Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

CUTTING THROUGH THE NOISE: MACHINE LEARNING PROXIES FOR HIGH DIMENSIONAL NESTED SIMULATION

Xintong Li
Mingbin Feng
Tony S. Wirjanto

Department of Statistics and Actuarial Science
University of Waterloo
200 University Ave W
Waterloo, ON N2L 3G1, CANADA

ABSTRACT

Deep learning models have gained great success in many applications, but their adoption in financial and actuarial applications have been received by regulators with some trepidation. The lack of transparency and interpretability of these models leads to skepticism about their resilience and reliability, which are important factors to ensure financial stability and insurance benefit fulfillment. In this study, we use stochastic simulation as a data generator to examine deep learning models under controlled settings. Our study shows interesting findings in fundamental questions like "What do deep learning models learn from noisy data?" and "How well do they learn from noisy data?". Based on our findings, we propose an efficient nested simulation procedure that uses deep learning models as proxies to estimate tail risk measures of hedging errors for variable annuities. The proposed procedure uses deep learning models to concentrate simulation budget on tail scenarios while maintaining transparency in the estimation.

1 INTRODUCTION

Machine learning models, particularly deep learning models (Hastie et al. 2009; LeCun et al. 2015), have attracted attentions of researchers and practitioners due to the successes in solving real-world tasks such as AlphaGo (Silver et al. 2016) and ChatGPT (OpenAI 2023). Since the first artificial neural network model (McCulloch and Pitts 1943) and the first algorithm for training a perceptron (Rosenblatt 1958), especially after the introduction of backpropagation (Rumelhart et al. 1985) and the growth of high-performance computing, the field of artificial neural network and deep learning in general has grown rapidly. Two specialized neural network architectures that are relevant to our study are recurrent neural networks (RNNs) (Williams and Zipser 1989; Sutskever et al. 2014) and long-short-term memory (LSTM) (Hochreiter and Schmidhuber 1997; Chung et al. 2014), as we need to train proxy models that take sequential observations as input.

Besides designing specialized neural network architectures and training algorithms, one branch of neural network research considers the effects of noise in the input data and the error tolerance of neural networks. For example, Luo et al. (2016) showed that adding noise to the input of a convolutional neural network can increase the effective receptive field of the network and improve its ability to capture global features. Srivastava et al. (2014) quantified the error tolerance by injecting noise with a custom Boltzmann machine hardware. Yang et al. (2018) studied the error tolerance of a convolutional neural network (CNN) on input image captured under a low-voltage setting. The aforementioned studies use real-world data, as is typically the case for many deep learning studies, where noise is already present in the data. Users of real-world data have little control over the noise level and usually examine the effect of noisy data by

injecting noise. But it is unclear whether the deep learning model trained on noisy data actually learns the real, i.e., noiseless, feature-label relationship. Due to their lack of transparency and interpretability, the adoption of deep learning models in financial and actuarial applications has been received by regulators with some skepticism.

The contributions of our study are two-fold:

1. We propose a two-stage nested simulation procedure that uses deep learning model to improve its efficiency without losing transparency. In essence, a deep learning proxy model is used in the first stage to identify a set of potential tail scenarios on which computations are performed in the second stage. Our numerical results show that deep learning proxy models can identify the tail scenarios accurately and so the proposed procedure can estimate tail risk measures with similar accuracy while, at the same time, using less simulation budget.
2. We study what deep learning models learn from noisy data by training them using simulated data based on well-designed simulation experiments. This is a novel way to study the effect of noisy data and error tolerance of deep learning models as one can *reduce noise* in the data by increasing the number of replications in a simulation model. This new way of studying deep learning models can provide more direct evidence on their transparency and interpretability.

We are curious about fundamental questions like "What do deep learning models learn from noisy data?" and "How well do deep learning models learn from noisy data?". Data-driven answers to these questions prevail in the existing literature. For example, deep supervised learning models are believed to learn from the given data about the feature-label relationship to predict new labels for unseen features. Cross validation using to assess a subset, i.e., the validation set, of the original data, is a common way to access the quality of learning. Generalization error in the test set is another popular assessment metric. But the test set is also a subset of the original data. In this study, we revisit these questions and propose an alternative approach to answer them. Instead of relying solely on real-data (splitting it into multiple subsets), we propose using stochastic simulation outputs as training data for deep learning models. By controlling the simulation design parameters, such as the number of independent replications, we can control the quality (and also the quantity) of the data fed into the deep learning model. In such a controlled environment, we obtain more clear-cut answers to the above fundamental questions.

As an illustrative example, we consider an actuarial application that requires complex stochastic simulations: estimating tail risk measures of the hedging errors for variable annuities (VAs) (Boyle and Schwartz 1977; Boyle and Hardy 1997; Hardy 2003; Bauer et al. 2008). VAs (also known as Segregated Funds in Canada) are tax-deferred equity-linked insurance products that are commonly used for retirement benefits that can last for decades. By design, VAs can have significant financial market risk exposures. The recent survey study by Feng et al. (2022) states that "Variable annuity is arguably the most complex individual retirement planning product in the financial market." Hedging, dynamic hedging in particular, is a common way to manage this market risk. From an insurer's perspective, VAs can be modeled as embedded options. In a simplified setting, a dynamic hedging program sets up an initial hedge portfolio at the inception of the contract then rebalances it periodically to alleviate changes of the embedded option value due to fluctuations of the underlying risk factors. The initial costs to set up the hedge portfolio, the periodic hedging gains and losses due to rebalancing, the unwinding of the hedge at maturity, as well as the payment of guaranteed benefits, and the management fee income, are recognized as part of the profit and loss (P&L) of the VA contract. The present value of these cash flows is the insurer's overall loss and is the loss random variable to which we apply a suitable risk measure.

Nested estimation (Gordy and Juneja 2010) is usually required for estimating tail risk measures of the hedging errors for VAs: the outer simulation projects the underlying risk factors to quantify the loss distribution under the real-world measure. In many risk measurement applications, the outer simulation projects the risk factors to one future time step (Bauer et al. 2012; Feng et al. 2016; Lin and Yang 2020b, for example). Similar to the dynamic hedging problems considered in Feng and Jing (2017), Lin and Yang (2020a), Dang et al. (2020), Dang et al. (2023), in this study we consider multi-period outer projections,

e.g., 20 or 40 years, and the time steps are based on the assumed hedge rebalancing frequency, e.g., monthly. The outer simulated paths are known as the *scenarios*, and may include simulated asset returns, guarantee value, interest rates, policyholder behaviour, and longevity experience, etc. Conditional on an outer scenario, the inner simulation estimates the risk-neutral price (and Greeks) of the guaranteed liability, i.e., the embedded options of VA contracts, at all rebalancing dates and then estimates the present value of the hedging error in that scenario. Despite modeling flexibility, standard nested simulation is notorious for its heavy computational burden. Proxy modeling is a popular way to alleviate this drawback.

