary response variable in the population that describes those *cured* and those susceptible to the event. This approach has been broadly developed in parametric, semi-parametric, and nonparametric formulations, and to handle continuous, discrete, and longitudinal data (see [10] for a comprehensive review). In contrast, the PTCM, which originates from cancer studies, assumes each subject has unobserved competing risk factors, such as cancer cells. In this situation, a cured patient will have zero cancer cells, while a susceptible patient's event will occur when the first cell develops into a palpable cancer mass. Although initially conceived for tumors, its statistical principles apply to broader contexts. For example, in credit-related applications, competing risk factors include causes for borrower default, such as job loss, inability to work, strategic default, and failed businesses [11].

Manuscript received 29 April 2023; revised 9 February 2024; accepted 1 May 2024. The work of Victor Medina-Olivares and Nadja Klein was supported by the Deutsche Forschungsgemeinschaft (DFG), German Research Foundation through the Emmy Noether under Grant KL 3037/1-1. The work of Stefan Lessmann was supported by the Project "AI4EFin Artificial Intelligence (AI) for Energy Finance," under Romania's National Recovery and Resilience Plan, Apel nr. Romanian translation of National Recovery and Resilience Plan [Planul National de Redresare si Rezilienta (PNRR)]-III-C9-2022-I8, under Contract CF162/15.11.2022. *(Corresponding author: Victor Medina-Olivares.)*

Victor Medina-Olivares and Nadja Klein are with the Chair of Uncertainty Quantification and Statistical Learning, Research Center Trustworthy Data Science and Security, University Alliance Ruhr, 44227 Dortmund, Germany, and also with the Department of Statistics, Technische Universität Dortmund, 44227 Dortmund, Germany (e-mail: victor.medina@tu-dortmund.de).

Stefan Lessmann is with the School of Business and Economics, Humboldt-Universität zu Berlin, 10099 Berlin, Germany, and also with Bucharest University of Economic Studies, 010374 Bucharest, Romania.

This article has supplementary downloadable material available at https://doi.org/10.1109/TNNLS.2024.3398559, provided by the authors.

Digital Object Identifier 10.1109/TNNLS.2024.3398559

Addressing existing challenges, we make three significant contributions in this manuscript: two methodological and one empirical. From a methodological standpoint, first, we reformulate the PTCM using a deep neural network (DNN) architecture. We label our approach *Deep-PTCM*.

The second methodological contribution allows the decomposition of the predictor into linear and nonlinear components, with the latter estimated through a DNN. This separation aims to facilitate the interpretation of covariate effects, a common criticism when applying DL approaches. However, it is known that a neural network (NN) can approximate any continuous function [12], in particular, a linear one. Hence, to avoid identifiability issues, we follow [13] and add an orthogonalization layer. This layer projects the output of the nonlinear component onto the orthogonal complement of the linear one, achieved through a QR decomposition. This greatly enhances interpretability. The novelty of our approach lies in its capability to account for the censoring of certain subjects.

From an empirical perspective and to the best of our knowledge, this is the first study to apply such a general and flexible framework of PTCMs in the credit risk context. First, most of the cure models studied in credit scoring belong to the class of MCMs, leaving the PTCM relatively unexplored. However, we do not find any solid justification in the literature for choosing MCM over PTCM, and its preference may be due to its popularity. Second, none of the cure models, regardless of the class selected, allow for complex and often more realistic nonlinear relationships and interactions between covariates and survival. As we show later, this assumption limits the predictive power, an essential aspect of credit risk management [14]. Concretely, we build a cure model to predict the time to default in a large U.S. mortgage portfolio. We show that the Deep-PTCM significantly outperforms the standard PTCM in calibration and discrimination.

Overall, our Deep-PTCM has the following highly relevant advantages over existing competitors.

1) It generalizes the standard PTCM, which assumes linear dependency in its predictor. This can be seen as a one-layer NN with one unit.
2) It provides more flexibility than traditional estimation pipelines by replacing data preprocessing and feature engineering with a differentiable loss function estimated via gradient descent. That facilitates the model to handle structured and unstructured data such as text and images, broadening its applicability.
3) The employed orthogonalization cell gives access to directly interpretable linear and nonlinear effects, whereas the additional high-order interactions ensure high predictive power.
4) It is scalable since all model parameters are integrated into an end-to-end DNN, making the estimation procedure computationally efficient and easily parallelizable (GPUs/TPUs).
5) Its implementation uses the TensorFlow framework, making it easy to accommodate all layers, optimizers, and features available there.

In particular, 2)–5) provide comparative advantages over recent efforts from medical research by Xie and Yu [15]. The authors propose a PTCM with a DNN component and show it can increase the model's performance compared with nonparametric approaches with splines. However, this is the only work in this respect, indicating that the interface of PTCMs and DNNs is an underexplored area from both a modeling and an application perspective. An example is the estimation procedure, carried out iteratively using the expectation-maximization (EM) algorithm introduced by [16], where the DNN is optimized at each maximization step. That results in a computationally inefficient procedure that limits the approach's materiality in increasingly prevalent big data environments. Through a simulation study, we demonstrate that Deep-PTCM scales better than the approach proposed in [15]. This improvement allows us to estimate the model on a training set with approximately 150k borrowers, the largest in this context (see Table I), in a few minutes rather than hours.

The article is structured as follows: Section II summarizes relevant literature. Section III outlines the PTCM, its reformulation in an end-to-end DNN framework, and efficient estimation for large datasets. Section IV details performance metrics for cure model evaluation. Section V includes two simulation studies: one comparing our approach to Xie and Yu [15], and the second examining the Deep-PTCM's recovery of linear effects with the orthogonalization step. Section VI presents our credit risk study, and Section VII concludes.

II. Related Work

Although the contributions presented have the potential for applications beyond the credit-related context, the motivation for this work arises from the importance of credit-scoring models in a predominantly data-driven industry and the lack of studies combining cure models and DL.

Most of the cure models applied so far belong to the class of MCMs. Tong et al. [17] introduce the MCM and compare its performance to the logistic regression and the Cox Proportional Hazard model (Cox PH), noting the ability to distinguish among borrowers' susceptibility is appealing for risk management. Similarly, Dirick et al. [18] compare the performance of different survival approaches in ten datasets. They find comparable performance between the MCM, Cox PH, and accelerated failure time models, with a promising economic performance by the MCM.

Moreover, Louzada et al. [19] demonstrate that the flexibility of the MCM allows modeling survival data even when the proportional hazard assumption is not satisfied. Extensions to include exogenous time-varying covariates can be found in [20], from a discrete-time perspective, and in [5], for the continuous-time one. Furthermore, Zhang et al. [21] introduce a new MCM to allow the noncured borrowers to be susceptible to a subset of risks instead of all of them as it is commonly assumed in competing risk settings.

While much of the literature focuses on the MCM, some work, all from the same group, have studied the PTCM. Namely, in [22], a PTCM is applied to relate the intensity of default and recovery rates in a Brazilian loan portfolio. This study, however, does not include covariates in the model. In addition, in [11], different activation mechanisms of the PTCM are analyzed. A bivariate survival process is considered

TABLE I
LIST OF REFERENCES IN THE CREDIT RISK LITERATURE WITH CURE MODELS. T|N|X REFER TO THE MAXIMUM PERFORMANCE PERIODS (MONTHS), THE SAMPLE SIZE, AND THE NUMBER OF COVARIATES (BEFORE PREPROCESSING)

