

Predictive Analytics and Machine Learning – Practical Applications for Actuarial Modeling (Nested Stochastic)

May | 2023





Predictive Analytics and Machine Learning

Practical Applications for Actuarial Modeling (Nested Stochastic)

AUTHORS Jean-Philippe Larochelle, FSA, MAAA, CERA

Peter Carlson, FSA, MAAA

Vincent Carrier Cote, FSA, MAAA

Ying Lu, FSA, MAAA

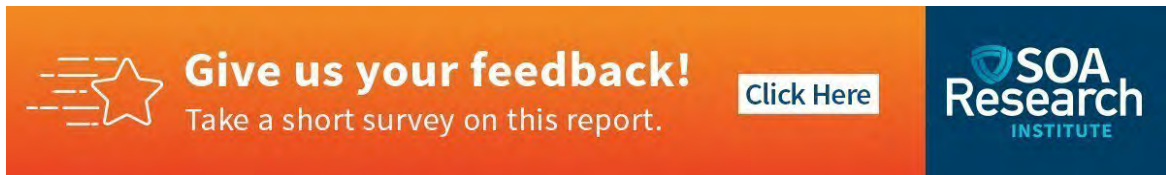
Noah Shapiro, FSA

Alex Tam, ASA, MAAA

Viresh Thusu

Ally Zhang, FSA, MAAA

SPONSOR Actuarial Innovation and Technology
Strategic Research Program Steering
Committee

A horizontal banner with an orange-to-red gradient background. On the left is a white star icon with horizontal lines extending from its left side. To the right of the star is the text "Give us your feedback!" in a large, bold, white sans-serif font. Below this is the text "Take a short survey on this report." in a smaller white font. To the right of the text is a white rectangular button with the text "Click Here" in a blue font. On the far right of the banner is the SOA Research Institute logo, which consists of a blue shield icon and the text "SOA Research INSTITUTE" in white.

Caveat and Disclaimer

The opinions expressed and conclusions reached by the authors are their own and do not represent any official position or opinion of the Society of Actuaries Research Institute, the Society of Actuaries or its members. The Society of Actuaries Research Institute makes no representation or warranty to the accuracy of the information.

Copyright © 2023 by the Society of Actuaries Research Institute. All rights reserved.

CONTENTS

Executive Summary	6
Background.....	6
Objectives and Content	7
Key Takeaways and Words of Caution.....	8
Section 1: Predictive Analytics, Artificial Intelligence, and Machine Learning Overview	9
1.1. History and Recent Developments	9
1.1.1 Preliminary Developments and “AI Winter”	9
1.1.2 Recent Developments.....	10
1.1.3 Looking Forward.....	10
1.2 Fundamentals of AIML	11
1.2.1 Key Definitions.....	11
1.2.2 AIML Learning Approaches.....	12
1.2.3 Neural Networks	13
1.3 Key Phases of Building AIML Models	14
1.3.1 Business Problem Definition.....	14
1.3.2 Data Discovery, Acquisition and Generation	15
1.3.3 Data Transformation	15
1.3.4 Model Selection	15
1.3.5 Model Training	15
1.3.6 Overfitting.....	15
1.3.7 Train Test Split.....	16
1.3.8 Model Testing.....	16
1.3.9 Learning Curve	17
1.3.10 Hyperparameter Tuning	17
1.3.11 Auto ML	17
1.4 Model Development Cycle and Management	18
1.4.1 ML OPS Overview	18
1.4.2 Model Build	19
1.4.3 Model Packaging and Registering	19
1.4.4 Model Deploy	19
1.4.5 Application Testing.....	19
1.4.6 Production Release	19
1.4.7 Model Monitor	20
1.4.8 Model Analyzing	20
1.4.9 Model Governance.....	20
1.4.10 Model Retraining	21
Section 2: Literature Review	22
2.1 Relevant Research from the SOA and Other Actuarial Organizations	22
2.1.1 A Tour of AI Technologies in Time Series Prediction.....	22
2.1.2 Emerging Data Analytics Techniques with Actuarial Applications.....	24
2.1.3 Literature Review: Artificial Intelligence and Its Use in Actuarial Work.....	25
2.1.4 Cloud Computing and Machine Learning Uses in the Actuarial Profession.....	26
2.1.5 Considerations for Predictive Modeling in Insurance Applications.....	26
2.1.6 Nested Stochastic Modeling for Insurance Companies	27
2.2 Academic Papers Exploring AIML For Actuarial Modeling	29
2.2.1 Efficient Dynamic Hedging for Large Variable Annuity Portfolios with Multiple Underlying Assets	29
2.2.2 Efficient Nested Simulation for Conditional Tail Expectation of Variable Annuities	32

2.2.3 Efficient Simulation Designs for Valuation of Large Variable Annuity Portfolios	33
2.2.4 Fast and Efficient Nested Simulation for Large Variable Annuity Portfolios: A Surrogate Modeling Approach	35
2.2.5 Nested Simulation in Portfolio Risk Measurement	37
2.2.6 Efficient Risk Estimation via Nested Sequential Simulation	39
2.2.7 Risk Estimation via Regression	40
2.2.8 A Comparative Study of Risk Measures for Guaranteed Minimum Maturity Benefits by a PDE Method	41
2.2.9 Valuation of Large Variable Annuity Portfolios under Nested Simulation: A Functional Data Approach	42
Section 3: Practical Application of AIML for Actuarial Modeling	44
3.1 AIML Actuarial Modeling Applications	44
3.1.1 Clustering	44
3.1.2 Proxy Models	46
3.1.3 Scenario Selection	48
3.1.4 Hybrids	48
3.2 Approach Overview	49
3.2.1 Problem Statement, Business Case and Planning	49
3.2.2 Environment Preparation	50
3.2.3 Data Generation	50
3.2.4 Feature Engineering and Selection	51
3.2.5 Model Testing and Selection	51
3.2.6 Hyperparameter Tuning	51
3.2.7 Actuarial Evaluation	51
Section 4: Professional, Fairness and Regulatory Considerations	52
4.1. ASOP Review	52
4.2. Bias and Fairness	52
4.3. Regulatory Developments	53
Section 5: Case Study 1 – Index Crediting	55
5.1. Introduction	55
5.2. Product and Actuarial Methodology Specifications	58
5.3. AIML Model Development	59
5.3.1. Preparation	60
5.3.2. Data Generation	60
5.3.3. Feature Engineering and Feature Selection	64
5.3.4. Model Testing and Selection	68
5.3.5. Actuarial Evaluation	76
5.4. Business Case	79
5.5. Conclusions	82
Section 6: Case Study 2 - Variable Annuity Fair Value	83
6.1. Introduction	83
6.1.1 Background and Product Feature Overview	83
6.2 Product and Actuarial Methodology Specifications	85
6.3 AIML Model Development	88
6.3.1. Preparation	90
6.3.2. Data Generation	90
6.3.3 Feature Engineering, Feature Selection and Output Definition	95
6.3.4 Model Testing and Selection	101
6.3.5 Hyperparameter Tuning	105
6.3.6 Actuarial Evaluation	107
6.4 Conclusions	113
Section 7: Acknowledgments	114

- Appendix A: Variable Annuity Product Specifications..... 115**
 - A1: Base Product Features 115
 - A2: Guaranteed Minimum Death Benefit..... 116
 - A3: Guaranteed Lifetime Withdrawal Benefit 117

- Appendix B: Variable Annuity Actuarial Assumptions 119**
 - B1: Premiums 119
 - B2: Mortality 119
 - B3: Lapse 119
 - B4: Partial Withdrawal Behavior 120
 - B5: Maintenance Expenses 121
 - B6: Reinvestment Strategy 121

- Appendix C: ASOP Review..... 123**

- References..... 129**

- About The Society of Actuaries Research Institute 132**

Executive Summary

BACKGROUND

Actuaries in the life and annuity space have a well-established bottoms-up approach to modeling cash flows and calculating actuarial reserves and related balances for financial reporting.