Borrowing terminologies from machine learning research, we can view a set of simulated outer scenarios and the estimated hedging errors for those scenarios as the *features* and (noisy) *labels*. One can train supervised learning models using these simulated features and labels then are used to replace the time-consuming inner simulations by the trained models. We refer to the trained supervised learning models as *proxy models*. There are many proxy models proposed in the nested simulation literature: parametric regression (also known as least-squares Monte Carlo) (Longstaff and Schwartz 2001; Broadie et al. 2015; Ha and Bauer 2015), kernel smoothing (Hong et al. 2017), Gaussian process modeling (also known as Kriging) (Liu and Staum 2010; Gan 2013), and the likelihood ratio method (Dang et al. 2023; Zhang et al. 2022), just to name a few. These proxies have been shown, theoretically or numerically, to work well when the scenarios, i.e., the features, have relatively low dimensions. For complex nested simulation model and high-dimensional scenarios, more powerful proxy models are needed. Inspired by Dang et al. (2020), Dang et al. (2022), Dang et al. (2023), in this study we propose a two-stage nested simulation procedure to estimate tail risk measures for the hedging errors of VAs. Stage I uses a deep learning proxy model to identify a set of proxy tail scenarios. Stage II performs computations only on the proxy scenarios to save simulation budget.

The rest of this article is organized as follows: Section 2 presents the problem settings for tail risk measures and dynamic hedging of VAs. Section 3 proposes an efficient two-stage nested simulation procedure that uses machine learning models as proxies to help reduce simulation budget by only performing computations on identified tail scenarios. Section 4 demonstrates the efficiency of deep learning proxies and examines error tolerance of two LSTM proxy models with different numbers of trainable parameters.

2 TAIL RISK ESTIMATION FOR DYNAMIC HEDGING ERRORS OF VARIABLE ANNUITIES

In this section we present notations, problem settings, and a simulation model for tail risk estimation for hedging errors of variable annuities. For readers who are interested in the examination of a machine learning proxy model, it is sufficient to understand that our simulation model generates data with 240 features and 1 real-value label.

2.1 Tail Risk Measures: VaR and CVaR

Measuring and monitoring risks, particularly tail risks, are important risk management tasks for financial institutions like banks and insurance companies. Two most popular tail risk measures are Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) (Rockafellar and Uryasev 2002). Other names of CVaR include Conditional Tail Expectation (CTE), Tail Value-at-Risk (TailVaR), and Expected Shortfall (ES).

Consider a loss random variable L whose losses and gains lie in the right and left tails, respectively, of its distribution. For a given confidence level $\alpha \in [0,1]$, the α-VaR is defined as the α-quantile of L: $\text{VaR}_\alpha = q_\alpha = \inf\{q : \Pr(L \le q) \ge \alpha\}$. The α-CVaR of L is defined as $\text{CVaR}_\alpha = \frac{1}{1-\alpha}\int_{v=\alpha}^{1} q_v dv$. Tail risk measures like VaR and CVaR are widely used for setting regulatory and economic capital, which is the amount of capital a financial institution holds to cover its risk. For example, European insurers set regulatory capital at 99.5%-VaR according to Solvency II (EIOPA 2014). In Canada, the regulatory capital requirement for VAs is set based on CVaRs as prescribed in OSFI (2017).

Let $L_1, L_2, \ldots, L_M$ be M independent and identically distributed (i.i.d.) simulated losses of L and let $L_{(1)} \le L_{(2)} \le \ldots \le L_{(M)}$ be the corresponding ordered losses. For a given confidence level α (assume that

αM is an integer for simplicity), α-VaR can be estimated by the sample quantile $\widehat{\text{VaR}}_\alpha = L_{(\alpha M)}$. Also, α-CVaR can be estimated by

$$\widehat{\text{CVaR}}_\alpha = \frac{1}{(1-\alpha)M} \sum_{i=\alpha M+1}^{M} L_{(i)} = \frac{1}{(1-\alpha)M} \sum_{i \in \mathscr{T}_{(1-\alpha)M}} L_i,$$

where we define a *true tail scenario set* of size k as $\mathscr{T}_k = \{i : L_i > L_{(M-k)}\}$. In this study, the loss random variable of interest is the hedging error for VA (Section 2.3).

2.2 Simulation Model for Variable Annuity Payouts

Variable annuity contracts offer different types of guarantees. Generally speaking, a portion of the VA premium is invested in a sub-account whose return is linked to some stock indices.

Two relevant types of guarantees in our studies are:

- **Guaranteed Minimum Maturity Benefit (GMMB)**: A GMMB contract pays a maturity benefit equal to the greater of the sub-account value and a fixed guarantee value. The guarantee value is often set as a percentage, e.g., 75% or 100%, of the initial premium.
- **Guaranteed Minimum Withdrawal Benefit (GMWB)**: A GMWB contract guarantees the minimum amount of periodic withdrawal the policyholder can take from the sub-account until maturity, even if the sub-account value reduces to zero. The minimum withdrawal benefit is typically a fixed percentage of the guarantee value. The guarantee value will decrease if the withdrawal exceeds the guaranteed minimum. The GMWB is typically offered with an accumulation period, during which no withdrawals are made but a GMDB is usually offered. Additional features offered with the GMWB include roll-up, ratchet, and reset (The Geneva Association 2013).

For a comprehensive review of other types of VA contracts such as Guaranteed Minimum Death Benefit (GMDB), Guaranteed Minimum Accumulation Benefit (GMAB) and Guaranteed Lifetime Withdrawal Benefit (GLWB), we refer readers to Hardy (2003).

Consider a generic VA contract with maturity $T > 0$ periods, e.g., $T = 240$ months. Let S_t, F_t, and G_t be the indexed stock price, the subaccount value and the guarantee value, respectively, at time $t = 1, 2, \ldots, T$. Evolution of the subaccount value and the guarantee value of a VA contract affect the contract payout. Note that the policyholder's (random) time of death also affects the timing of the benefit payout for certain types of VA such as GMDB, but this is not considered in our study for simplicity. For clarity, we use F_t and F_{t_+} to denote the sub-account value just before and just after the withdrawal at time t, if any. Let η_g be the gross rate of management fee that is deducted from the fund value at each period and let $\eta_n < \eta_g$ be the net rate of management fee income to the insurer. The difference between the gross management fee and the net management fee income represents the incurred investment expenses.