Reference	Class	Data	T—N—X	Metric(s)	Non-linear	DL
Tong et al. [17]	MCM	UK Personal loans	36—27527—14	AUC; H-measure; KS	✗	✗
De Leonardis et al. [20]	MCM	SMEs	84—27579—9	AUC	✗	✗
Louzada et al. [19]	MCM	Brazilian Personal loans	12—40115—2	Expected loss	✗	✗
Liu et al. [31]	MCM	Chinese Mortgage loans	36—14068—14	Gini; KS	✗	✗
Dirick et al. [32]	MCM	UK Personal loans	36—7521—8	AIC; AUC	✗	✗
Wycinka & Jurkiewicz [33]	MCM	Polish Personal loans	24—5000—12	AUC; H-measure	✗	✗
Dirick et al. [18]	MCM	10 datasets (personal loans and SMEs)	60—80641—31	AUC; MAE; MSE; Financial metrics	✗	✗
Dirick et al. [5]	MCM	Belgian Personal loans	36—20000—13	AIC	✗	✗
Zhang et al. [21]	MCM	P2P loans	36—50000—5	AIC	✗	✗
Jiang et al. [26]	MCM	P2P loans	12—52573—31	AUC; Concordance corr.; H-measure; KS	✓	✗
Dirick et al. [34]	MCM	UK Personal loans	36—7521—8	AIC; BIC; MAE; MSE	✗	✗
Oliveira & Louzada [22]	PTCM	Brazilian Personal loans	84—20000—0	Expected LGD	✗	✗
Barriga et al. [11]	PTCM	Brazilian Personal loans	36—236—1	AIC; BIC; RMSE	✗	✗
Cancho et al. [23]	PTCM	Brazilian customer data	36—1188—2	AIC; BIC; DIC	✗	✗
Ribeiro de Oliveira Jr et al. [24]	PTCM	Brazilian Personal loans	60—5733—3	AIC	✗	✗
Toledo et al. [25]	PTCM	Brazilian loans	24—9645—4	AIC; BIC	✗	✗
This work	PTCM	US Mortgage loans	**154—149561**—19	AUC_{cure}; IBS	✓	✓

in [23], and de Oliveira Jr et al. [24] extend it to account for events in time zero. More recently, Toledo et al. [25], using a baseline risk function following a Gompertz distribution, also allow events at time zero with fractions incorporating covariate effects.

Table I compares relevant approaches to cure models applied in credit risk. Except for [26], which incorporates random forests in the incidence model and linear effects in the latency model, all other contributions, whether based on MCMs or PTCMs, assume linear covariate effects—an assumption often deemed unrealistic and overly simplistic. Additionally, in the context of cure rate models incorporating nonlinear effects, traditional nonparametric methods like smoothing splines struggle with high-dimensional interactions [15]. The Deep-PTCM addresses these limitations as it not only captures complex covariate effects but is also scalable by leveraging the flexibility of DL.

In recent years, numerous DL methods for survival analysis without a cure fraction have emerged, leveraging the benefits of NNs while building upon established statistical approaches. Notably, [27] introduces a Cox PH model parameterized by an NN. While excelling at handling complex data representations, it remains constrained to the Cox PH structure. A more recent development is DeepHit [28] and its extension to handle time-varying covariates [29]. This model discretizes the survival timeline and directly learns first-hitting times among competing risks without assuming the underlying stochastic process. However, its architecture confines the prediction of failure times to a fixed-size discrete set, potentially impractical for scenarios with extended survival horizons. These represent some of the most impactful efforts, and for a more comprehensive study, refer to [30]. Despite not addressing the cure context, these advancements underscore the significance of exploring DL in time-to-event prediction.

III. METHODOLOGY

In this section, we briefly review the PTCM in Section III-A, while Sections III-B and III-C detail the Deep-PTCM and an efficient estimation algorithm using existing DL libraries. This allows the framework to be applied to large and unstructured datasets.

Throughout, consider a population of N subjects ($i = 1, \ldots, N$) with covariate vectors $\mathbf{x}_i \in \mathbb{R}^q$. The time to event and right-censoring times for subject i are denoted by T_i^* and C_i, respectively. The observed event time is $T_i = \min\{C_i, T_i^*\}$. Let t_i represent the realization of T_i, and δ_i be the event indicator (1 if the event occurs at time t_i, and 0 otherwise). Population-level notation is represented by $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^\top \in \mathbb{R}^{N\times q}$, $\mathbf{t} = (t_1, \ldots, t_N)^\top$, and $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_N)^\top$.

A. Promotion Time Cure Model

The PTCM assumes that subject i has K_i unobserved competing risk factors, each of which can lead to the occurrence of the event. Furthermore, it is assumed that K_i is Poisson-distributed with mean $\theta(\mathbf{x}_i) > 0$. The cured subjects are those for whom $K_i = 0$. Denote by Y_{ik}, $k = 1, \ldots, K_i$, the random time for the kth risk factor. Given K_i, the variables Y_{ik} are assumed to be mutually independent, independent

of K_i, and distributed with cumulative distribution function (cdf) $F(t)$. Then, T_i^* is defined as the time elapsed until the first unobserved competing risk factor is triggered, that is, $T_i^* = \min\{Y_{i1}, \ldots, Y_{iK_i}\}$, such that

$$\begin{aligned} &S_p(t; \mathbf{x}_i) \\ &= P(\text{no event by time } t; \mathbf{x}_i) \\ &= P(K_i = 0; \mathbf{x}_i) + P(Y_{i1} > t, \ldots, Y_{iK_i} > t, K_i \geq 1; \mathbf{x}_i) \\ &= \exp(-\theta(\mathbf{x}_i)) + \sum_{j=1}^{\infty} \frac{[(1 - F(t))\theta(\mathbf{x}_i)]^j}{j!} \exp(-\theta(\mathbf{x}_i)) \\ &= \exp(-\theta(\mathbf{x}_i)F(t)). \end{aligned} \tag{1}$$

The cure fraction is $\lim_{t\to\infty} S_p(t; \mathbf{x}_i) = \exp(-\theta(\mathbf{x}_i))$. Note that, since $\lim_{t\to\infty} S_p(t; \mathbf{x}_i)$ can be positive, S_p is not a proper survival function. We deliberately call it the *survival function of the population* and add the subindex p to differentiate it from $S(t) = 1 - F(t)$, the (proper) survival function of the risk factors.

The traditional PTCM relates θ to a linear predictor of the covariates, that is, $\eta(\mathbf{x}_i) = \mathbf{w}^\top \mathbf{x}_i + b$, through $\theta(\mathbf{x}_i) = \exp(\eta(\mathbf{x}_i))$, where $\mathbf{w} \in \mathbb{R}^q$ is the vector of regression coefficients, and $b \in \mathbb{R}$ is the intercept [35]. The hazard function of the population is then $h_p(t; \mathbf{x}_i) = \exp(\mathbf{w}^\top \mathbf{x}_i + b) f(t)$, with $f(t) = \mathrm{d}F(t)/\mathrm{d}t$. Since the hazard function preserves proportionality, the PTCM is also known as the proportional hazard cure model [36].

The PTCM has been studied and extended in several directions in the statistical community. For example, an EM algorithm to estimate the model with missing covariates [16], extensions to handle interval-censored data [37] and to include random effects [38], or categorical time-varying covariates [39], latent risk classes [40], and longitudinal covariates [41] have been developed. While the linear relationship between covariates and survival in the PTCM facilitates interpretation, it also restricts the regression predictor considerably. Nonparametric approaches have recently been explored to relax the linearity assumption, in particular, to model univariate covariate effects by including smoothing splines [42], [43]. Still, these methods do not account for higher-order covariate interactions and struggle to handle large-scale applications. These limitations motivate the Deep-PTCM presented next.

B. Deep-PTCM

We redefine the PTCM as an end-to-end DNN architecture, which we denote by *Deep-PTCM*. This approach lets us consider not only linear and nonlinear relationships but also high-dimensional interactions between covariates and survival.

1) Predictor Structure: In the Deep-PTCM, the predictor $\eta : \mathbb{R}^q \to \mathbb{R}$ is a general continuous function rather than a linear combination of the covariates $\mathbf{x}_i$. Specifically, we proceed similar to [15] and model η through a DNN. According to the universal approximation theorem [12], we know that an NN, under certain conditions, can approximate any continuous function. Moreover, the linear specification described in Section III-A, $\eta(\mathbf{x}_i) = \mathbf{w}^\top \mathbf{x}_i + b$, can also be parameterized with an NN consisting of a single layer with one neuron.