Actuaries have traditionally modeled long-duration life and annuity products at the policy or cell level before aggregating. This reflects:

- Policy cash flows and related items as needed (e.g., premiums, policy charges/fees, policy benefits, commissions, expenses, reserves, account value, and amount in-force).
- Actuarial assumptions (e.g., mortality, morbidity, lapse, and partial withdrawal).
- Economic assumptions (e.g., interest rates, equity returns, inflation).

This approach can be very cumbersome and runtime intensive with large blocks of business (which may include hundreds of thousands or millions of policies). The most common runtime challenge faced by actuaries is when “inner-loop” stochastic calculations are performed to calculate reserves or other metrics such as fair value at selected time steps in an “outer-loop” projection scenario. The outer-loop scenario(s) could be (i) a single deterministic scenario, (ii) multiple deterministic scenarios, or (iii) a stochastic simulation.

Actuaries refer to this situation as “nested stochastic.” Various research has been produced on nested stochastic applications and challenges, including by the SOA (Feng, et al., 2016).

Common occurrences of nested stochastic include the following:

1. **Financial reporting:** When a financial reporting valuation or projection is based on a metric requiring stochastic valuation such as fair value or Principles-Based Reserves (PBR) for Variable Annuities (VA).
2. **Pricing:** Pricing life and annuity insurance products using distributable earnings (with required capital) generally requires projecting the total asset requirement over the life of the product. Depending on the jurisdiction, this total asset requirement may be based on stochastic calculations. Further, actuaries commonly use stochastic projections along which the total asset requirement needs to be calculated to price market-sensitive products.
3. **Financial planning and analysis:** Life insurance companies forecast their balance sheet and income statement to understand the future anticipated financial performance of their business. This generally requires insurers to project reserves, capital, asset allocation, hedging, and other relevant actuarial balances, some of which will require a nested stochastic calculation. This process is generally manual and requires back and forth between finance and actuarial.
4. **Risk management:** Additionally, many life insurers will perform extensive analysis to evaluate the solvency of the insurance company. This analysis requires the projection of the total asset requirement along with assets and hedge derivatives to evaluate the likelihood of insolvency.

Further, while nested stochastic calculations are the most common computing challenge of insurance companies, there are other situations such as point-in-time stochastic models where the bottoms-up actuarial modeling can be onerous.

The runtime challenge facing actuaries has been exacerbated by the development of increasingly complex insurance products and changes in reporting frameworks (such as IFRS17 (IFRS, 2023), LDTI (FASB, 2016),

PBR (AAA, 2016) and Solvency II (EIOPA, 2016), which are increasing the reliance on complex stochastic calculations.

Actuaries can increase computing resources (e.g., cloud computing) to address these requirements. However, the cost can often be prohibitive, and it may be complex to fully utilize the computing resources available (actuaries need to employ sophisticated distribution strategies).

This may leave actuaries to use approximative methodologies to help mitigate runtime limitations, including, for example, simplifying calculations or scenarios, using proxy models and clustering. While these techniques may improve runtime, it will generally come at the cost of reduced precision (sometimes significantly reduced). Further, there are many cases where actuaries cannot run a given use case without the use of approximations, which complicates or makes it impossible to quantify the impact of the reduction in precision. Lastly, these simplifications are sometimes not well understood and may be difficult to maintain within desired tolerances if they rely on core simplifying assumptions that may not be valid over time.

Given these challenges, it is natural for actuaries to continue seeking to innovate their modeling approaches to overcome the runtime challenges with first principles, bottoms-up modeling for long-duration life and annuity products.

OBJECTIVES AND CONTENT

Paper objectives

The objective of this paper is to provide a practical guide with concrete case studies to help actuaries implement artificial intelligence and machine learning (AIML) to help accelerate the speed of analysis for life and annuity actuarial modeling. We do so by researching the existing literature regarding AIML and its applicability to stochastic modeling and exploring case studies.

We do not position AIML models as a replacement to traditional actuarial models, but rather as a strategic capability that can dramatically reduce runtimes and computing costs associated with complex stochastic or nested stochastic applications.

In particular, this paper aims to provide a balance of theoretical knowledge along with practical concrete case studies that tackle some of the most prominent nested stochastic runtime challenges.

Content overview

This paper provides the following:

- Overview and fundamental knowledge of AIML with resources for actuaries looking to become more proficient in this field (see section [1. Predictive analytics, artificial intelligence, and machine learning overview](#)).
- Review of existing literature on this topic (see section [2. Literature review](#)).
- Overview of the applications of AIML in the actuarial modeling space (see section [3. Practical application of AIML for actuarial modeling](#)).
- Review of regulatory and other considerations when applying the AIML methodologies described in this paper (see section [4. Professional, fairness, and regulatory considerations](#)).
- Various case studies to provide practical examples for a wide range of actuarial applications representative of common nested stochastic runtime issues:
 - We explored a relatively more straightforward case study to provide a gentle

introduction to practical applications for the reader. This first case explores the application of AIML to proxy the Monte Carlo or risk-neutral valuation of exotic index crediting strategies, such as those that can be found on indexed products like Fixed Indexed Annuities (FIA), Registered Indexed Linked Annuities (RILA) and Indexed Universal Life (IUL). This more straightforward application allowed us to provide both the data generation and model calibration code for the user (see section [5. Case study 1 – Index crediting](#)).

- o We then moved on to a more complex case study where we proxied the fair value of various typical variable annuity guarantees. This is a very common example of where actuaries face runtime challenges due to nested stochastic (see section [6. Case study 2 – Variable annuity fair value](#)).

Note that we will be using the term AIML in this paper to collectively refer to predictive analytics, machine learning (ML), and artificial intelligence (AI). We will avoid getting bogged down in technicalities between these areas.

AI is generally defined as developing computers and programs that mimic human intelligence. ML is considered to be a subset of AI and explores algorithms that can learn insights from data or experience (Columbia, n.d.). In this paper, we adopt commonly accepted terms such as artificial intelligence, machine learning, and neural network. (Efron, et al., 1993).

KEY TAKEAWAYS AND WORDS OF CAUTION

This paper demonstrates through case studies that AIML can be a powerful tool for actuaries looking to address runtime challenges associated with complex stochastic and nested stochastic models for life and annuity products.

The AIML techniques discussed in this paper can be put to use to help address the growing analytical demands to meet business needs and the increased complexity in actuarial calculations introduced through the recent accounting changes happening across the globe.

However, we advise readers of this paper to apply critical thinking in applying the AIML techniques. Significant actuarial and analytical judgment is required to employ the appropriate controls and guardrails when using AIML for proxy modeling or other techniques that address runtime concerns.

Actuaries using these AIML techniques may report results to management or make actuarial decisions based on the output of AIML models instead of the traditional actuarial models needed to be confident in the reliability of the AIML model. Actuaries need to understand the limit of these AIML models (e.g., understand the input range where the model will and will not perform) and need to periodically validate the AIML models to ensure that they remain reliable proxies to the original actuarial calculation. Subsection [4.1 – ASOP Review](#) goes into more detail about some of the limitations to consider.

Section 1: Predictive Analytics, Artificial Intelligence, and Machine Learning Overview

Before exploring specific methodologies and case studies to apply AIML to actuarial applications, we first provide an overview of the key principles of predictive analytics and data science.

Depending on the reader's experience in this field, this section can be taken either as a gentle introduction to predictive analytics and data science or as a quick refresher.

Further, readers who are already proficient in artificial intelligence and machine learning may skip this section and move on to [Section 3](#).

However, we expect that most readers come from an actuarial background and are curious about the AIML capabilities introduced in this paper with limited to no prior AIML experience. Those readers will benefit from carefully reading through this section first to gain fundamental knowledge that will prove useful in approaching [Section 3](#) and the case studies.