At the inception of the contract, i.e., $t = 0$, we assume that the whole premium is invested in the stock index and the guarantee base is set to the sub-account value: that is, $S_0 = F_0 = G_0$. At each time $t = 1, \ldots, T$, the following events take place in the following order:

1. The sub-account value changes according to the growth of the underlying stock and the (gross) management fee is deducted. That is, $F_t = F_{(t-1)_+} \cdot \frac{S_t}{S_{t-1}} \cdot (1 - \eta_g)$, where $(x)^+ = \max\{x, 0\}$ and $F_{(t-1)_+}$ will be defined later. The insurer's income at time t is the net management fee, i.e., $F_t \eta_n$.
2. The guarantee value ratchets up (ratcheting is a common feature in GMWB) if the sub-account value exceeds the previous guarantee value, i.e., $G_t = \max\{G_{t-1}, F_t\}$.
3. The withdrawal is made (for GMWB) and is deducted from the sub-account value, i.e., $F_{t_+} = (F_t - I_t)^+$, where $I_t = \gamma G_t$. A GMMB can be modeled with $\gamma = 0$.

We see from the above modeling steps that the status of a generic VA contract is summarized by a triplet (S_t, F_t, G_t) whose evolution is driven by the stochasticity of S_t. In practice, the simulation model may also incorporate additional complications like mortality, lapse, and excess withdrawal, etc.

At any time $t = 1, \ldots, T$, the insurer's liability in a VA contract is the present value of all payments, net of the fee income. For example, suppose that the per-period risk-free rate is r, then the insurer's time-t liability for a GMMB contract is $V_t = e^{-r(T-t)} \cdot (G_T - F_T)^+ - \sum_{s=t+1}^{T} e^{-r(T-s)} F_s \eta_n$. Also, the insurer's time-t liability for a GMWB contract is $V_t = \sum_{s=t+1}^{T} e^{-r(T-s)} [(I_s - F_s)^+ - \eta_n F_s]$.

For example, consider the time-t liability V_t of a GMWB: Suppose that given the stock sample path, e.g., an outer path $S_1, \ldots, S_t$, one can simulate future stock prices $\widetilde{S}_{t+1}, \ldots, \widetilde{S}_T$, e.g., inner sample paths, based on some asset model such as a Black-Scholes model. The tilde symbol ($\sim$) over a quantity denotes its association with the inner simulation. Given the time t state (S_t, F_t, G_t), following Cathcart et al. (2015) the sensitivity of V_t with respect to S_t can be estimated by a pathwise estimator (Glasserman 2004)

$$\Delta_t(\widetilde{S}_{t+1}, \ldots, \widetilde{S}_T | S_t) = \frac{\partial V_t}{\partial S_t} = \sum_{s=t+1}^{T} e^{-r(T-s)} \left[\mathbf{1}\{\widetilde{I}_s > \widetilde{F}_s\} \cdot \left(\frac{\partial \widetilde{I}_s}{\partial S_t} - \frac{\partial \widetilde{F}_s}{\partial S_t} \right) - \eta_n \frac{\partial \widetilde{F}_s}{\partial S_t} \right], \quad t = 0, \ldots, T-1, \tag{1}$$

where $\mathbf{1}\{\cdot\}$ is an indicator function and

$$\begin{aligned}
\frac{\partial \widetilde{F}_s}{\partial S_t} &= \mathbf{1}\{\widetilde{I}_{s-1} < \widetilde{F}_{s-1}\} \cdot \left(\frac{\partial \widetilde{F}_{s-1}}{\partial S_t} - \frac{\partial \widetilde{I}_{s-1}}{\partial S_t} \right) \cdot \frac{\widetilde{S}_s}{\widetilde{S}_{s-1}} \cdot (1 - \eta_g), \\
\frac{\partial \widetilde{G}_s}{\partial S_t} &= \mathbf{1}\{\widetilde{G}_{s-1} < \widetilde{F}_s\} \cdot \frac{\partial \widetilde{F}_s}{\partial S_t} + \mathbf{1}\{\widetilde{G}_{s-1} \geq \widetilde{F}_s\} \cdot \frac{\partial \widetilde{G}_{s-1}}{\partial S_t}, \\
\frac{\partial \widetilde{I}_s}{\partial S_t} &= \gamma \frac{\partial \widetilde{G}_s}{\partial S_t}.
\end{aligned}$$

The recursion is initialized with $(\widetilde{S}_t, \widetilde{F}_t, \widetilde{G}_t) = (S_t, F_t, G_t)$, $\frac{\partial \widetilde{F}_s}{\partial S_t} = \frac{\widetilde{F}_t}{S_t}$, and $\frac{\partial \widetilde{G}_s}{\partial S_t} = \frac{\partial \widetilde{I}_s}{\partial S_t} = 0$.

2.3 Dynamic Hedging for Variable Annuities

Below we provide a scheme used to perform a multi-period nested simulation in estimating (profit and loss) P&L for one outer scenario.

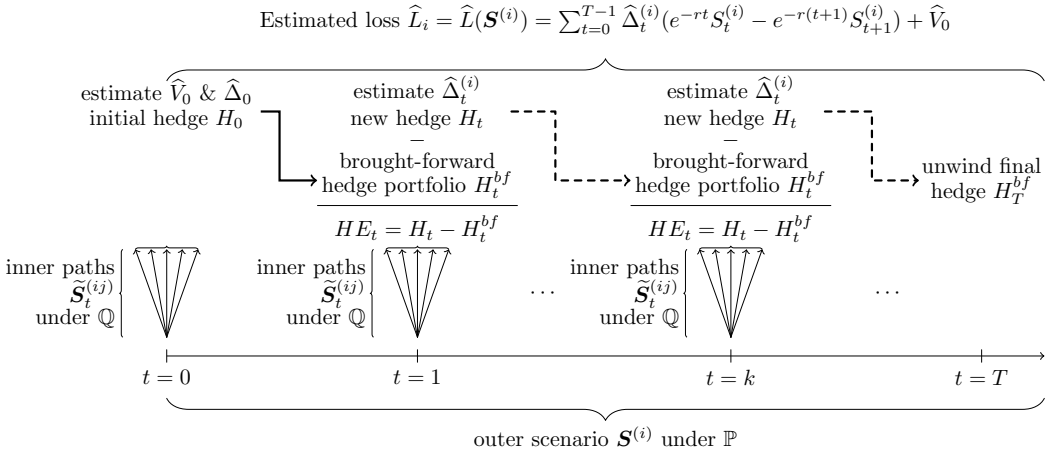


Figure 1: Illustration of multi-period nested simulation that estimates the P&L for one outer scenario.