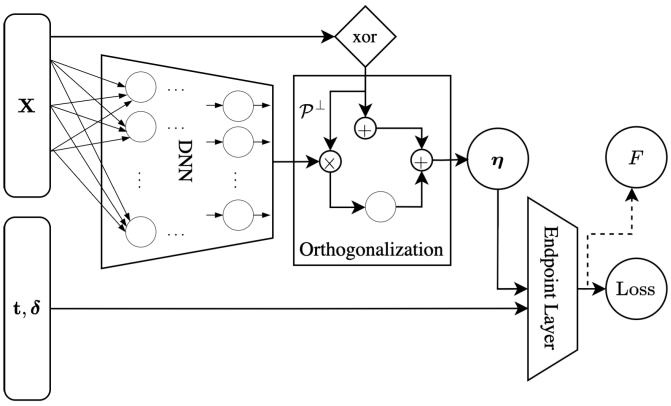

Fig. 1. Generic representation of the Deep-PTCM architecture.

Therefore, the Deep-PTCM subsumes the traditional PTCM as a special case.

2) Identifiability via Orthogonalization: The Deep-PTCM is designed to be flexible enough to represent complex covariate relationships and handle unstructured data, such as images or text. However, there are situations where the goal is to identify whether the predictor η exhibits structured linear effects for ease of interpretation. To achieve this, we enable the framework to estimate η as an identifiable sum of linear (η^{lin}) and nonlinear (η^{non}) predictor components, where the latter may also comprise DNN structures.

Drawing inspiration from the orthogonalization procedure introduced in [13], we ensure empirical identifiability of $\boldsymbol{\eta} = (\eta(\mathbf{x}_1), \ldots, \eta(\mathbf{x}_N))^\top$ in the Deep-PTCM by computing orthogonal projection matrices $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{P}}^\perp \in \mathbb{R}^{N\times N}$, such that $\boldsymbol{\eta}^{\text{lin}} = \boldsymbol{\mathcal{P}}\boldsymbol{\eta}$ and $\boldsymbol{\eta}^{\text{non}} = \boldsymbol{\mathcal{P}}^\perp\boldsymbol{\eta}$.). The projection matrices are obtained through a QR decomposition of $\tilde{\mathbf{X}} = [\mathbb{1}_N, \mathbf{X}]$, where $\mathbb{1}_N$ is an N-dimensional vector of ones. In other words, $\tilde{\mathbf{X}} = \mathbf{QR}$, where $\mathbf{Q}$ and $\mathbf{R}$ are $N \times N$ orthonormal and upper-triangular matrices, respectively. The resulting projection matrices are then given by $\boldsymbol{\mathcal{P}} = \mathbf{QQ}^\top$ and $\boldsymbol{\mathcal{P}}^\perp = \mathbf{I}_{N\times N} - \mathbf{QQ}^\top$, where $\mathbf{I}_{N\times N}$ is the N-dimensional identity matrix.

3) Architecture: Fig. 1 illustrates the core concept behind the Deep-PTCM architecture. The `DNN` block takes the covariates $\mathbf{X}$ as inputs, allowing us to tailor the architecture to the specific nature of the data. For instance, when dealing with unstructured data like images, convolutional NNs [44], [45] may be incorporated into this block. The output then proceeds to the `Orthogonalization` layer. Here, $\boldsymbol{\eta}$ is built by the sum of a linear predictor and the appropriate projection of the `DNN` block output into the orthogonal complement of that linear predictor (e.g., a subset of covariates). If no specific separation is required, the whole predictor $\boldsymbol{\eta}$ can also be estimated without this decomposition, such that $\boldsymbol{\mathcal{P}}^\perp = \mathbf{I}_{N\times N}$.

The `Endpoint Layer` takes the inputs $\mathbf{t}$, $\boldsymbol{\delta}$, and $\boldsymbol{\eta}$ and passes them into the loss function, which in the Deep-PCTM is the negative log-likelihood, as introduced in (2). In addition, the `Endpoint Layer` specifies the cdf F, which is required to calculate the loss, with common choices in the PTCM context being the Weibull or piecewise exponential functions. Following the approach outlined in [15] and [16], we opt for

the latter in Section III-C, but other specifications can be easily accommodated.

4) Illustration: For illustrative purposes, let us consider a DNN specified as a fully connected feedforward NN (FNN) with L hidden layers. Specifically, suppose that layer l ($l = 1, \ldots, L$) has n_l neurons, hence the output of the lth layer, $\mathbf{g}^{(l)} : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$ ($n_0 = q$), follows:

$$\mathbf{g}^{(l)}(\mathbf{z}) = \left[g_1^{(l)}(\mathbf{z}), \ldots, g_{n_l}^{(l)}(\mathbf{z})\right]^\top$$

with $g_m^{(l)}(\mathbf{z}) = a^{(l)}(\mathbf{w}_m^{(l)\top}\mathbf{z} + b_m^{(l)})$, $m = 1, \ldots, n_l$, $a^{(l)} : \mathbb{R} \to \mathbb{R}$ is the activation function for the lth layer, $\mathbf{w}_m^{(l)} \in \mathbb{R}^{n_{l-1}}$ are the weights associated with the mth neuron of the lth layer, $b_m^{(l)} \in \mathbb{R}$ is the corresponding intercept, and $\mathbf{z} = \mathbf{g}^{(l-1)}(\mathbf{z})$ is the output of the $(l-1)$th layer ($\mathbf{g}^{(0)}(\mathbf{z}) = \mathbf{X}$). Many activation functions have been proposed (see [46], Ch. 5.1.2). One popular choice is the rectified linear unit (ReLU) [47], which is defined as $a(x) = \max(0, x)$.

Therefore, when an FNN is considered in the `DNN` block, its output is a vector $\tilde{\boldsymbol{\eta}} \in \mathbb{R}^{n_L}$ formed by the composition of the L hidden layers, that is, $\tilde{\boldsymbol{\eta}}(\mathbf{x}_i) = (\mathbf{g}^{(L)} \circ \cdots \circ \mathbf{g}^{(2)} \circ \mathbf{g}^{(1)})(\mathbf{x}_i)$. Moreover, if no orthogonalization is performed, $\eta(\mathbf{x}_i)$ is simply computed through a final linear layer, that is, $\eta(\mathbf{x}_i) = \mathbf{w}^{(L+1)\top}\tilde{\boldsymbol{\eta}}(\mathbf{x}_i) + b^{(L+1)}$. On the other hand, if orthogonalization is carried out for all covariates, then

$$\boldsymbol{\eta} = \mathbf{X}\mathbf{w}^{\text{lin}} + b^{\text{lin}}\mathbb{1}_N + \boldsymbol{\mathcal{P}}^{\perp}\begin{pmatrix}\tilde{\boldsymbol{\eta}}(\mathbf{x}_1)^\top \\ \vdots \\ \tilde{\boldsymbol{\eta}}(\mathbf{x}_N)^\top\end{pmatrix}\mathbf{w}^{(L+1)}$$

where $\mathbf{w}^{\text{lin}} \in \mathbb{R}^q$ and $b^{\text{lin}} \in \mathbb{R}$ are, respectively, the vector of linear coefficients and the intercept.

C. Estimation of the Deep-PTCM

1) Training: Traditionally, estimation in the PTCM is carried out using the EM algorithm, where the number of risk factors K_i for subject i is treated as missing data [16]. As we illustrate in Section V, however, this approach does not scale well when considering NNs and large datasets. To overcome this computational limitation, we present an end-to-end framework to estimate both the predictor $\boldsymbol{\eta}$ and the parameters associated with F through a DNN optimization problem. For that, first, note that the log-likelihood is

$$\begin{aligned} l(\boldsymbol{\eta}, F) &= \sum_{i=1}^{N} \delta_i \log(h_p(t_i; \mathbf{x}_i)) + \log(S_p(t_i; \mathbf{x}_i)) \\ &= \sum_{i=1}^{N} \delta_i[\eta(\mathbf{x}_i) + \log(f(t_i))] - \exp(\eta(\mathbf{x}_i))F(t_i). \quad (2) \end{aligned}$$

Considering F as a piecewise exponential function, we partition the length of the study in J intervals according to the distribution of the events, that is, $u_0 = 0 < u_1 < \cdots < u_J$ with $u_J > \max_{i\in\{1,\ldots,N\}} t_i$. In each interval $(u_{j-1}, u_j]$, the hazard function of the competing risk factors is assumed to be constant. Denote these constants by λ_j, $j = 1, \ldots, J$. Thus, for $t \in (u_{j-1}, u_j]$, F and f can be expressed as $F(t) = 1 - \exp[-\lambda_j(t - u_{j-1}) - \sum_{s=1}^{j-1}\lambda_s(u_s - u_{s-1})]$ and $f(t) = \lambda_j \exp[-\lambda_j(t - u_{j-1}) - \sum_{s=1}^{j-1}\lambda_s(u_s - u_{s-1})]$.