Further, we recommend that actuaries looking to implement AIML capabilities for nested stochastic seek additional reference materials to build up their knowledge of the core principles of AIML and/or work with data scientists trained in this field. One example is the book *Deep Learning* (Goodfellow, et al, n.d.) by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.

Given the interest in this field, there are many resources available online – including a certificate from the SOA (SOA, 2023) – in the form of formal online classes, blogs, articles, academic papers, and others. Many of these resources also provide code and references to learn how to apply these concepts.

Lastly, we also caution that the paradigm behind some principles underpinning AIML methodologies may differ for actuarial applications. We will cover these nuances in [3. Practical Application of AIML for Actuarial Modeling](#).

1.1. HISTORY AND RECENT DEVELOPMENTS

In this section, we provide an overview of key events that led to the development of the latest AIML capabilities. This background is useful to understand how AIML has evolved and the key events that eventually led to the breakthroughs, widespread adoption, and innovations powered by AIML that we see today.

1.1.1 PRELIMINARY DEVELOPMENTS AND “AI WINTER”

The invention and exploration of the possibilities of AIML dates as far back as the 1940s and 1950s by researchers like Alan Turing and Marvin Minsky.

The breakthrough of the first artificial neural network using perceptrons sparked waves of excitement and research interest in AI as a formal discipline. For instance, the U.S. Defense Advanced Research Projects Agency (DARPA) supported much of the early development of machine intelligence and those projects subsequently transpired to the broader communities.

A Perceptron is an algorithm for supervised learning of binary classifiers. This algorithm enables neurons to learn and processes elements in the training set one at a time.

However, the lack of computational power (hardware) and high-quality data at the time, limited the applicability of early-day AI and led to many high-profile initiatives being abandoned after failure to deliver

results.

The interest and funding died down in the late 1970s and brought the onset of what is commonly referred to as the “AI Winter” where little attention was paid to AIML.

1.1.2 RECENT DEVELOPMENTS

The reasons for the early limitations or failings of AIML were circumvented by recent advances in computing power, as well as the significant growth in the volume and quality of data available as a whole. Computing power and the volume and quality of data have grown exponentially since the 1940s and 1950s.

Two decades ago, computers may have had about ten gigabytes of memory. Today, it’s not uncommon for big data technology companies to take in over half a billion terabytes of data daily. Big data today contains greater variety, arriving in increasing volumes and with more velocity. These are known as the “three Vs” – variety, volume, and velocity.

Three more Vs of data have emerged over the past few years, value, veracity, and variability. First, big data has become capital; a large part of the value the world’s leading tech companies offer comes from their data. They are constantly analyzing and producing new products using big data. Secondly, the degree to which big data can be trusted has increased a lot. Lastly, how the data can be used and formatted has developed dramatically as well. Expertly curated data, parallel computing, and smaller and faster chips are among a few of the contributors that turned the unviable algorithms of the past into the commercially achievable AI applications of today. These applications are an integral part of today’s social fabric. With the explosion of research breakthroughs enabled by technology backbones, intelligent systems such as autonomous driving, natural language processing, and generative design systems are no longer dreams of the distant past, but commonplace utilities facilitating aspects of everyday lives.

Below are some notable developments in AIML applications in recent years:

- 2011 Digital Assistants, Apple launches Siri
- 2015 Driverless Cars: Tesla announces Autopilot and Waymo takes to the streets
- 2017 Facebook Face AI: Facebook rolls out facial recognition to help with tagging
- 2018 Just walk out: Amazon releases Just Walk Out tech to streamline retail
- 2021 Delivery bots: Amazon begins testing delivery robot “Scout” in select cities
- 2023 Generative AI: OpenAI ChatGPT, Google Bard showcases generative Natural Language Processing AI agents

In the actuarial space, there are also advancements ranging from ML-driven assumption settings to adopting deep learning for nested stochastic approximations. These are revolutionizing the traditional actuarial processes with ever-increasing efficiencies and insights.

1.1.3 LOOKING FORWARD

AI advancement will be accelerated with a continuation in the overall improvement of hardware. The launch of specialized AI chips, such as Tensor Processing Unit (TPU) and Neural Processing Unit (NPU) customized for machine learning tasks, are new developments that hold much excitement.

In the meantime, increasingly large models are being developed for real world applications, with a seismic shift towards more fair, explainable and ethical AI. The SOA has multiple publications (Smith, et al., 2022) that go in depth on those topics and the growing concerns around privacy and fairness. Government, regulators, and professionals are also paying more attention to this topic. [Section 4](#) of the paper provides a

scan of the latest developments.

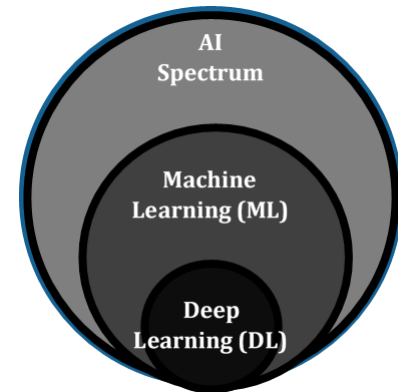
Furthermore, the increased research continues to expand the use cases and accessibility to broader audiences and expanded use cases, such as healthcare, education, and entertainment. AI will gradually become more commonplace in our daily lives, including in the finance and actuarial industries.

1.2 FUNDAMENTALS OF AIML

This section provides fundamental technical knowledge of AIML.

First, while there are multiple definitions of AI, this field can be considered as the overarching domain exploring the simulation of intelligence by machines. To achieve this, multiple components may be used, including natural language processing, text-to-speech, vision capabilities, robotics and decision-making.

Machine learning can be seen as a subcomponent that enables many of the AI components listed above. Machine learning would cover the range of models and techniques used to calibrate these models to the data or situation at hand. Learning methods include supervised learning, unsupervised learning and reinforcement learning (more on that below).



Deep learning is a subset of machine learning and is focused on the various types of neural networks (a type of machine learning model that is particularly promising).

Categories of traditional predictive analytics techniques have achieved various progress in the past for runtime reduction. The summary of those techniques is provided under the literature review section of this paper. **This paper will focus on the latest development, machine learning and, in particular, deep learning (Raden, et al., 2019) that drove the major AI advancement for solving highly complex tasks in this paper.** Those are the fields that will enable us to cut down on stochastic model runtime with minimal loss of precision or accuracy.

It is also worth mentioning predictive analytics, which bears similarity in techniques and models to machine learning, especially for some supervised learning applications. The main distinction with predictive analytics is its focus – which is on predicting future outcomes from experience.

As mentioned previously, we will use the term AIML in this paper for simplicity to avoid clouding the paper in distinguishing among AI, ML, and predictive analytics. We will use these techniques holistically to solve nested stochastic runtime challenges.

1.2.1 KEY DEFINITIONS

While definitions vary in the field of data science, the following are used for this paper:

1. **Dataset:** a collection of data used for the development and training of the model(s). This is further split into:
 - a. **Training set:** Data used to train or calibrate the AIML model.
 - b. **Validation set:** Data retained to control the AIML model against overfitting.
 - c. **Testing set:** Data that is used to evaluate the performance of the model. This data is held back throughout the development of the AIML model.

2. **AIML algorithm:** AIML algorithm is the underlying model form regardless of calibration. For instance, AIML algorithms explored in this paper include algorithms such as neural networks, random forests, etc.
3. **AIML model:** AIML model is a model that has been calibrated for a given purpose.
4. **Parameters:** In AIML, a model is defined or represented by the model parameters, such as its weights and biases for each node.
5. **Hyperparameters:** Hyperparameters are parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up learning. The prefix 'hyper' suggests that they are 'top-level' parameters that control the learning process and the model parameters that result from it.
6. **Training:** Hyperparameters are set before training begins and the learning algorithm uses them to learn the parameters. Behind the training scene, parameters are continuously being updated and the final ones at the end of the training constitute a model.