Insurers commonly use dynamic hedging to mitigate a market risk exposure in VA contract's embedded options. In a dynamic hedging program, a hedge portfolio is set up and periodically rebalanced for a portfolio of VA contracts using stocks, bonds, futures, and other derivatives. For simplicity, in this study we consider delta hedging for a generic VA liability using one stock and one bond. The machine learning proxy algorithm in Section 3 can be trivially adapted to more general hedging strategies.

Consider a generic VA contract whose delta hedge portfolio at any time t, $t = 0, 1, \ldots, T-1$, consists of Δ_t units in the underlying stock and B_t amount of a risk-free zero-coupon bond maturing at time T. The value hedge portfolio at time time-$(t-1)$ is $H_{t-1} = \Delta_{t-1} S_{t-1} + B_{t-1}$, where S_t is the underlying stock price and any time $t > 0$. This hedge portfolio is brought forward to the next rebalancing time t, when its value becomes $H_t^{bf} = \Delta_{t-1} S_t + B_{t-1} e^r$. Therefore, the time t hedging error, i.e., the cash flow incurred by the insurer due to rebalancing at time t, is

$$HE_t = H_t - H_t^{bf}, \quad t = 1, \ldots, T-1. \tag{2}$$

The P&L of the VA contract includes the cost of the initial hedge (H_0), the hedging errors (2), the unwinding of the hedge at maturity (H_T^{bf}), and the unhedged liability (V_0). Mathematically, the present value of these cash flows is given by

$$L = H_0 + \sum_{t=1}^{T-1} e^{-rt} HE_t - e^{-rT} H_T^{bf} + V_0 = \sum_{t=0}^{T-1} \Delta_t (e^{-rt} S_t - e^{-r(t+1)} S_{t+1}) + V_0, \tag{3}$$

where the second equality holds by a telescopic sum simplification of $e^{-rt} B_t$, $t = 0, \ldots, T-1$.

In (3), Δ_t and V_0 are determined by using a risk-neutral measure $\mathbb{Q}$ while the distribution of L is under a real-world measure $\mathbb{P}$. If Δ_t and V_0 cannot be calculated analytically, a nested simulation is required to estimate the tail risk measure of L. Recall from Section 2.2 that the stock sample path, regardless of the inner or outer simulation or a combination of both, determines the evolution of the triplet (S_t, F_t, G_t). For example, for GMWB, a standard nested simulation procedure to estimate the α-CVaR of L is as follows:

1. For $i = 1, \ldots, M$, simulate outer scenarios $\boldsymbol{S}^{(i)} = (S_1^{(i)}, \ldots, S_T^{(i)})$ under the real-world measure $\mathbb{P}$. Figure 1 illustrates the inner simulation experiments needed for an outer scenario.
2. For $t = 0$, simulate time-0 inner paths $\widetilde{\boldsymbol{S}}_0^{(j)} = (\widetilde{S}_1^{(j)}, \widetilde{S}_2^{(j)}, \ldots, \widetilde{S}_T^{(j)})$, $j = 1, \ldots, N$ under $\mathbb{Q}$ and then estimate V_0 by $\widehat{V}_0 = \sum_{s=1}^{T} e^{-r(T-s)} [(I_s - F_s)^+ - \eta_n F_s]$ and estimate $\widehat{\Delta}_0 = \Delta_0(\widetilde{S}_1^{(j)}, \ldots, \widetilde{S}_T^{(j)} | S_0)$ based on (1). Note that S_0 is known, so these inner paths do not depend on any outer scenario. The same $\widehat{V}_0$ and $\widehat{\Delta}_0$ are used in all scenarios.
3. Given each scenario $\boldsymbol{S}^{(i)}$, the following inner simulation is needed to estimate the corresponding loss $\widehat{L}_i$ for that scenario.
   (a) At each time $t = 1, \ldots, T-1$, simulate inner paths $\widetilde{\boldsymbol{S}}_t^{(ij)} = (\widetilde{S}_{t+1}^{(ij)}, \ldots, \widetilde{S}_T^{(ij)})$, $j = 1, \ldots, N$ under $\mathbb{Q}$ and then estimate Δ_t by $\widehat{\Delta}_t^{(i)} = \Delta_t(\widetilde{S}_{t+1}^{(ij)}, \ldots, \widetilde{S}_T^{(ij)} | \boldsymbol{S}_t^{(i)})$ based on (1).
   (b) Use scenarios $\boldsymbol{S}^{(i)}$ and $\widehat{V}_0$ and $\widehat{\Delta}_t^{(i)}$ to calculate losses $\widehat{L}_i^{MC}$, $t = 0, \ldots, T-1$, based on (3). Then sort them as $\widehat{L}_{(1)}^{MC} \le \widehat{L}_{(2)}^{MC} \le \cdots \le \widehat{L}_{(M)}^{MC}$.
4. Estimate α-CVaR of L by $\widehat{\text{CVaR}}_\alpha^{MC} = \frac{1}{(1-\alpha)M} \sum_{i=\alpha M+1}^{M} \widehat{L}_{(i)}^{MC} = \frac{1}{(1-\alpha)M} \sum_{i \in \widehat{\mathscr{T}}_{(1-\alpha)M}^{MC}} \widehat{L}_i^{MC}$ where $\widehat{\mathscr{T}}_k^{MC}$ denotes a *Monte Carlo tail scenario set* associated with the largest k estimated losses.

We refer to the collection of experiments needed conditional on one scenario $\boldsymbol{S}^{(i)}$ to estimate L_i, that is, all upward arrows in Figure 1, as one inner simulation experiment. We make three observations: (1) each inner simulation is time-consuming, as it includes T simulation experiments, one at each time $t = 0, \ldots, T-1$, (2) after running inner simulations for M scenarios, we obtain simulated data, that is, feature-label pairs, $(\boldsymbol{S}^{(i)}, \widehat{L}_i)$, $i = 1, \ldots, M$; the feature vector $\boldsymbol{S}$ is T dimensional, and (3) most importantly, when estimating tail risk measures such as α-CVaR, only a small number of estimated losses, that is, those associated with the set of tail scenarios $\widehat{\mathscr{T}}_k$ are used in the estimator.