We train this model efficiently using backpropagation [44]. This process includes: 1) initializing the weights of the network randomly; 2) feeding the input data through the Deep-PTCM architecture in Fig. 1 and evaluating the loss; and 3) adjusting the weights to minimize the loss in Equation (2) by backpropagation. Steps 2) and 3) are repeated by feeding the input data through the network, calculating the loss, and adjusting the weights until the loss in a validation set is minimized.

2) Prediction: Once the network is optimized, its estimated weights can be used for prediction. Specifically, with the trained `DNN` block, we can infer the predictor $\boldsymbol{\eta}$ for new data. Additionally, we can retrieve the weights associated with the cdf F from the `Endpoint Layer` block and create quantities of interest, such as S_p and S.[1]

3) Implementation: We created a Python package, `deepcure`, for the estimation of the Deep-PTCM, which is available on GitHub.[2] The implementation uses TensorFlow, allowing for seamless integration of all available optimizers and additional features provided by the framework.

IV. Performance Metrics

In the empirical study presented in Section VI, we evaluate the performance of the models under two metrics. The area under the receiver operating characteristic curve (AUC) for cure proportions, which measures how well the model distinguishes between cured and noncured subjects, and the integrated Brier score (IBS), which measures the calibration throughout the whole study period. We review these metrics in the following.

A. AUC for Cure Proportions (AUC_{cure})

The AUC [48] is commonly used in survival analysis to evaluate the performance of a corresponding model. However, the classical formulation does not take cure proportions into account. The receiver operating characteristic curve can be regarded as the curve formed by the true positive rate (TPR) and the false positive rate (FPR) for all cut-off points c in $[0, 1]$. Asano et al. [49] propose the imputation-based AUC for MCMs, and [15] extend it to the PTCM. This version of the AUC evaluates the TPR and FPR concerning the probability of being cured. Denote the estimated long-term survival probability as $\hat{\pi}(\mathbf{x}_i) := \lim_{t\to\infty} \hat{S}_p(t; \mathbf{x}_i) = \exp(-\exp(\hat{\eta}(\mathbf{x}_i)))$, where $\hat{\eta}(\cdot)$ is the point estimate of $\eta(\cdot)$. Therefore, the estimates of TPRs and FPRs, for a given cut-off point c are given by

$$\widehat{\text{TPR}}(c) = \frac{\sum_{i=1}^{N} \mathbb{1}(\hat{\pi}(\mathbf{x}_i) \le c) \cdot (1 - \hat{\pi}(\mathbf{x}_i))}{\sum_{i=1}^{N}(1 - \hat{\pi}(\mathbf{x}_i))}$$

$$\widehat{\text{FPR}}(c) = \frac{\sum_{i=1}^{N} \mathbb{1}(\hat{\pi}(\mathbf{x}_i) \le c) \cdot \hat{\pi}(\mathbf{x}_i)}{\sum_{i=1}^{N} \hat{\pi}(\mathbf{x}_i)}$$

where $\mathbb{1}(A) = 1$ if A is true and zero otherwise. AUC_{cure} is calculated using trapezoidal integration over $c \in [0, 1]$.

[1] We also provide examples of how this is implemented in the GitHub repository.

[2] https://github.com/vhmedina/deepcure

TABLE II
SIMULATION RESULTS FOR ALL N AND SCENARIOS 1–3 BASED ON $R = 100$ INDEPENDENT REPLICATIONS

		EM-PTCM				Deep-PTCM			
N	Scenario	Time (min)	ΔS	ΔS_p	$\Delta\eta$	Time (min)	ΔS	ΔS_p	$\Delta\eta$
50,000	1	32.4	**0.0001**	0.0002	**0.0113**	**0.5**	0.0029	**0.0001**	0.0167
	2	18.5	0.0007	0.0003	0.0166	**0.3**	**0.0005**	**0.0003**	**0.0065**
	3	26.1	**0.0002**	0.0052	0.1037	**1.0**	0.0022	**0.0030**	**0.0886**
100,000	1	106.9	**0.0001**	0.0002	**0.0096**	**0.7**	0.0016	**0.0001**	0.0101
	2	58.4	**0.0006**	0.0003	0.0136	**0.7**	0.0012	**0.0002**	**0.0075**
	3	61.4	0.0002	0.0051	0.1028	**1.2**	**0.0001**	**0.0025**	**0.0673**
150,000	1	187.7	**0.0001**	0.0002	**0.0081**	**0.9**	0.0026	**0.0001**	0.0196
	2	100.8	0.0006	0.0003	0.0125	**0.9**	**0.0001**	**0.0001**	**0.0025**
	3	89.9	**0.0004**	0.0060	0.1187	**1.3**	0.0006	**0.0025**	**0.0692**

TABLE III
SIMULATION RESULTS FOR ALL N AND SCENARIO 4 BASED ON $R = 100$ INDEPENDENT REPLICATIONS WITHOUT (DEEP-PTCM) AND WITH (DEEP-PTCM-ORT) ORTHOGONALIZATION

	Deep-PTCM				Deep-PTCM-Ort			
N	Time	ΔS	ΔS_p	$\Delta\eta$	Time	ΔS	ΔS_p	$\Delta\eta$
50,000	**0.1**	**0.0001**	**0.0006**	**0.0263**	0.2	0.0023	0.0029	0.0985
100,000	**0.2**	0.0001	0.0001	0.0133	1.6	**0.0000**	**0.0000**	**0.0005**
150,000	**0.2**	**0.0000**	0.0001	0.0122	1.5	0.0021	**0.0000**	**0.0106**

B. Integrated BS

The Brier score (BS) [50] corresponds to the mean squared error of the predicted probabilities for binary classification. In the survival context, we can estimate whether a subject survives longer or not at a specific time t. Moreover, Graf et al. [51] introduced a generalization of the BS to handle censoring. This is the version that we use and is specified as

$$\widehat{\mathrm{BS}}(t) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\hat{S}_p(t;\mathbf{x}_i)^2\mathbb{1}(t_i \le t, \delta_i = 1)}{\hat{G}(t_i)} + \frac{(1-\hat{S}_p(t;\mathbf{x}_i))^2\mathbb{1}(t_i > t)}{\hat{G}(t)}\right]$$

where $\hat{G}(\cdot)$ is the Kaplan–Meier estimator of the censoring survival function. By integrating the time-dependent BS over time, we obtain the IBS [51].

V. SIMULATION STUDY

The purpose of this section is threefold. First, we illustrate how the proposed estimation framework scales efficiently to accommodate large sample sizes, commonly seen in the credit context. Second, by using simulation setups identical to those in [15], we show the computational advantages of estimating the model through an end-to-end trained DNN architecture, as opposed to iteratively optimizing the DNN in the maximization step of the EM algorithm. Finally and third, we analyze how the orthogonalization procedure can recover the structured linear predictor without compromising performance compared to the setting without orthogonalization.

A. Simulation Design

We study three sample sizes $N \in \{50\,000, 100\,000, \text{and } 150\,000\}$ subjects. The sample sizes from the works presented in Table I have, on average, ∼30 000 subjects, with a maximum N of 80 641. Therefore, we consider 50 000 as a relevant starting sample size in this context and scale it to 150 000, which is roughly the size of our dataset in Section VI (and the largest we are aware of).

We evaluate four simulated scenarios: three presented in [15] and a fourth in which we added a linear component to study the orthogonalization feature. All scenarios are described in detail in Section A of the Supplementary Material.