1.2.2 AIML LEARNING APPROACHES

Depending on the use case, fundamentally different approaches can be used to calibrate the AIML model. These approaches include supervised learning, unsupervised learning, and reinforcement learning.

The approach that should be used will generally depend on the task at hand and how the problem is framed for the machine to learn.

Supervised learning

With supervised learning, the AIML model is trained on labeled data to predict unseen future outcomes, detecting the underlying patterns and relationships between input with respect to output labels. The model is trained by minimizing errors of the predictions against the target output over the training data, enabling it to yield accurate results for classification and regression problems.

The purpose of regression models is to predict a numerical value given a set of input features. In a supervised learning setting, the machine learning model is provided with a set of input features and calibrated to predict the associated numerical output features.

The purpose of classification models is to predict a category given a set of input features (e.g., prediction is an image, a cat or a dog, a yes or no, etc.). In a supervised learning setting, the machine learning model is provided a set of input features and calibrated to predict the associated categorical output feature.

Unsupervised learning

With unsupervised learning, the AIML model uses unlabeled data to extract patterns and relationships inherent in the data. The algorithms, in absence of external labels, can learn about the data and capture unexpected findings. Clustering and network associations are examples of unsupervised learning.

Unsupervised models work on their own to discover patterns and information from the data provided. Distance-based clustering, such as nearest neighbor, follows a centroid model where clusters are defined by proximity to the mean vector. Density-based clustering, such as DBSCAN, defines clusters by connected dense region in the data space. These are often used in anomaly detection, lexical ambiguity, and recommender systems.

Dimensionality reductions transform the data from a high-dimensional space project into a low-dimensional space. This is commonly used in combination with other machine learning models to improve

the model. High-dimensional data can be computationally intractable and resource intensive. Principal component analysis is most prevalently used.

Reinforcement learning

Reinforcement learning is very different from supervised and unsupervised learning in the sense that it does not rely on historical data. Instead, reinforcement learning is established based on a system framework where an agent acts in an environment based on observations and receives the corresponding rewards to optimize strategy to accomplish a task.

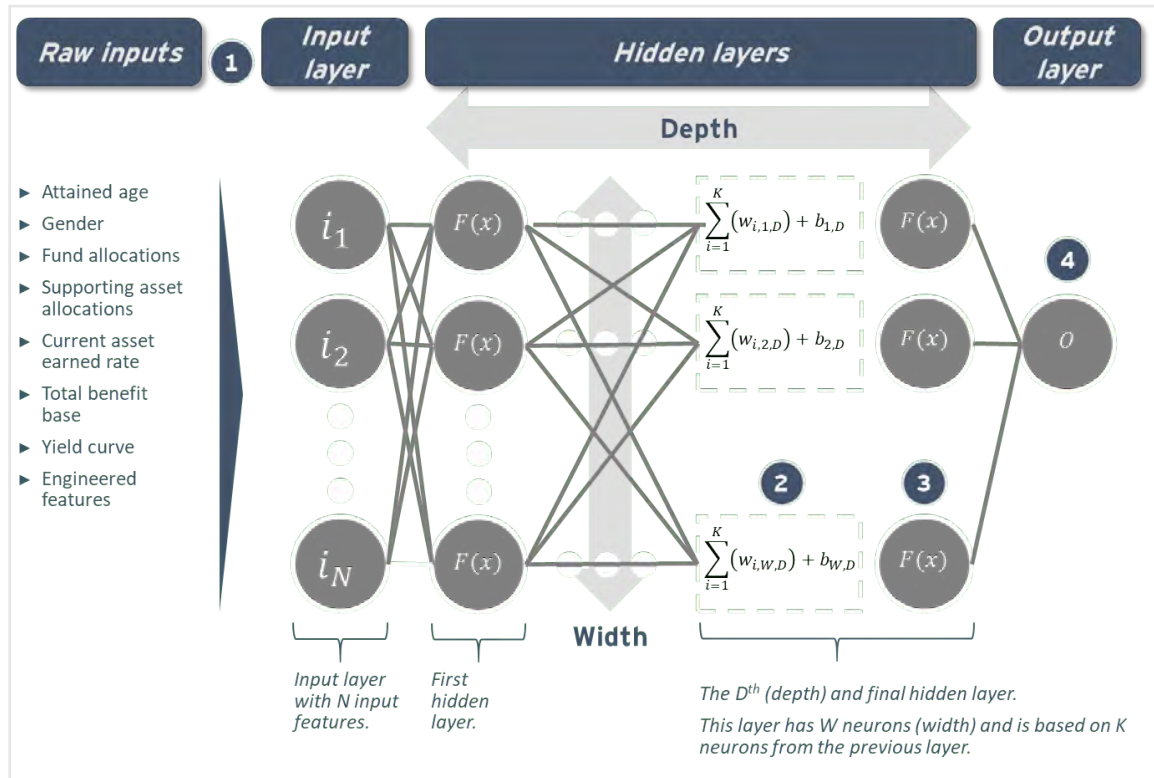
1.2.3 NEURAL NETWORKS

An artificial neuron is a digital construct that seeks to simulate the behavior of a biological neuron in the brain. Artificial neurons are typically used to make up an artificial neural network that are modeled after human brain activities. Depending on the network structure, it can be used to solve a variety of tasks. For example, the temporal sequence can be captured through Recurrent Neural Networks (RNN) and vision tasks can be solved through convolution neural networks (CNN).

In the nested stochastic use case, an artificial neural network outperforms other traditional approaches through its capacity to capture high dimensional, non-linear, complex state space. These are achieved through the design and specialization of a network structure.

A simplified illustration of how it can be applied with actuarial inputs and use case is below; this presents a general practical construct for actuarial use cases.

Figure 1
ILLUSTRATION OF NEURAL NETWORK



The four major components associated with the illustrations are:

1. **Feature scaling:** Neural networks require all input values to be within a similar range. Therefore, the first step is to scale the original features. A common approach is the MinMaxScaler where the features are scaled linearly between the minimum and maximum values.
2. **Weight previous layers:** Process information from the previous layer by applying the calibrated weights and adjusting with a calibrated bias for each neuron.
3. **Apply the activation function:** to the output of step 2. The output of the activation function will be used as an output to the next layer.
4. **Weight final layer:** Apply weights to the output of the final layer (D). This weighted output is used as the final output of the neural network. The network can produce one or more outputs.

1.3 KEY PHASES OF BUILDING AIML MODELS

1.3.1 BUSINESS PROBLEM DEFINITION

The first phase for successful machine learning model development is business understanding and planning. A clear business requirement is the key to project success. This phase generally involves business subject matter experts, actuaries, and data scientists who work closely together. They analyze existing data and find patterns and identify data requirements needed to solve a problem.

1.3.2 DATA DISCOVERY, ACQUISITION AND GENERATION

Once the problem statement and data requirements are established, data acquisition or generation can begin. Data acquisition can involve generating data through running existing actuarial models, sourcing data through admin systems, and gathering alternative data from publicly available or proprietary data. In certain cases, live streaming of data from real-time inference is needed. This phase is critical in the success of a machine learning project, and the quantity and quality of the data collected available for training the model will directly contribute to the accuracy of the results.

1.3.3 DATA TRANSFORMATION

Data in their raw form is not very informative to a machine learning algorithm. The collected or generated data cannot be directly fed into the training process for the model. The data preparation and wrangling phase includes key steps such as imputation for missing values, removing duplicated data, and correcting for invalid or noisy data. Data-preprocessing, data transformations and feature engineering for exploratory data analysis (EDA) are crucial exercises during this step.

The purpose of EDA is to analyze the datasets to summarize their main characteristics. The initial investigations on data with the support of summary statistic and graphical representations can help discover patterns, detect outliers, spot anomalies, and check assumptions. An integral step of doing EDA is data visualization where data in the form of tables or graphs and various charts can be created. Data transformation typically also includes normalizing data, scaling features, and selecting key features. It is a comparative analysis where the data are compared in visual forms to understand the relationships among variables.