3 MACHINE LEARNING PROXY MODELS FOR NESTED SIMULATION

Based on the three observations above and inspired by Dang et al. (2020), we propose a two-stage nested simulation procedure which uses machine learning proxies to identify potential tail scenarios. We present

our proposed procedure as a competitor to the standard procedure with M outer scenarios and N inner replications for each outer scenario, as described in Section 2.3. We propose a two-stage procedure with a machine learning proxy that aims to produce a CVaR estimate that's as accurate as that of the standard procedure, but uses less computations as than latter. An overview of the proposed procedure is as follows:

I. **Train machine learning proxy using simulation data.**
   - Use a fraction of the total budget to run Steps 1, 2, and 3 in the standard procedure with the same number of outer scenarios, M, but a much smaller number of inner replications, i.e., $N' \ll N$, in each scenario. Then obtain M simulated data points, i.e., feature-label pairs, $(\boldsymbol{S}^{(i)}, \widehat{L}_i)$, $i = 1, \ldots, M$.
   - Use the simulated data, $(\boldsymbol{S}^{(i)}, \widehat{L}_i)$, $i = 1, \ldots, M$ to train a machine learning model. We refer to the trained model as a *proxy model* as denote it by $\widehat{L}^{PX}(\boldsymbol{S})$. Denote the proxy losses for the outer scenarios by $\widehat{L}_i^{PX} = \widehat{L}^{PX}(\boldsymbol{S}^{(i)})$, $i = 1, \ldots, M$.
   - Sort the proxy losses $\widehat{L}_{(1)}^{PX} \leq \widehat{L}_{(2)}^{PX} \leq \cdots \leq \widehat{L}_{(M)}^{PX}$ to identify a *proxy tail scenario set* associated with the largest proxy losses, i.e., $\widehat{\mathscr{T}}_m^{PX} := \{i : \widehat{L}_i^{PX} > \widehat{L}_{(M-m)}^{PX}\}$. The number of proxy tail scenarios, m, is a user's choice and will be discussed later.

II. **Concentrate simulation on proxy tail scenarios.**
   - Run Steps 2 and 3 of the standard procedure with the same number of inner replications, N, but only on the proxy tail scenarios, i.e., scenarios associated with $\widehat{\mathscr{T}}_m^{PX}$. Denote a estimated losses and sorted losses by $\widehat{L}_i^{ML}$ and $\widehat{L}_{(i)}^{ML}$, respectively, $i = 1, \ldots, m$.
   - Estimate α-CVaR of L by $\widehat{\text{CVaR}}_\alpha^{ML} = \frac{1}{(1-\alpha)M} \sum_{i=\alpha M+1}^{M} \widehat{L}_{(i)}^{ML} = \frac{1}{(1-\alpha)M} \sum_{i \in \widehat{\mathscr{T}}_{(1-\alpha)M}^{ML}} \widehat{L}_i^{ML}$ where $\widehat{\mathscr{T}}_k^{ML}$ denotes a *machine learning tail scenario set* associated with the largest k estimated losses.

Similar to Dang et al. (2020), the proposed two-stage procedure uses the proxy model to identify the proxy tail scenario set in Stage I. But, different from their fixed-budget simulation design, we attempt to achieve a target accuracy. Specifically, in Stage II we propose using the same number of inner replications, N, as the competing standard procedure. We believe that the goal of designing an efficient simulation procedure is to solve practical problems faster, so a target-accuracy design is more suitable, which refers to obtaining a similar level of accuracy as the standard procedure but with much less simulation budget.

The size of the proxy tail scenario set in Stage I, m, is an important experiment design parameter that affects the correct identification of true tail scenarios and ultimately affects the estimation accuracy for CVaR. Clearly, there is a lower bound $m \geq (1-\alpha)M$ because the α-CVaR is estimated by the average of $(1-\alpha)M$ largest losses at the end of Stage II. For ease of reference, we call the additional percentage of proxy tail scenarios above this lower bound, i.e., $\varepsilon = \frac{m-(1-\alpha)M}{M}$, as a *safety margin*. On one hand, large ε is not desirable because it increases computations in Stage II. On the other hand, ε should be set reasonably large so more true tail scenarios are included in the the proxy tail scenario set $\widehat{\mathscr{T}}_m^{PX}$ and are ultimately included in $\widehat{\mathscr{T}}_{(1-\alpha)M}$ at the end of Stage II. The selection of m is highly dependent on the choice of the proxy model. In the numerical experiments, we examine the relationship between the safety margin and the correct identification of true tail scenarios for different proxy models.

Lastly, we note that other studies, such as Liu and Staum (2010), Broadie et al. (2015), Hong et al. (2017), and Zhang et al. (2022) use stochastic kriging, regression, kernel smoothing, and the likelihood ratio method, respectively, as proxy models in nested simulation procedures. Our proposal has two key distinctions over the existing ones: (1) our proxy model's input, or feature, has high dimension. For example, for a 20-year VA contract whose hedge is rebalanced monthly, the feature vector is 240 dimension, which is at least one order of magnitude larger than the input dimension in the aforementioned studies, and (2) our proxy model is only used for tail scenario identification but is *not* used in the estimation of the tail risk measures at the end of Stage II. This is a feature designed particularly to convince regulators that the losses used in estimating the risk measure are based on a transparent inner simulation model rather than on some black-box proxy models.

4 NUMERICAL EXPERIMENTS

We conduct a series of simulation experiments to (1) demonstrate the efficiency of the proposed two-stage procedure and (2) examine the error tolerance to noisy training data in deep learning models. The problem settings in our experiments are identical to those in Dang et al. (2020): we consider estimating the 95% CVaR of the hedging loss of a GMWB contract, which is one of the most complex VA contracts in the market. The VA contracts have a 20-year maturity and are delta-hedged with monthly rebalancing, i.e., $T = 240$ rebalancing periods. The gross and net management fees are $\eta_g = 0.2\%$ and $\eta_n = 0.1\%$, respectively. The risk-free rate is 0.2% per period and the underlying asset S_t is modeled by a regime-switching geometric Brownian motion with parameters specified in Table 2 of Dang et al. (2020).