B. Summary of Results

Table II shows the comparison between the EM implementation (EM-PTCM) and the Deep-PTCM for each combination of sample size (N) and the first three scenarios. The column Time is the average time in minutes needed to estimate the model for the corresponding setting. Moreover, the columns ΔS, ΔS_p, and $\Delta\eta$ show the mean square difference between the true and estimated quantities S, S_p, and η, respectively. That is, for example, $\Delta S = (1/(R \cdot N))\sum_{r=1}^{R}\sum_{i=1}^{N}(\hat{S}^{(r)}(t_i;\mathbf{x}_i) - S^{(r)}(t_i;\mathbf{x}_i))^2$, where $\hat{S}^{(r)}(\cdot)$ and $S^{(r)}(\cdot)$ are, respectively, the estimated and the true survival function for replication r. The other cases follow analogously. These metrics are evaluated on $R = 100$ holdout datasets with the same data generation process but different random seeds.

For each metric, the best-performing method is shown in bold. We observe that the mean square differences, for S_p and η, are generally lower for Deep-PTCM than for EM-PTCM.

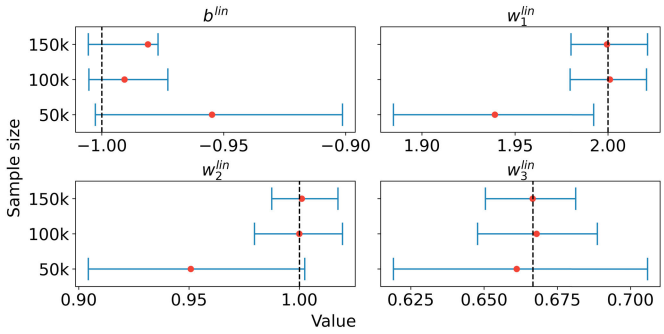

Fig. 2. Linear coefficients estimated by the Deep-PTCM with orthogonalization (Deep-PTCM-Ort).

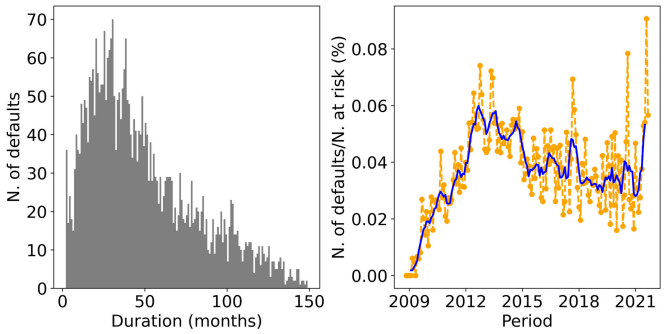

Fig. 3. Single-family loan-level dataset from Freddie Mac. Left: Distribution of default events versus duration. Right: Ratio between the number of default events and borrowers at risk over calendar time. The blue solid line is the moving average for a six-month window.

In the case of S, this difference is not so clear. Nevertheless, since both implementations are meant to estimate the same model, it is not surprising that these results are indeed comparable. The great advantage, however, is that the Deep-PTCM is significantly faster than the EM implementation (more than 100 times for some cases) without compromising accuracy.

To demonstrate the benefits of orthogonalization, we create a fourth setting with $\eta = \eta^{\text{lin}} + \eta^{\text{non}}$ in which $\eta^{\text{lin}} = b^{\text{lin}} + w_1^{\text{lin}} x_1 + w_2^{\text{lin}} x_2 + w_3^{\text{lin}} x_3$ and η^{non} is the one defined in *Scenario 2*. Table III summarizes the results from 100 independent replications comparing the Deep-PTCM and its version with orthogonalization. We note that, in general, the performance of both models is similar concerning the mean square differences. Moreover, Fig. 2 depicts the 2.5%–97.5% range of the estimations of b^{lin}, w_1^{lin}, w_2^{lin} and w_3^{lin} across the 100 replications. We observe a suitable recovery of the true parameter values (dashed vertical lines), especially when increasing the sample size. The Deep-PTCM would not allow us to extract and interpret these coefficients directly.

VI. Application

A. Data

We analyze the publicly available single-family loan-level dataset from Freddie Mac.[3] The dataset contains loan-level origination and monthly performance for fixed-rate U.S. mortgages and is periodically updated. The event of interest is credit default, defined as the time when the loan is 90 or more days past due. "Cure," in this context, represents accounts assumed to be nonsusceptible to default. If the account defaults, we know it is susceptible, whereas when it is censored, the account may or may not be susceptible to default. Censoring includes events such as refinancing or early full repayment. One future research avenue beyond the scope of this work is to explicitly incorporate these and other credit-related events by generalizing the modeling approach to handle multivariate survival data [52].

The training set contains 149 561 loans granted between 2009 and 2011. The test set includes 49 888 loans granted in 2013. The monitoring periods for both sets date from loan origination to December 2021. Fig. 3 shows the default events of the training set over duration and calendar time. The data include eleven categorical and eight numerical variables. Tables I and II in the Supplementary Material describe the categorical and the numerical variables, respectively.

Some categorical variables present high cardinality, which can be challenging from an estimation perspective (poor generalization and high resource usage [53]). Common practice is to either drop these variables, thus discarding valuable information or to transform the attributes into numerical representations, such as target encoding [54]. To compare different preprocessing practices for these variables when estimating the standard PTCM, we employ target encoding, one-hot encoding, and principal component analysis (PCA) for dimensionality reduction. When estimating the Deep-PTCM, we only use one-hot encoding, arguing that the DNN should be able to generalize well without further preprocessing steps. The dimension of the feature space after encoding is 921. We normalize all numerical variables to make the training procedure more efficient.

B. Network Architecture and Training

For the given data, we employ an FNN as `DNN` block (see Fig. 1). This architecture is commonly used in tabular data settings, proving effective for capturing intricate relationships within features. Notably, the FNN is typically not specified with numerous hidden networks for tabular data, potentially making the term "deep" excessive in this context. Despite its application in predicting the time to default in a mortgage portfolio, it is crucial to highlight that the Deep-PTCM implementation, based on TensorFlow, facilitates the easy adaptation of any architecture for diverse applications.

To train the network, the architecture of the FNN needs to be tuned to achieve high prediction accuracy. This includes defining the number of layers, the number of units for each layer, the activation functions, and which optimization algorithm to use. These hyperparameters are not learned during backpropagation and must be set manually. Hyperparameters can significantly impact the performance of the model, and several tuning strategies have been proposed [55]. We use the random search strategy commonly employed in DL [56]. We then chose the combination of hyperparameters that accomplished the minimum average loss in three execution runs per trial with independent random initializations. Running each trial

[3] http://www.freddiemac.com/research/datasets/sf_loanlevel_dataset.page

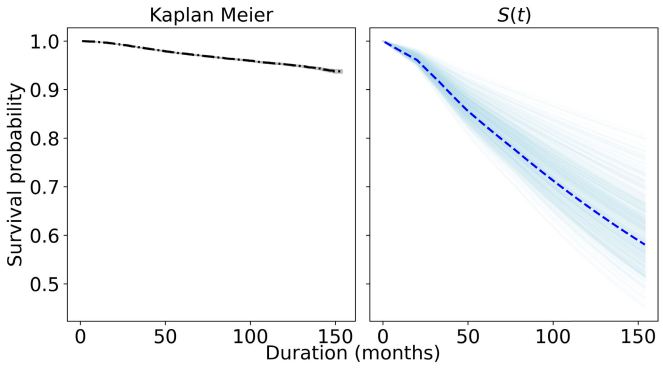

Fig. 4. Left: Kaplan–Meier curve. Right: Survival function of the risk factors $S(t)$ (dashed). The blue-shaded curves are those obtained by 500 bootstrap samples with replacement.

multiple times means avoiding making the final decision strictly dependent on the initial random values.

In addition, to prevent overfitting when training each trial, we use early stopping. We set a maximum number of epochs and a "patience" parameter, which is the number of epochs to wait before deciding to stop the training process. During training, we monitor the model performance on a hold-out or validation set not used in the optimization. If the model performance on the validation set does not improve for a certain number of epochs (the patience parameter), then the training is stopped.