1.3.4 MODEL SELECTION

Model selection involves choosing the type of model or algorithm to be used based on the use case. From the range of machine learning models available, such as supervised, unsupervised, classification, regression, clustering, and reinforcement learning, the model architecture that is fit for purpose will be selected. In a simple statistic example for numeric prediction, linear regression, random forest, and decision trees could be used. For classification problems, logistic regression, K nearest neighborhood, and Naïve bayes could be used.

1.3.5 MODEL TRAINING

In the model development lifecycle, the model training part is where actuaries attempt to fit the best combination of parameters, weights, and biases to the model to minimize a loss function over the prediction range. The goals of the model building and training are establishing the best mathematical representation of the relationship between data features and a target label in supervised learning or among the features themselves in unsupervised. Since how to optimize the machine learning algorithms needs to be specified, loss functions are a crucial component of model training. Different models would require optimizing the model with different loss functions. Common loss functions are cross-entropy, hinge loss, mean square error or quadratic loss, etc. The values generated by the loss functions will be used to measure how good the model is in terms of predicting the expected outcome.

1.3.6 OVERFITTING

Overfitting refers to a model that displays the training data too well. When a model fits more data than it needs, it starts catching the noise and inaccurate values in the dataset. This means that the noise or random fluctuations in the training data are picked up and learned as concepts by the model. As these concepts do not apply to new, broader data, it decreases the efficiency and accuracy of the model. It is a very common problem in machine learning and data science practice.

To avoid overfitting, the model training could be stopped at an earlier stage. However, early stopping can also lead to the model not being able to learn enough from the training data. In that case, the resulting model would not be accurately capturing the pattern and trends of the dataset; this is known as underfitting.

Ideally, the model selected is at the sweet spot between underfitting and overfitting. The performance of a machine learning algorithm over time should be monitored as it is learning the training data. Both the skill on the training data and the skill on a validation dataset held back from the training process could be plotted. Over time, as the algorithm learns, the number of errors for the model on the training data goes down and so do the number of errors on the test dataset. As the number of errors for the unseen test dataset start to rise again, the model's ability to generalize decreases. Therefore, the sweet spot is usually the point just before the number of errors on the test dataset start to increase where the model has good skill on both the training and unseen test datasets.

1.3.7 TRAIN TEST SPLIT

The train-test split process is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. It is a fast and easy procedure to perform, the results of which could be used to compare the performance of machine learning algorithms for the predictive modeling problem.

The procedure involves taking a dataset and dividing it into two subsets. The first subset is called the training dataset, used to train and fit the model. The other subset is called the validation dataset, and not used in training. It is instead used in the model for predictions and compared to the expected values. The objective is to estimate the performance of the machine learning model on data not used in training the model.

The train-test split procedure is appropriate when there is a sufficiently large dataset available. This requires that the original dataset is also a suitable representation of the problem domain. That means there are enough records to cover all common cases and most uncommon cases in the population. In addition to dataset size, another reason to use the train-test split evaluation procedure is computational efficiency. Some models are very costly to train, and repeated evaluation used in other procedures is unmanageable.

The procedure has one main configuration parameter, which is the size of the train and test datasets. This is expressed as a percentage between 0 and 1 for either the train or test datasets. The optimal split percentage depends on the project's objectives. Some common split percentages include 80%, 67% and 50% for the train set (e.g., 80% data for training and 20% data for testing).

1.3.8 MODEL TESTING

In this step, how well the trained model performs on the test dataset is assessed. The inference of the trained model is measured against selected metrics depending on the use cases. The output of this step is the prediction accuracy. For example, for a conventional binary classification model, accuracy and recall scores are used to assess the model's performance. Two different outcomes can be classified to evaluate false positives and true positives to truly understand the model performance. This step can be an iterative process until one model's results satisfies the requirement. Otherwise, increasing the size of the training dataset and repeating the previous steps can help to get a more comprehensive model. Similar to the regression models, appropriate metrics will be used for regression model predictions to evaluate the corresponding models, such as MAE, MSE, RMSE, etc. It is key that the model performs well on the model testing before preceding to the next phase.

1.3.9 LEARNING CURVE

A learning curve is a plot for monitoring model learning performance over experience or time. Learning curves are a widely used diagnostic tool in machine learning for algorithms that learn from a training dataset incrementally. The model can be evaluated on the training dataset against the validation dataset. After each update during training, the measured performance can be plotted to show learning curves. Reviewing learning curves of models can be used to diagnose problems such as an underfitting or overfitting model, as well as whether the training and validation datasets are suitably representative.

Generally, a learning curve is a plot that shows time or experience on the x-axis and learning or improvement on the y-axis. The metric used to evaluate learning could be minimizing loss or error. The better scores (smaller numbers) indicate more learning and a value of 0.0 indicates that the training dataset was learned perfectly, and no mistakes were made.

The shape and dynamics of a learning curve can be used to diagnose the behavior of a model and suggest configuration changes that may be made to improve learning and model performance.

A good fit is the goal of the learning algorithm and exists between an overfit and underfit model. A good fit is identified by a training and validation loss that decreases to a point of stability with a minimal gap between the two final loss values. The loss of the model will almost always be lower on the training dataset than the validation dataset. This means that some gap between the training and validation loss learning curves are expected. This gap is referred to as the “generalization gap.”

A plot of learning curves shows a good fit if the plot of training loss decreases to a point of stability and the plot of validation loss decreases to a point of stability and has a small gap with the training loss. Continued training of a good fit will likely lead to an overfit.

1.3.10 HYPERPARAMETER TUNING

Optimal accuracy can be achieved through hyperparameter tuning, changing the feature set, or adding more features. Experiments can be performed across various permutations of algorithms, datasets, features and tuning parameters as well to achieve a high accuracy model. This experimentation and iteration process can be costly and very time consuming. After n number of iterations when the process results in a desired accuracy with a set of combinations, the model can be finalized as well trained and deployed into production.

1.3.11 AUTO ML

Traditionally, the hyperparameter tuning model combinations are created manually, and run one by one to compare the accuracy of the results. With the onset of tools like Auto ML and Auto AI, this step of model selection can be accelerated by having machines programmatically test common models and deliver an analysis of which models perform the best, eventually speeding up the model development process dramatically.

With Auto ML, the data, features, algorithms, and other settings are selected to be tested. At the backend, the tool will automatically train the model on various algorithms with varied data and features and, at last, list the accuracy of all the combinations it used. From there, the precision of various runs can be compared to choose the highest accuracy of value combinations produced by the Auto ML.