To compare the numerical performances of different simulation procedures, we create a benchmark dataset with a large-scale nested simulation: We first simulate $M = 50{,}000$ outer scenarios, i.e., 240-periods stock paths $\boldsymbol{S}^{(1)}, \ldots, \boldsymbol{S}^{(M)}$ under $\mathbb{P}$ and used these outer scenarios in all further experiments. Note that the 5% tail scenario set includes 2,500 scenarios. As the hedging loss for these scenarios cannot be calculated analytically, we run inner simulations with a large number of replications, $N = 10{,}000$, conditional on each of the M scenarios. We denote these losses by $L_1, \ldots, L_M$ and will refer to them as *true losses*. We also use these true losses to estimate $\widehat{\text{CVaR}}_{95\%}$ and denote the corresponding *true tail scenario set* by $\mathscr{T}_{2500}$. Lastly, we refer to the set of feature-label pairs $\{(\boldsymbol{S}^{(i)}, L_i) : i = 1, \ldots, M\}$ as a *true dataset*. Note that the feature vector $\boldsymbol{S}$ is a 240-dimension stock path.

We compare our two-stage procedure to a standard nested simulation procedure that runs $N = 1{,}000$ inner replications for each of the $M = 50{,}000$ outer scenarios. In Stage I of the proposed procedure, we first running inner simulations with $N' = 100$ inner replications for each of the $M = 50{,}000$ outer scenarios. So, Stage I's simulation budget is 10% of the standard procedure's. The resulting feature-label pairs $\{(\boldsymbol{S}^{(i)}, \widehat{L}_i) : i = 1, \ldots, M\}$ is used for training different proxy models. Specifically, following the convention in machine learning research, we split this dataset into three parts: The training, validation, and test sets have 42,300, 4,700, and 3,000 data points (84.6%, 9.4%, and 6% of the dataset), respectively. At the end of Stage I, m proxy tail scenarios, are identified. In Stage II, $N = 1{,}000$ inner replications are run for all proxy tail scenarios. Stage II's simulation budget is $\frac{m}{M}$ of the standard procedure's. In short, the two-stage procedure uses $15\% - 30\%$ of the standard procedure's budget for safety margin between $0\% - 15\%$.

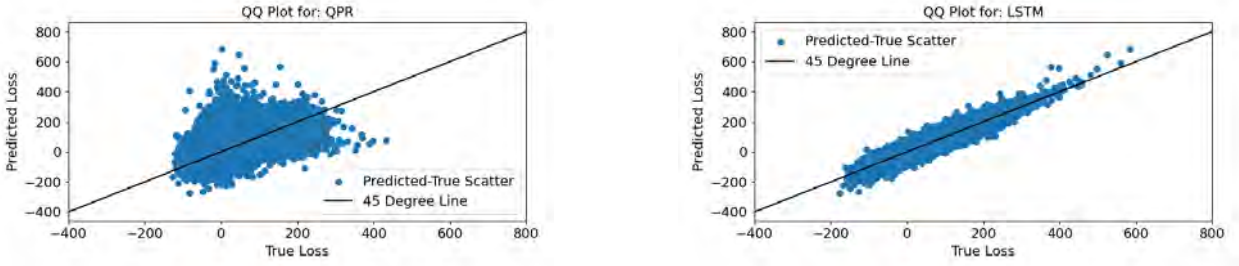
Five proxy models are considered in the experiment: multiple linear regression (MLR), quadratic polynomial regression (QPR) without interaction terms, feed-forward neural network (FNN), recurrent neural network (RNN), and long short-term memory (LSTM) network. MLR and QPR are considered as extensions of regression proxy models in the nested simulation literature. FNN is a generic deep learning model while RNN and LSTM are specialized models to accommodate the sequential structure of time-series. A tanh activation function is used for RNN and LSTM layers, and a ReLU activation is used for the fully-connected layers. All neural network proxies are trained by the Adam optimizer (Kingma and Ba 2014) with an initial learning rate of 0.001 and a exponential learning rate decay schedule. FNN is trained with a dropout rate of 20%. RNN and LSTM are trained with a dropout rate of 10%. The architectures and training settings are typical choices in the deep learning literature. The training labels are normalized to have zero mean and unit standard deviation. The architectures and the numbers of trainable parameters are shown in the first two columns of Table 1. We see that the three deep-learning proxy models' have orders of magnitudes more trainable parameters than the two regression models. In machine-learning terminologies, the three deep-learning proxy models have much higher *capacities*.

The last three columns in Table 1 display the average squared errors between the proxy losses and the losses (labels) in different datasets. We first observe that the two regression proxy models' errors are larger than those of three deep-learning models. This is because the regression models' capacities are too low to learn the complex dynamic hedging simulation model. There are also differences among the three deep-learning models. The FNN is a generic model, its test error is larger than the training error, which is a sign of over-fitting and poor generalization. In contrast, RNN and LSTM are networks designed to capture

Table 1: Architectures and MSEs of proxy models for GMWB dynamic hedging losses.

Model	**Layer size**	**# parameters**	**Training errors**	**Test errors**	**True errors**
MLR	N/A	241	0.7075	0.6885	0.6932
QPR	N/A	481	0.5415	0.5411	0.5294
FNN	$240 \times 128 \times 16 \times 1$	35,009	0.2119	0.3053	0.2089
RNN	$(240,32),(240,4),32$	32,021	0.1264	0.1702	0.1198
LSTM	$(240,32),(240,4),32$	35,729	0.0902	0.0958	0.0789

temporal relationship in the feature. As a result, they have lower training errors, i.e., they fit the data better, and have lower testing error, i.e., they generalize better. Notably, the true errors for the deep-learning proxy models are lower than the test errors. Recall that all three data sets are generated from the same simulation model, but the true data set is less noisy than the training and test data. Part of the test error is due to simulation noise, and this noise is smaller in the true error. We observe that the two regression proxy models not only generalize to the test data poorly but also generalize to the true data poorly. In contrast, the deep-learning proxy models generalizes better to the true data than to the test data. This implies that these models indeed learn the true feature-label relationship in the dynamic hedging simulation model (i.e., true data) even though they are trained on noisy observations (e.g., training data) of the model.



(a) QPR proxy model

(b) LSTM proxy model

Figure 2: QQ-plots between proxy losses (x-axis) and true losses (y-axis) for the same outer scenarios.

Figure 2 shows the quantile-quantile (QQ) plots between the true losses and the proxy losses. We see that the LSTM proxy models' QQ-plots closely follow that 45-degree line. Again, these proxy models are trained on noisy data, so the good fit to the true data should not be taken for granted. Figure 2 and Table 1 both show that well-chosen deep learning models can *cut through the noise* in the training data and learn the true underlying feature-label relationship.