After hyperparameter tuning, the resulting network architecture for the DNN block of the Deep-PTCM has three layers (two hidden plus the output layer). The first two hidden layers have 512 units, ReLU activation functions, and a dropout rate of 0.2. The output layer, representing the predictor η, has one unit and a linear activation function. The best optimization algorithm was found to be (stochastic gradient descent (SGD); see [46], Ch. 12.4) with a learning rate schedule that follows the inverse time decay with an initial rate of 0.01, a decay rate of 0.75, and 100 decay steps. The final model is retrained using the entire training set (see Table III in the Supplementary Material for further details on the search space of the hyperparameters).

C. Results

Fig. 4 illustrates the Kaplan–Meier curve for the whole population (left) and the estimated survival function $S(t) = 1 - F(t)$ of the risk factors (right). The transparent curves represent 500 estimations of the survival function based on resampling with replacement. The interpretation of $S(t)$ is sometimes mistakenly considered as the survival function of noncured subjects (e.g., [15]). But since $F(t)$ is the cdf of the risk factors, and the time to the event is when the first one is triggered, $S(t)$ represents an upper bound of the survival function of the susceptible individuals [57]. Therefore, since the Kaplan–Meier estimator does not control for cured and noncured subjects, it calculates, for instance, that the probability of default, or "not surviving," would be ∼5% after ten years of payments. However, the Deep-PTCM estimates that if the subject belonged to the susceptible population, the probability of default would not be lower than ∼35%.

TABLE IV
AUC$_{\text{CURE}}$ AND IBS RESULTS EVALUATED IN THE TEST SET

	PTCM	ENC-PTCM	PCA-PTCM	Deep-PTCM	Deep-PTCM-Ort
AUC$_{\text{cure}}$	0.85305 (0.00094)	0.82245 (0.00077)	0.84524 (0.00079)	**0.88301** (0.00073)	0.8628 (0.00074)
IBS	0.02317 (0.00062)	0.02351 (0.00063)	0.02299 (0.00061)	**0.02231** (0.00059)	0.02362 (0.00063)

Studying the predictive power of credit scoring models is relevant from the perspective of credit risk management [14]. In particular, we are interested in how the Deep-PTCM performs compared to the traditional PTCM and what gains the Deep-PTCM offers. To this end, we consider five models. The first three correspond to different versions of the PTCM with linear effects in the predictor, where the preprocessing technique of the features gives the distinction. The first one employs one-hot encoding (*PTCM*), the second target encoding (*ENC-PTCM*), and the third PCA for dimensionality reduction (*PCA-PTCM*). The purpose is to apply, on the one hand, the standard practices when modeling in the presence of high-cardinality categorical variables and, on the other, to make the comparison to the deep version more comprehensive. The other two models, *Deep-PTCM* and *Deep-PTCM-Ort*, correspond to the deep approach where the difference is that the latter applies orthogonalization.

Table IV depicts the results obtained on the test dataset for the performance metrics described in Section IV. The numbers in parenthesis are the standard deviations obtained from 100 bootstrap samples of the same size as the original data. We notice that among the three PTCMs, the best discrimination, as measured by AUC$_{\text{cure}}$, is obtained by the version with one-hot encoding (*PTCM*). In terms of calibration, as measured by IBS, the *PCA-PTCM* showed the minimum among the three, but the difference is not significant. Moreover, compared to the deep versions, neither of the three PTCMs with linear predictors performed better in discrimination. Between the two deep versions, we note orthogonalization does not improve predictive performance for this case study. However, interpretability gains are, of course, always present. The best results for discrimination and calibration are accomplished by *Deep-PTCM*, showing an AUC$_{\text{cure}}$ of 0.88, compared to 0.85 from *PTCM*, and an IBS of 0.022, compared to 0.023 from *PCA-PTCM*.

For completeness, we also compare the Deep-PTCM with noncure survival methods, including the traditional benchmark Cox PH [58] and two DL-based models DeepSurv [27] and DeepHit [28]. Our approach demonstrates competitive results across performance metrics, even when not considering the model's ability to distinguish between cure and noncure subjects. Refer to Section F of the Supplementary Material for detailed results.

To illustrate the nonlinear effects of the numerical covariates on the survival in the best performing model *Deep-PTCM*,

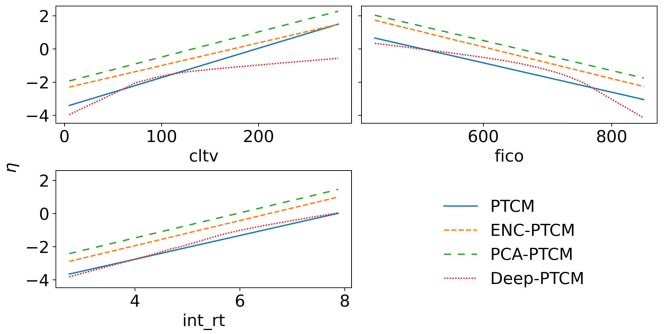

Fig. 5. Comparison of the effect of numerical covariates on the predictor η for four cure models.

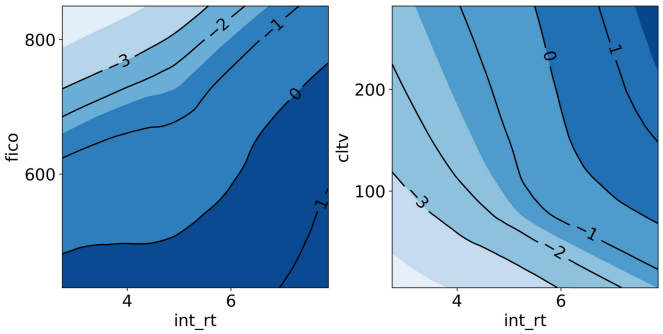

Fig. 6. Bivariate interactions slices of the predictor η for covariates *int_rt-fico* (left) and *int_rt-cltv* (right).

we emulate fictional borrowers with all covariates centered on their base values except the one in question and plot the effect as a function of this variable. Results for the three numerical covariates are presented in Fig. 5, where for comparison, we also add the linear estimates from the *PTCM, the ENC-PTCM, and the PCA-PTCM*. One can first notice that the deep version measures nonlinear relationships between some covariates and the predictor. Two remarkable examples are the combined loan-to-value ratio *cltv* and the borrower's external credit score *fico*. Both covariates have shown conforming signs in the credit risk literature when linearity is assumed [59], [60]. Greater values of *cltv* are associated with a greater risk of default, and greater *fico* values are associated with lower risk. We show the same trend but in a nonlinear way.

In the *cltv* case, we observe that the *Deep-PTCM*, like the *PTCM*, reckon similar risk increments between 50 and 120. However, for values lower than 50 or above 120, the risk estimated by the deep version is lower. For *fico*, we observe that the effect between 550 and 750 calculated by *Deep-PTCM* is more significant than the one shown by *PTCM*. Yet, if the score assigned by the credit bureau is higher than ∼750 (good creditworthiness), the risk measured by the deep version starts to go down comparatively.

In addition, we note that there are covariates, such as the interest rate *int_rt*, where both the *PTCM* and the *Deep-PTCM*, estimate a linear relationship, despite the fact the last one is not restricted to do so. The effects of the other numerical covariates are in the Supplementary Material.

However, the Deep-PTCM can reveal not only the nonlinearities of single covariate effects but also potential interactions. To illustrate this, Fig. 6 visualizes a slice of the bivariate interaction of the pair *int_rt-fico* and *int_rt-cltv*. We observe that the effect of *int_rt* for values of *fico* less than 600 does not change substantially. Similarly, we see that for loans with interest rates close to 6%, the effect of *cltv* is maintained for values greater than 80. The traditional PTCM cannot provide this information.

Overall, we conclude that the Deep-PTCM can recover the simpler embedded PTCM without requiring this (often too restrictive) assumption to be made in advance.

The estimates for the linear effects further support this conclusion. For the variables *cltv*, *fico*, and *int_rt*, these are 0.30/0.26, −0.41/−0.52, and 0.39/0.38 for the *PTCM* and *Deep-PTCM-Ort* models, respectively (see Supplementary Material for more details). However, the increased flexibility of *Deep-PTCM-Ort* comes at the cost of having more unknown parameters and, therefore, greater parameter uncertainty. Nevertheless, given the slightly better prediction performance of *Deep-PTCM* and our simulations, which demonstrate that uncertainty can be significantly reduced with more data, we believe our approach makes a valuable and innovative contribution to future large-scale credit risk applications.