1.4 MODEL DEVELOPMENT CYCLE AND MANAGEMENT

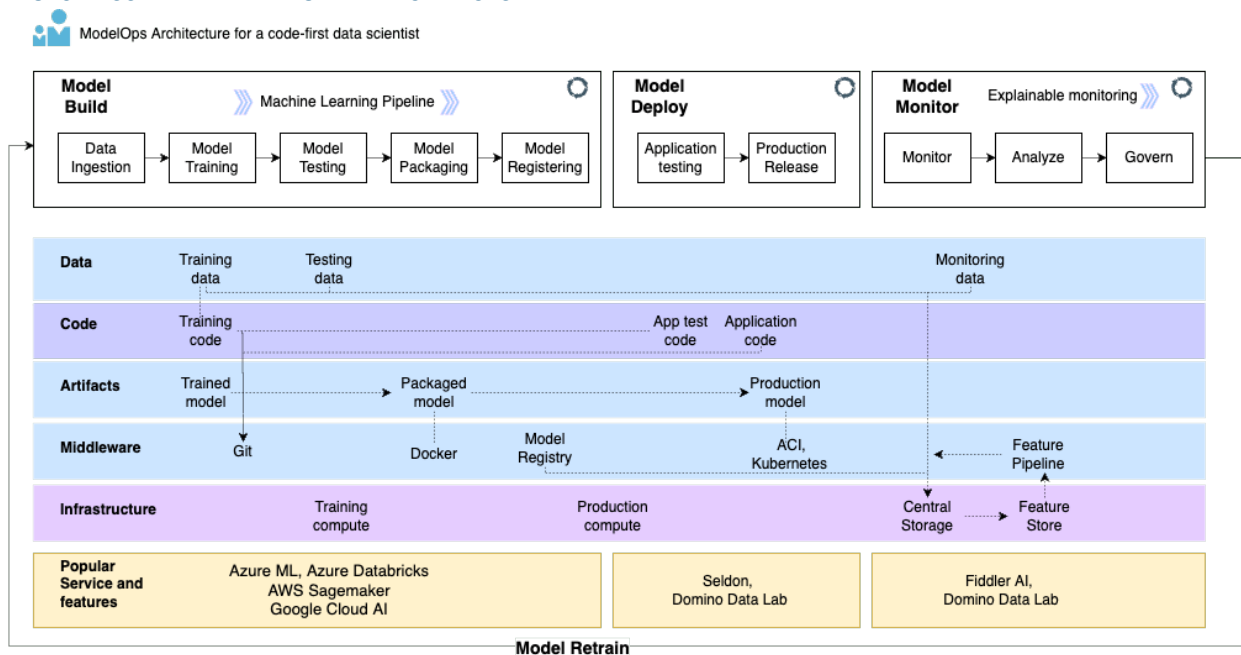
The production and operation of an AIML model after its initial development needs to be taken into consideration. Machine learning operations (ML Ops) are a set of principles and practices to standardize and streamline machine learning lifecycle management. It is an integration between the development and operational processes, where teams collaborate to build, automate, test, and monitor the end-to-end machine learning pipelines. The origin of ML Ops is no different from software development operation (Dev Ops) principles, with the added process of a retraining of machine learning models.

1.4.1 ML OPS OVERVIEW

Below is a generic ML Ops workflow that is the result of many design cycle iterations. It brings together model build, deploy, and monitor in a streamlined development framework. It is used to efficiently deliver the developed model into production and maintain the entire machine learning model development lifecycle.

The ML Ops pipeline is the top layer, and the entire architectural diagram can be leveraged for proofs of concept or implementation. It can also be executed for many types of ML solutions in any industry since it is highly modular and adaptable.

Figure 2
MLOPS ILLUSTRATIVE END TO END ARCHITECTURE



In addition, the essential elements of the ML Ops process are also mentioned under the ML Ops layer:

- **Data:** data required to build the model,
- **Code:** the programs and algorithms used to train and test the model,
- **Artifacts:** trained and tested models warehousing,
- **Middleware:** software, tool, and platform for model deployment,
- **Infrastructure:** cloud computing infrastructure, some popular services, tools, and platforms (e.g., AWS Sagemaker, Azure ML, Databricks, Google Cloud AI), as well as other associated auxiliary elements.

A single ML Ops process where it extends to enterprise-level Model Ops can only be successfully established when all elements operate together.

1.4.2 MODEL BUILD

The most fundamental and important part of the ML Ops pipeline is the model build or model development. A model will go through data ingestion, model training, model testing, model packaging, and model registering operations in this cycle. Then, the best performing model will be registered and kept in the model registry for version control and management.

1.4.3 MODEL PACKAGING AND REGISTERING

After testing the performance in the previous step, the trained model can be deployed to the UAT or Prod environment by packaging and serializing the files. It can also build a containerized docker image to wrap up the model. The serialized or containerized models from the previous stage will be registered and saved in the model registry at this stage. It is important to maintain model version control. In general, some auxiliary Python packages and tools can be utilized to achieve this, with four key components:

- **Tracking:** record and query experiments, code, data, configuration, and results.
- **Projects:** package data science code in a format to reproduce runs on any platform.
- **Models:** deploy ML models in diverse serving environments.

1.4.4 MODEL DEPLOY

Model deployment is the process of incorporating a machine learning model into an already-existing production environment to make useful business decisions based on data. The deployment module makes it possible to put the models that were created and trained in the previous module into operation. In the model deploy module, the model performance, functionalities, and behavior can be tested in the production-like (UAT or QA) environment. This will validate and ensure the model's scalability and resiliency of the model in production.

To summarize this process, there are two main components. First, application testing and production release. Second, the deployment pipeline is enabled by streamlined continuous integration and continuous deployment (CI/CD) pipelines connecting the development and production environments.

1.4.5 APPLICATION TESTING

Every model being deployed to the production environment needs to conduct an appropriate amount of essential testing, including unit, system, and traffic testing. They can ensure the model is resilient and performs well. Therefore, in this phase, actuaries and DevOps engineers should test the robustness and performance of the trained model in UAT. Models for testing are deployed in this testing environment as APIs or streaming services to deployment targets, such as Kubernetes services, container instances, or edge devices, depending on need and use case.

1.4.6 PRODUCTION RELEASE

Previously tested and approved models can be deployed in the production environment with a variety of options. This allows the business units and product users to interface with the models and generate business or operational values. Then, the product is released and deployed to the production environment enabled by the CI/CD pipelines.

1.4.7 MODEL MONITOR

Model monitoring is an operational stage in the machine learning lifecycle after model deployment. It comprises detecting changes in the deployed models, such as model degradation, model drift, data integrity, feature drift, etc.

Model monitoring works in sync with the deploy module to ensure that the deployed models are acceptable in operation and performance. Typically, a comprehensive model monitor consists of three main components, including model performance monitoring, model/data in-depth analysis, and model governance. These are all important tools to help us gain a rational and comprehensive understanding of the data and models in a production environment.

Metrics are defined to measure and interpret the parameters that are used for model performance. The models in the lifecycle can be monitored based on some key metrics. For example, feature importance value can be used to explain which features have a more significant impact on the output. Changes in the feature importance over time can indicate model instabilities. In addition, Shapley (SHAP) value is a useful metric to evaluate the explainability of the model. A positive SHAP value means a positive impact on prediction, leading the model to predict one. A negative SHAP value means a negative impact, leading the model to predict zero.

Metrics like feature importance value and SHAP value are used to monitor a model's life cycle and execute reassurance solutions, so the machine learning application can be governed using alerts and action-based quality assurance and control. This ensures a robust monitoring mechanism for the production system.

1.4.8 MODEL ANALYZING

For optimal performance and governance in connection to a business decision, it is crucial to monitor all models in the production environment under one central dashboard. Properly monitoring and evaluating all the models' performances in real time can be done by leveraging some model explainability techniques and cloud app servers, such as AWS CloudWatch, Azure Monitor, etc. These techniques and tools can be used to assess and evaluate important aspects of the model in fairness, bias, transparency, drift, and error analysis. The target variable, which is the statistical characteristic or metric to monitor or forecast, may vary in some unexpected ways over time. This is called model drift.

For instance, in the scenario when an ML recommender system has been deployed to suggest appropriate goods for different consumers, it's very difficult for the model itself to "guess" what items the customers are looking for on a given day. Those unpredictable patterns are not visible in the historical data. As the data was used to train the model, it may cause changes in result user behavior. To guarantee that deployed models deliver the highest and most pertinent business value, it is crucial to consider some unanticipated things or events. When model drift is noticed, one of the following should be done:

1. The model owner or quality assurance expert needs to be alerted.
2. The model needs to be switched or updated.
3. Retraining the pipeline should be triggered to retrain and update the model as per the latest data or needs.