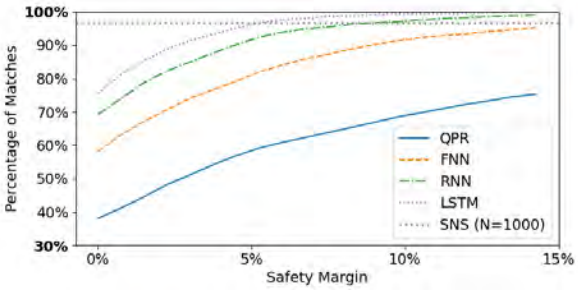
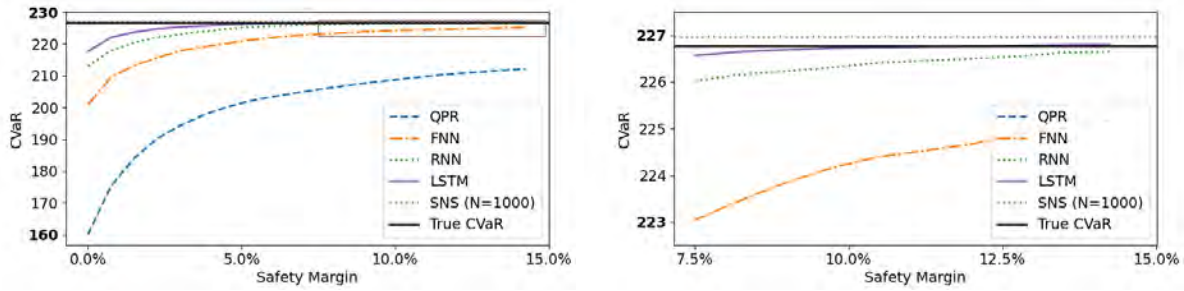


Figure 3: Percentage of correctly identified true tail scenarios by different proxy models.

Next we examine the proxy models' abilities to correctly identify true tail scenarios, which is crucial for the performance of the proposed two-stage procedure. Figure 3 depicts the percentage of correctly identified true tail scenarios by different proxy models for different safety margins. We see that a poor proxy model like the QPR identifies less than 40% of the true tail scenario without any safety margin. In contrast, a good proxy model like the LSTM identifies more than 75% of the true tail scenarios without any safety margin and more than the QPR proxy does with 15% of safety margin. For comparison, the horizontal line shows the true tail identification percentage for the standard nested simulation procedure. The LSTM proxy reaches similar percentage with 5% of safety margin, which corresponds to 5 times less simulation budget than the standard procedure's.



(a) Safety margin 0% - 15%.

(b) Safety margin 7.5% - 15%.

Figure 4: 95%-CVaR estimates by different procedures. Right figure is a zoomed-in version of left figure.

Lastly, we return to our original goal of estimating the 95%-CVaR of the dynamic hedging error. Figure 4 shows the 95%-CVaR estimates for different proxy models with different safety margins. Figure 4b is a zoomed version of Figure 4a. Because the safety margin only affects the two-stage procedure, the true 95%-CVaR and the one estimated by the standard procedure are horizontal (solid and dotted, respectively) lines in Figure 4. We see that, with reasonable safety margins, the two-stage procedures with RNN and LSTM proxy models produce estimates that are closer to the true value than the standard procedure's estimate. The LSTM proxy model is particularly superior as it accurately identifies the true tail scenarios and produces highly accurate estimates with small safety margins.

5 CONCLUSION

The proposed two-stage nested simulation procedure with a machine learning proxy model results in substantial computational savings in estimating CVaR of the hedging loss of a VA contract from accurately identifying the tail scenarios. An LSTM fueled by noisy observation pairs cuts through the noise and learns the true inner simulation model. When new outer scenarios are generated, a trained machine learning proxy model can distinguish between tail and non-tail scenarios without the need to run new inner simulations.

REFERENCES

Bauer, D., A. Kling, and J. Russ. 2008. "A Universal Pricing Framework for Guaranteed Minimum Benefits in Variable Annuities". *ASTIN Bulletin: The Journal of the IAA* 38(2):621–651.

Bauer, D., A. Reuss, and D. Singer. 2012. "On the Calculation of the Solvency Capital Requirement Based on Nested Simulations". *ASTIN Bulletin: The Journal of the IAA* 42(2):453–499.

Boyle, P. P., and M. R. Hardy. 1997. "Reserving for Maturity Guarantees: Two Approaches". *Insurance: Mathematics and Economics* 21(2):113–127.

Boyle, P. P., and E. S. Schwartz. 1977. "Equilibrium Prices of Guarantees under Equity-Linked Contracts". *Journal of Risk and Insurance* 44(4):639–660.

Broadie, M., Y. Du, and C. C. Moallemi. 2015. "Risk Estimation via Regression". *Operations Research* 63(5):1077–1097.

Cathcart, M. J., H. Y. Lok, A. J. McNeil, and S. Morrison. 2015. "Calculating Variable Annuity Liability "Greeks" Using Monte Carlo Simulation". *ASTIN Bulletin: The Journal of the IAA* 45(2):239–266.

Chung, J., C. Gulcehre, K. Cho, and Y. Bengio. 2014. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". https://arxiv.org/abs/1412.3555. Accessed 7th Sep 2023.

Dang, O., M. Feng, and M. R. Hardy. 2020. "Efficient Nested Simulation for Conditional Tail Expectation of Variable Annuities". *North American Actuarial Journal* 24(2):187–210.

Dang, O., M. Feng, and M. R. Hardy. 2022. "Dynamic Importance Allocated Nested Simulation for Variable Annuity Risk Measurement". *Annals of Actuarial Science* 16(2):319–348.

Dang, O., M. Feng, and M. R. Hardy. 2023. "Two-Stage Nested Simulation of Tail Risk Measurement: A Likelihood Ratio Approach". *Insurance: Mathematics and Economics* 108:1–24.

EIOPA 2014. "The Underlying Assumptions in the Standard Formula for the Solvency Capital Requirement Calculation". Technical report, The European Insurance and Occupational Pensions Authority.

Feng, R., Z. Cui, and P. Li. 2016. "Nested Stochastic Modeling for Insurance Companies". Technical report, Society of Actuaries.