VII. Conclusion

Survival models are a class of supervised learning models that integrate elements of conventional classification and regression to predict the time until subjects experience an event of interest while accounting for censoring. Recent work proposes approaches for DL-based time-to-event modeling and evidences the effectiveness of such models [4], [27], [28]. However, survival models assume that all subjects are, sooner or later, prone to the event. There are applications, such as mortgage default prediction, where it is noticeable that some individuals never experience the event. Under these circumstances, cure rate models are preferable. The literature on credit risk modeling with cure fractions mostly considers the MCM approach (see Table I), whereas another class of models, the PTCM, has not received the same level of attention in this context.

We introduce Deep-PTCM, a novel method for predicting time-to-event data in the presence of cure fractions based on an end-to-end DL framework. Via simulations, we demonstrate the scalability of our method compared to an existing DL-based cure model that employs the EM algorithm, reducing, in some cases, the average training time to the one-hundredth part. In addition, we show that the Deep-PTCM can significantly improve discrimination and calibration metrics compared to the standard PTCM when predicting the time to default in a large U.S. mortgage portfolio. Finally, we explore how the DNN flexibility accounts for the effects of the covariates on the predictor, observing, first, its ability to correctly detect present deviations from linearity in the predictor as assumed by the classic PTCM and, second, to recover it if the evidence supports it.

Our work contributes to the discourse on when and how to use DL in the context of time-to-event modeling by

extending its applicability to scenarios where the assumption of universal susceptibility is implausible, providing evidence of DL's superior performance and scalability compared to non-DL approaches. To achieve this, we cast the PTCM into an end-to-end DL framework that simultaneously estimates covariate effects and survival distribution parameters. This allows us to account for complex and often more realistic nonlinear relationships between covariates and survival. Moreover, it also facilitates computationally efficient model estimation through leveraging parallel computing. Therefore, the Deep-PTCM scales well to large datasets, such as the ones often seen in credit risk applications. Beyond scalability, the Deep-PTCM also offers more flexibility than traditional estimation pipelines by replacing data preprocessing and feature engineering with a differentiable loss function estimated via gradient descent. This facilitates the model to process structured and unstructured data (such as text and images) and provides a more powerful approach to handle high cardinal categorical variables. All these features further broaden the applicability of the Deep-PTCM and, by extension, DL.

A commonly voiced criticism is that DL models are opaque and do not reveal insights into the model-induced feature-to-target relationship. This opaqueness has led to calls for abandoning DL altogether in high-stake settings [61], highlighting the importance of interpretability in model selection. We aim to provide an accurate and transparent framework, enabling decision-makers to make informed decisions without imposing restrictive linearity assumptions a priori, while maintaining interpretability akin to simple linear models. To achieve this, we follow an "interpretability-by-design" approach and incorporate an orthogonalization layer into the Deep-PTCM architecture. This design gives access to directly interpretable linear and nonlinear effects while leveraging the benefits of DL to accommodate additional high-order interactions and ensure high predictive power.

Despite focusing on one specific use case, credit risk modeling, in this article, we emphasize that the statistical and computational properties of the Deep-PTCM ensure broad applicability in modeling time-to-event data in other fields. Our DL-based approach is particularly relevant in the presence of censoring and cure fractions, aiming to enhance predictive performance by exploiting DL's advantages in data representation, accommodating nonlinear covariate effects, and extracting interpretable components—all in a computationally efficient manner. Lastly, the lack of robust, ready-to-use software can hinder the adoption of novel DL-based approaches. By leveraging the TensorFlow library and providing a Python package, `deepcure`, we empower researchers and practitioners to implement the Deep-PTCM, explore its value across different domains, thereby advancing our understanding of when to use DL and how to unlock its potential to create value.

REFERENCES

[1] B. R. Gunnarsson et al., "Deep learning for credit scoring: Do or don't?" *Eur. J. Oper. Res.*, vol. 295, no. 1, pp. 292–305, 2021.

[2] M. Stevenson, C. Mues, and C. Bravo, "The value of text for small business default prediction: A deep learning approach," *Eur. J. Oper. Res.*, vol. 295, no. 2, pp. 758–771, Dec. 2021.

[3] K. Korangi, C. Mues, and C. Bravo, "A transformer-based model for default prediction in mid-cap corporate markets," *Eur. J. Oper. Res.*, vol. 308, no. 1, pp. 306–320, Jul. 2023.

[4] G. Blumenstock, S. Lessmann, and H.-V. Seow, "Deep learning for survival and competing risk modelling," *J. Oper. Res. Soc.*, vol. 73, no. 1, pp. 26–38, Jan. 2022.

[5] L. Dirick, T. Bellotti, G. Claeskens, and B. Baesens, "Macro-economic factors in credit risk calculations: Including time-varying covariates in mixture cure models," *J. Bus. Econ. Statist.*, vol. 37, no. 1, pp. 40–53, Jan. 2019.

[6] V. T. Farewell, "The use of mixture models for the analysis of survival data with long-term survivors," *Biometrics*, vol. 38, no. 4, p. 1041, Dec. 1982.

[7] J. W. Boag, "Maximum likelihood estimates of the proportion of patients cured by cancer therapy," *J. Roy. Stat. Soc. Ser. B: Stat. Methodology*, vol. 11, no. 1, pp. 15–44, Jan. 1949.

[8] A. Y. Yakovlev, A. D. Tsodikov, and B. Asselain, *Stochastic Models of Tumor Latency and Their Biostatistical Applications*. Singapore: World Scientific, 1996.

[9] A. Tsodikov, "Estimation of survival based on proportional hazards when cure is a possibility," *Math. Comput. Model.*, vol. 33, nos. 12–13, pp. 1227–1236, Jun. 2001.

[10] M. Amico, "Cure models in survival analysis: From modelling to prediction assessment of the cure fraction," Ph.D. dissertation, Fac. Econ. Bus., Université Catholique de Louvain, Belgium, 2018.

[11] G. D. C. Barriga, V. G. Cancho, and F. Louzada, "A non-default rate regression model for credit scoring," *Appl. Stochastic Models Bus. Ind.*, vol. 31, no. 6, pp. 846–861, Nov. 2015.

[12] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals, Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989.

[13] D. Rügamer, C. Kolb, and N. Klein, "Semi-structured distributional regression," *Amer. Statistician*, vol. 78, no. 1, pp. 88–99, Jan. 2024.

[14] L. Thomas, J. Crook, and D. Edelman, *Credit Scoring and Its Applications*. Philadelphia, PA, USA: SIAM, 2017.

[15] Y. Xie and Z. Yu, "Promotion time cure rate model with a neural network estimated nonparametric component," *Statist. Med.*, vol. 40, no. 15, pp. 3516–3532, Jul. 2021.

[16] M. Chen and J. G. Ibrahim, "Maximum likelihood methods for cure rate models with missing covariates," *Biometrics*, vol. 57, no. 1, pp. 43–52, Mar. 2001.

[17] E. N. C. Tong, C. Mues, and L. C. Thomas, "Mixture cure models in credit scoring: If and when borrowers default," *Eur. J. Oper. Res.*, vol. 218, no. 1, pp. 132–139, Apr. 2012.

[18] L. Dirick, G. Claeskens, and B. Baesens, "Time to default in credit scoring using survival analysis: A benchmark study," *J. Oper. Res. Soc.*, vol. 68, no. 6, pp. 652–665, Jun. 2017.

[19] F. Louzada, V. Cancho, M. de Oliveira Jr., and Y. Bao, "Modeling time to default on a personal loan portfolio in presence of disproportionate hazard rates," *J. Statist. Appl. Probab.*, vol. 3, no. 3, pp. 1–11, 2014. [Online]. Available: https://ssrn.com/abstract=2416547

[20] D. De Leonardis and R. Rocci, "Default risk analysis via a discrete-time cure rate model," *Appl. Stochastic Models Bus. Ind.*, vol. 30, no. 5, pp. 529–543, Sep. 2014.