1.4.9 MODEL GOVERNANCE

In addition, the deployed ML models should be governed to ensure the best performance aligns with business goals through monitoring and analyzed similarly to the example above. It is very necessary to set up a specific set of alerts and actions to govern the MLOps system for all models. Without proper governance, firms may put themselves in some dangerous situations, especially financial and e-commercial companies. With lower accuracy and higher bias prediction where they are below the pre-defined

threshold, the model owner or a quality assurance expert should receive an alert and be notified about this issue. Then, any decisions and actions can be executed by experts in a timely manner.

Model governance is an important component in compliance with local and global laws and regulations. Models can only be deployed to production if they meet legal preconditions as needed. For compliance, model interpretability and transparency are critical, and this comparably applies to technology companies. Many deep learning models have poor interpretability because of the large model size and too many training parameters resulting in a vague model's interpretability. This often causes certain doubts and worries to users and regulators. Hence, model auditing and reporting should be conducted as well in order to provide end-to-end traceability and explainability for production models. This way, the ML system is well-governed to serve the business needs.

1.4.10 MODEL RETRAINING

Retraining a model is done to make sure that the deployed model consistently produces the best results. It is crucial to identify what makes an output the most accurate for your business use case and how to assess this accuracy for a successful retraining process. When a configured model monitoring procedure sends an alert of a problem in performance and degradation, an automatic retraining procedure can be used.

As previously indicated, the primary reason for carrying out an automatic retraining procedure is performance deterioration. With the most recent data, model performance is always anticipated to be the best, but the cost increases with the frequency of retraining. By doing an offline experiment, the leading time for data drift and conceptual drift impact on the model performance below a baseline threshold can be determined. Any modification in data, model, or code should be followed by model retraining as part of the continuous integration (CI) process. As a best practice, monitoring for retraining models via dashboards, alerts, and reports should be a shared team effort, involving machine learning engineers, data scientists, actuaries, and business stakeholders.

Section 2: Literature Review

This section reviews a collection of academic papers that have explored various approaches to gain modeling efficiencies using AIML. We have provided a high-level summary along with key takeaways in an effort to help readers quickly grasp the key concepts explored in these papers. We also provided references to each paper should the readers want to get more detail.

We also made mention of past work performed by the Society of Actuaries or other actuarial organizations that are related to this topic. This section builds upon the general overview of AIML, including the history and recent developments, fundamentals of AIML, key phases to build an AIML model and model development cycle, and model management provided in section [1. Predictive analytics, artificial intelligence, and machine learning overview](#).

We first summarize previous research performed by the SOA and other actuarial organizations that are related to this paper in subsection [2.1 Relevant Research From The SOA and Other Actuarial Organizations](#). Then, we go over academic papers that applied AIML in various forms to address nested stochastic modeling challenges in subsection [2.2 Academic Papers Exploring AIML For Actuarial Modeling](#).

We provided the following information for each paper:

- Reference and link to the original paper
- Overview and objective
- Approach
- Key takeaways

Lastly, this section is not meant to be exhaustive, and readers will find additional academic and research papers that will prove useful in developing AIML models to address modeling runtime concerns. We also included papers that we felt may be useful to the reader even if those papers are not focused on addressing model runtime concerns (for instance, we included papers that provide an overview of actuarial applications of AIML and papers that provide an introduction to AIML for actuaries).

2.1 RELEVANT RESEARCH FROM THE SOA AND OTHER ACTUARIAL ORGANIZATIONS

2.1.1 A TOUR OF AI TECHNOLOGIES IN TIME SERIES PREDICTION

Author: Victoria Zhang

Issue year: 2019

Accessing the paper

This paper can be accessed through the link below:

[A Tour of AI Technologies in Time Series Prediction](#) (Zhang, 2019)

Paper overview and objective

This paper reviews multiple machine learning (ML) and deep learning (DL) models and explain how those models work and their possible applications in the actuarial field.

This paper also demonstrates how AIML can be used for time series prediction. It explores the models that are best suited for time series problems, including machine learning, deep learning, and recurrent neural network models.

Lastly, the paper also provided insights into the advantages and challenges actuaries are facing with AI technology, and how actuaries could adjust to this thriving technology.

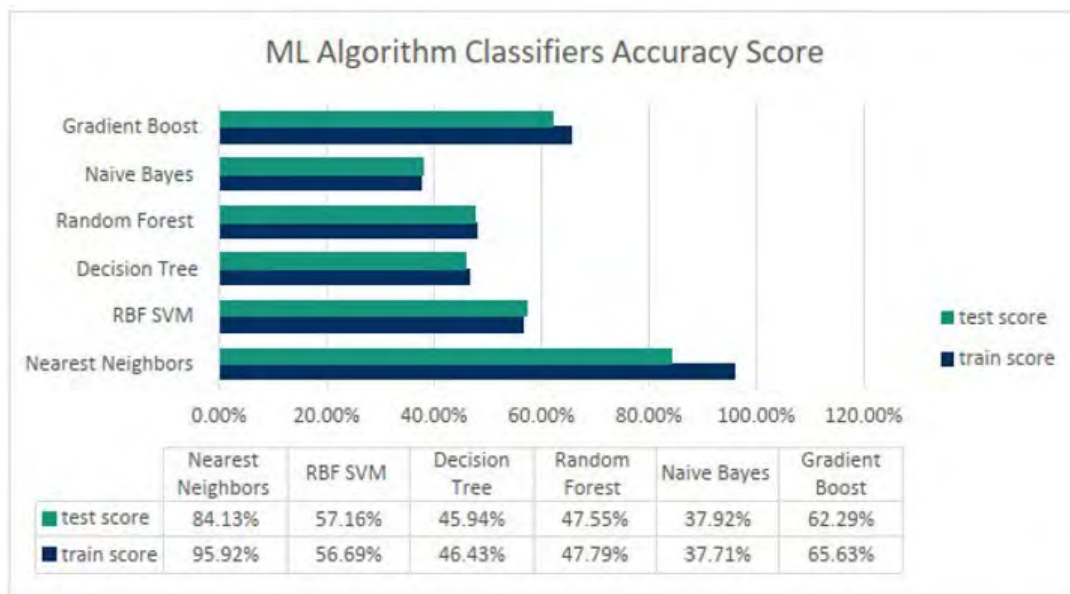
Approach and analysis

The paper first introduces six supervised learning classifiers to the reader:

- Naive Bayes Classifier
- K-Nearest Neighbor (KNN)
- Support Vector Machine (SVM)
- Decision Trees
- Random Forests
- Gradient Boosted Trees (GBT).

This paper tested the six models to classify whether the NASDAQ index trend was going up or down or would stay stationary based on the last T data points. After training all of the six classifier models with the same data, the model applied the trained models to the reserved 30% testing data, which had never been seen by those models to evaluate the classification accuracy. The end results showed three of the models were effective with an accuracy score higher than 50%. The chart below shows the classification accuracy score among the six classifiers.

Figure 3
CLASSIFICATION ACCURACY SCORE BY SIX ML CLASSIFIERS



Then, this paper explored deep learning models, introduced Deep Neural Networks (DNN) and discussed its

use in time series prediction. Two types of DNNs, Multi-Layer Perceptron (MLP) and Convolutional Neural Networks (CNN), were compared using the case study of predicting bitcoin prices. The result was that CNN was the preferred model in the time series analysis because the testing results between the two models were both excellent and CNN only required half of the parameters from MLP.

The paper also explored using recurrent neural networks (RNN) for time series predictions, and its vanishing and exploding gradient issue. Furthermore, it introduced the long short-term memory (LSTM) model, which is an RNN variant and can mitigate the vanishing and exploding gradient issues. Two case studies were analyzed. The first one was to design an LSTM network using the stock price and current NASDAQ index value to predict the future index value and compare it against the traditional forecasting methods. The LSTM can uncover correlations that are typically not seen with traditional forecasting techniques and is better at modeling non-linear relationships and long-term forecasting where the traditional method is better at forecasting short-term and univariate problems. The other case study was to use LSTM to detect anomalies in time series data. The idea was to train an LSTM model to predict future data and, when there is a vast difference between the prediction and the incoming data, then there might be an outlier.