Feng, R., G. Gan, and N. Zhang. 2022. "Variable Annuity Pricing, Valuation, and Risk Management: A Survey". *Scandinavian Actuarial Journal* 2022(10):867–900.

Feng, R., and X. Jing. 2017. "Analytical Valuation and Hedging of Variable Annuity Guaranteed Lifetime Withdrawal Benefits". *Insurance: Mathematics and Economics* 72:36–48.

Gan, G. 2013. "Application of Data Clustering and Machine Learning in Variable Annuity Valuation". *Insurance: Mathematics and Economics* 53(3):795–801.

Glasserman, P. 2004. *Monte Carlo Methods in Financial Engineering*, Volume 53. Springer.

Gordy, M. B., and S. Juneja. 2010. "Nested Simulation in Portfolio Risk Measurement". *Management Science* 56(10):1833–1848.

Ha, H., and D. Bauer. 2015. "A Least-Squares Monte Carlo Approach to the Calculation of Capital Requirements". In *World Risk and Insurance Economics Congress*, Volume 6, 2–6. https://danielbaueracademic.files.wordpress.com/2013/11/bauerha_2015.pdf. Accessed 7th Sep 2023.

Hardy, M. R. 2003. *Investment Guarantees: Modeling and Risk Management for Equity-Linked Life Insurance*, Volume 168. John Wiley & Sons.

Hastie, T., R. Tibshirani, and J. H. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Volume 2. Springer.

Hochreiter, S., and J. Schmidhuber. 1997. "Long Short-Term Memory". *Neural Computation* 9(8):1735–1780.

Hong, J. L., S. Juneja, and G. Liu. 2017. "Kernel Smoothing for Nested Estimation with Application to Portfolio Risk Measurement". *Operations Research* 65(3):657–673.

Kingma, D. P., and J. Ba. 2014. "Adam: A Method for Stochastic Optimization". *arXiv preprint arXiv:1412.6980*. https://arxiv.org/pdf/1412.6980.pdf. Accessed 7th Sep 2023.

LeCun, Y., Y. Bengio, and G. E. Hinton. 2015. "Deep Learning". *Nature* 521(7553):436–444.

Lin, X. S., and S. Yang. 2020a. "Efficient Dynamic Hedging for Large Variable Annuity Portfolios With Multiple Underlying Assets". *ASTIN Bulletin: The Journal of the IAA* 50(3):913–957.

Lin, X. S., and S. Yang. 2020b. "Fast and Efficient Nested Simulation for Large Variable Annuity Portfolios: A Surrogate Modeling Approach". *Insurance: Mathematics and Economics* 91:85–103.

Liu, M., and J. Staum. 2010. "Stochastic Kriging for Efficient Nested Simulation of Expected Shortfall". *Journal of Risk* 12(3):3.

Longstaff, F. A., and E. S. Schwartz. 2001. "Valuing American Options by Simulation: A Simple Least-Squares Approach". *The Review of Financial Studies* 14(1):113–147.

Luo, W., Y. Li, R. Urtasun, and R. Zemel. 2016. "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks". *Advances in Neural Information Processing Systems* 29.

McCulloch, W. S., and W. Pitts. 1943. "A Logical Calculus of the Ideas Immanent in Nervous Activity". *The Bulletin of Mathematical Biophysics* 5:115–133.

OpenAI 2023. "ChatGPT". https://chat.openai.com/chat. Accessed 7th Sep 2023.

OSFI 2017. "Life Insurance Capital Adequacy Test". Technical report, Office of the Superintendent of Financial Institutes Canada.

Rockafellar, R. T., and S. Uryasev. 2002. "Conditional Value-At-Risk for General Loss Distributions". *Journal of Banking & Finance* 26(7):1443–1471.

Rosenblatt, F. 1958. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain". *Psychological Review* 65(6):386.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1985. "Learning Internal Representations by Error Propagation". Technical report, California University San Diego La Jolla Institute for Cognitive Science.

Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot et al. 2016. "Mastering the Game of Go with Deep Neural Networks and Tree Search". *Nature* 529(7587):484–489.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". *The Journal of Machine Learning Research* 15(1):1929–1958.

Sutskever, I., O. Vinyals, and Q. V. Le. 2014. "Sequence to Sequence Learning with Neural Networks". In *Advances in Neural Information Processing Systems*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Volume 27: Curran Associates, Inc.

The Geneva Association 2013. "Variable Annuities—An Analysis of Financial Stability". Technical report. https://www.genevaassociation.org/sites/default/files/research-topics-document-type/pdf_public/ga2013-variable_annuities_0.pdf. Accessed 7th Sep 2023.

Williams, R. J., and D. Zipser. 1989. "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks". *Neural Computation* 1(2):270–280.

Yang, L., D. Bankman, B. Moons, M. Verhelst, and B. Murmann. 2018. "Bit Error Tolerance of a CIFAR-10 Binarized Convolutional Neural Network Processor". In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–5. Institute of Electrical and Electronics Engineers.

Zhang, K., M. B. Feng, G. Liu, and S. Wang. 2022. "Sample Recycling for Nested Simulation with Application in Portfolio Risk Measurement". https://arxiv.org/abs/2203.15929. Accessed 2nd May 2023.

AUTHOR BIOGRAPHIES

XINTONG LI is a PhD student in the Department of Statistics and Actuarial Science at University of Waterloo. His research interests include quantitative risk management, Monte Carlo simulation design and analysis, and statistical machine learning. His email address is xintong.li1@uwaterloo.ca.

MINGBIN FENG is an Assistant Professor in the Department of Statistics and Actuarial Science at University of Waterloo. His research interests include quantitative risk management, financial engineering, Monte Carlo simulation design and analysis, and nonlinear optimization. He is particularly interested in the intersection of these fields such as statistical machine learning, portfolio optimization, efficient simulation algorithms for risk management, etc. His email address is ben.feng@uwaterloo.ca, and his website is https://www.math.uwaterloo.ca/~mbfeng/.

TONY S. WIRJANTO is a Professor in the Department of Statistics and Actuarial Science, School of Accounting and Finance, and David R. Cheriton School of Computer Science at University of Waterloo. His research interests lie in the intersection between statistics and econometrics. In particular, he conducts research in the field of financial time series with a focus on volatility modeling/forecasting and financial risk management, and in the field of financial mathematics with a focus on portfolio optimization in a high-dimensional setting and on global climate change risks. His email address is twirjanto@uwaterloo.ca, and his website is https://www.math.uwaterloo.ca/~twirjant/index.html.