[21] N. Zhang, Q. Yang, A. Kelleher, and W. Si, "A new mixture cure model under competing risks to score online consumer loans," *Quant. Finance*, vol. 19, no. 7, pp. 1243–1253, Jul. 2019.

[22] M. R. Oliveira and F. Louzada, "An evidence of link between default and loss of bank loans from the modeling of competing risks," *Singaporean J. Bus. Econ. Manage. Stud.*, vol. 3, no. 1, pp. 30–37, 2014.

[23] V. G. Cancho, A. K. Suzuki, G. D. C. Barriga, and F. Louzada, "A non-default fraction bivariate regression model for credit scoring: An application to Brazilian customer data," *Commun. Statistics: Case Stud., Data Anal. Appl.*, vol. 2, nos. 1–2, pp. 1–12, Apr. 2016.

[24] M. Ribeiro de Oliveira, F. Moreira, and F. Louzada, "The zero-inflated promotion cure rate model applied to financial data on time-to-default," *Cogent Econ. Finance*, vol. 5, no. 1, Jan. 2017, Art. no. 1395950.

[25] J. S. B. Toledo, V. L. D. Tomazella, C. M. M. Lima, and M. H. Felix, "Gompertz zero-inflated cure rate regression models applied to credit risk data," *Appl. Stochastic Models Bus. Ind.*, vol. 39, no. 2, pp. 177–197, Mar. 2023.

[26] C. Jiang, Z. Wang, and H. Zhao, "A prediction-driven mixture cure model and its application in credit scoring," *Eur. J. Oper. Res.*, vol. 277, no. 1, pp. 20–31, Aug. 2019.

[27] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger, "DeepSurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network," *BMC Med. Res. Methodol.*, vol. 18, no. 1, pp. 1–12, Dec. 2018.

[28] C. Lee, W. Zame, J. Yoon, and M. Van Der Schaar, "Deephit: A deep learning approach to survival analysis with competing risks," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–9.

[29] C. Lee, J. Yoon, and M. V. D. Schaar, "Dynamic-DeepHit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 1, pp. 122–133, Jan. 2020.

[30] S. Wiegrebe, P. Kopper, R. Sonabend, B. Bischl, and A. Bender, "Deep learning for survival analysis: A review," 2023, *arXiv:2305.14961*.

[31] F. Liu, Z. Hua, and A. Lim, "Identifying future defaulters: A hierarchical Bayesian method," *Eur. J. Oper. Res.*, vol. 241, no. 1, pp. 202–211, Feb. 2015.

[32] L. Dirick, G. Claeskens, and B. Baesens, "An Akaike information criterion for multiple event mixture cure models," *Eur. J. Oper. Res.*, vol. 241, no. 2, pp. 449–457, Mar. 2015.

[33] E. Wycinka and T. Jurkiewicz, "Mixture cure models in prediction of time to default: Comparison with Logit and Cox models," in *Contemporary Trends and Challenges in Finance*, K. Jajuga, L. T. Orlowski, and K. Staehr, Eds. Cham, Switzerland: Springer, 2017, pp. 221–231.

[34] L. Dirick, G. Claeskens, A. Vasnev, and B. Baesens, "A hierarchical mixture cure model with unobserved heterogeneity for credit risk," *Econometrics Statist.*, vol. 22, pp. 39–55, Apr. 2022.

[35] M.-H. Chen, J. G. Ibrahim, and D. Sinha, "A new Bayesian model for survival data with a surviving fraction," *J. Amer. Stat. Assoc.*, vol. 94, no. 447, p. 909, Sep. 1999.

[36] Y. Peng and B. Yu, *Cure Models: Methods, Applications, and Implementation*. New York, NY, USA: Chapman &Hall/CRC, 2021.

[37] H. Liu and Y. Shen, "A semiparametric regression cure model for interval-censored data," *J. Amer. Stat. Assoc.*, vol. 104, no. 487, pp. 1168–1178, Sep. 2009.

[38] C. M. Carvalho Lopes and H. Bolfarine, "Random effects in promotion time cure rate models," *Comput. Statist. Data Anal.*, vol. 56, no. 1, pp. 75–87, Jan. 2012.

[39] P. Lambert and V. Bremhorst, "Inclusion of time-varying covariates in cure survival models with an application in fertility studies," *J. Roy. Stat. Soc. Ser. A, Statist. Soc.*, vol. 183, no. 1, pp. 333–354, Jan. 2020.

[40] S. Kim, Y. Xi, and M.-H. Chen, "A new latent cure rate marker model for survival data," *Ann. Appl. Statist.*, vol. 3, no. 3, pp. 1124–1146, Sep. 2009.

[41] E. R. Brown and J. G. Ibrahim, "Bayesian approaches to joint cure-rate and longitudinal models with applications to cancer vaccine trials," *Biometrics*, vol. 59, no. 3, pp. 686–693, Sep. 2003.

[42] T. Chen and P. Du, "Promotion time cure rate model with nonparametric form of covariate effects," *Statist. Med.*, vol. 37, no. 10, pp. 1625–1635, May 2018.

[43] V. Bremhorst, M. Kreyenfeld, and P. Lambert, "Nonparametric double additive cure survival models: An application to the estimation of the non-linear effect of age at first parenthood on fertility progression," *Stat. Model.*, vol. 19, no. 3, pp. 248–275, Jun. 2019.

[44] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[45] X. Zhu, J. Yao, and J. Huang, "Deep convolutional neural network for survival analysis with pathological images," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, 2016, pp. 544–547.

[46] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," 2021, *arXiv:2106.11342*.

[47] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[48] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[49] J. Asano, A. Hirakawa, and C. Hamada, "Assessing the prediction accuracy of cure in the cox proportional hazards cure model: An application to breast cancer data," *Pharmaceutical Statist.*, vol. 13, no. 6, pp. 357–363, Nov. 2014.

[50] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly Weather Rev.*, vol. 78, no. 1, pp. 1–3, Jan. 1950.

[51] E. Graf, C. Schmoor, W. Sauerbrei, and M. Schumacher, "Assessment and comparison of prognostic classification schemes for survival data," *Stat. Med.*, vol. 18, nos. 17–18, pp. 2529–2545, 1999.

[52] M.-H. Chen, J. G. Ibrahim, and D. Sinha, "Bayesian inference for multivariate survival data with a cure fraction," *J. Multivariate Anal.*, vol. 80, no. 1, pp. 101–126, Jan. 2002.

[53] E. M. Carneiro, C. H. Q. Forster, L. F. S. Mialaret, L. A. V. Dias, and A. M. da Cunha, "High-cardinality categorical attributes and credit card fraud detection," *Mathematics*, vol. 10, no. 20, p. 3808, Oct. 2022.

[54] D. Micci-Barreca, "A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems," *ACM SIGKDD Explorations Newslett.*, vol. 3, no. 1, pp. 27–32, Jul. 2001.

[55] X. He, K. Zhao, and X. Chu, "AutoML: A survey of the state-of-the-art," *Knowledge-Based Syst.*, vol. 212, Jan. 2021, Art. no. 106622.

[56] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 1–25, 2012.

[57] Y. Peng and J. Xu, "An extended cure model and model selection," *Lifetime Data Anal.*, vol. 18, no. 2, pp. 215–233, Apr. 2012.

[58] D. R. Cox, "Regression models and life-tables," *J. R. Stat. Soc., B (Methodol.)*, vol. 34, no. 2, pp. 187–202, Jan. 1972.

[59] Z. Wang, J. Crook, and G. Andreeva, "Reducing estimation risk using a Bayesian posterior distribution approach: Application to stress testing mortgage loan default," *Eur. J. Oper. Res.*, vol. 287, no. 2, pp. 725–738, Dec. 2020.

[60] V. Medina-Olivares, R. Calabrese, J. Crook, and F. Lindgren, "Joint models for longitudinal and discrete survival data in credit scoring," *Eur. J. Oper. Res.*, vol. 307, no. 3, pp. 1457–1473, Jun. 2023.

[61] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, May 2019.