Key takeaways

This report provided a high-level overview of AI and how it could be applied to actuarial work. One of the goals was to inspire actuaries to think about how actuaries could bring AI into their daily work. The second goal of this report was to provide some concrete examples of solving different time-series related problems, to provide some clear starting point for time-series related problems.

At the end, the paper talked about the main challenges and obstacles in adapting AI technologies that come from censorship and privacy concerns, regulatory concerns, and significant initial investment and ongoing costs.

2.1.2 EMERGING DATA ANALYTICS TECHNIQUES WITH ACTUARIAL APPLICATIONS

Authors: Marie-Claire Koissi, Herschel Day, Vicki Whitledge

Issue year: 2019

Accessing the paper

This paper can be accessed through the link below:

[*Emerging Data Analytics Techniques with Actuarial Applications*](#) (Koissi, et al., 2019)

Paper overview and objective

The main aim of this research was to survey emerging data analytics techniques and discuss their evolution and growing use in the actuarial profession, including both life and non-life insurance.

Approach and analysis

The survey results are organized in the paper as follows:

1. Summarized the data analytics process, summarized the data sources, and discussed methods for data visualization.
2. Brief overview of data analytic techniques used in supervised learning (regression and generalized linear models, Trees, Neural Networks, and Predictive modeling), unsupervised techniques (Principal component analysis, Cluster Analysis, Generic Algorithms, and Neural Networks), Markov Chain Monte Carlo Simulation and Bayesian Analysis.
3. Overview of enhanced data analytic technologies and actuarial application. The actuarial applications discussed include machine learning technologies for mortality rate forecasting in life insurance, health care claims modeling in health insurance, reserves in life/non-life, and insurance fraud and other areas discussion for life/non-life. In addition, the paper introduced some actuarial packages in R and Python.
4. Case studies using open-source technologies (R, SAS and Python) for actuarial computational work.

Key takeaways

This paper found that advanced data visualization techniques and their use is expanding among actuaries, and then gave a brief overview of several data analytic techniques. In addition, this paper found that enhanced data analytic technologies are rising and their use is spreading in all areas of actuarial science. Open-source data analytic software can help actuarial practitioners and researchers efficiently take advantage of these new opportunities.

2.1.3 LITERATURE REVIEW: ARTIFICIAL INTELLIGENCE AND ITS USE IN ACTUARIAL WORK

Authors: Nicholas Yeo, Raymond Lai, Min Jyeh Ooi, Jie Yin Liew

Issue year: 2019

Accessing the paper

This paper can be accessed through the link below:

[Literature Review: Artificial Intelligence and Its Use in Actuarial Work](#) (Yeo, et al., 2019)

This paper highlighted the AIML technologies that will interact with the actuarial profession to provide a frame of reference for actuaries to use in actuarial work. First, the paper gave a brief introduction to AI, and described the history of AI. Then, the paper talked about the literature review of how AI can be used in different lines of actuarial work, including motor insurance pricing, loss reserving, mortality modeling, underwriting, and fraud and claims. At the end, the paper summarized the conclusions regarding the impact of AI on actuarial work and discussed the benefits, drawbacks, and challenges of implementing AI in actuarial work.

2.1.4 CLOUD COMPUTING AND MACHINE LEARNING USES IN THE ACTUARIAL PROFESSION

Authors: Van Beach, Alexandre Boumezoued, Josh Dobiac, Jonathan B. Glowacki, Joe Long, Antoine Ly, Makho Mashoba, Zohair Motiwalla, David South, Benjamin Buttin

Issue year: 2019

Accessing the paper

This paper can be accessed through the link below:

[Cloud Computing and Machine Learning Uses in the Actuarial Profession](#) (Beach, et al., 2019)

Cloud computing is growing in importance for actuarial models, especially given its ability to provide on-demand computing resources.

This paper introduced the cloud service models and their impact on the actuarial profession, discussed the use of the cloud in terms of financial modeling and actuarial processes analytics, and discussed the use of the cloud in terms of the increased ability to collect more data (and more granular forms of data) to perform advanced analytics.

At the end, this paper provided an outlook for the potential ways expanded use of cloud technology combined with machine learning/AI could impact the actuarial profession as the technology evolves. It discussed the potential impact on various actuarial functions, including pricing, valuation and reserving, enterprise risk management, and experience study and assumption development.

Below are the future opportunities that actuaries can potentially apply the cloud-based technology to:

1. Streamline processes and data, allowing actuaries to spend more time analyzing and communicating results,
2. Develop and implement granular, risk-based pricing tools,
3. Assist with the development of algorithms to target specific potential policyholders, and
4. Optimize pricing to maximize the risk / reward trade-off of products

2.1.5 CONSIDERATIONS FOR PREDICTIVE MODELING IN INSURANCE APPLICATIONS

Authors: Eileen Burns, Gene Dan, Anders Larson, Bob Meyer, Zohair Motiwalla, Guy Yollin

Issue year: 2019

Accessing the paper

This paper can be accessed through the link below:

[Considerations for Predictive Modeling in Insurance Applications](#) (Burns, et al., 2019)

This paper is a research report that includes a review of existing literature and current industry practice, and a comprehensive set of considerations for predictive modeling in insurance applications.

This paper identified the components of an advanced predictive modeling function that, based on our experience, a company must address to have the best chance of gaining buy-in from stakeholders. The

question of defining leading practices in each area was approached from a few angles, including completing a literature review, conducting a survey of SOA members to assess what was currently being used for each component, and compiling two case studies with a single life insurer to document examples of predictive modeling in development and in production within a familiar context. The case study approach was used to showcase how one company made many of the decisions required to adopt predictive modeling and to highlight successes and struggles along the way.

At last, the paper summarized their guidance as a compilation of concerns practitioners should address, with suggestions offered for how one might make the necessary decisions in a given application.

2.1.6 NESTED STOCHASTIC MODELING FOR INSURANCE COMPANIES

Authors: Runhuan Feng, Zhenyu Cui, Peng Li

Issue year: 2016

Accessing the paper

This paper can be accessed through the link below:

[*Nested Stochastic Modeling for Insurance Companies*](#) (Feng, et al., 2016)

Paper overview and objective

In recent years, the insurance industry has been moving toward more detailed and sophisticated financial reporting standards and practices, and the computational burden and technical difficulty has been rising with the increasing use of nested stochastic modeling. The purpose of this paper was to provide a resource to help financial reporting actuaries better understand a variety of nested stochastic techniques available, both in the industry and academic literature. In addition, the paper performed a comparative analysis of the accuracy and efficiency for the techniques selected.

Approach and analysis

The techniques selected in this paper were tested using two case studies.

Case I: Risk-neutral valuation of guaranteed minimum accumulation benefit

This case was designed to capture only the essential structure of a nested simulation. All the closed-form solutions can be obtained and used as benchmarks against other techniques. The focus was on the accuracy and validity of all techniques.

Case II: *AG-43 CTE calculation for guaranteed lifetime withdrawal benefit*

This case was intended to resemble an actual financial reporting model, and the most necessary elements of a financial reporting model on a single cell were included. Since case II had a more significant increase of structural complexity compared to case I, it was more to provide a realistic contrast on the modeling efficiency of various techniques.

Below is the summary of the comparison of the eight techniques analyzed:

Figure 4
COMPARISON OF EIGHT NESTED STOCHASTIC TECHNIQUES

Method	Brief description	Pros	Cons
A. Closed-form solutions	Valuations are largely based on closed-form formulas that produce exact values or approximation	Very accurate and efficient; Provides benchmarks approximation against which all other techniques can be tested.	Limited to certain stochastic models; Requires expertise to develop solutions.
B. Crude Monte Carlo	Straightforward simulations based on product design and projection of cash flows	Easy to implement; Requires minimal analysis.	Can be extremely time and resource consuming.
C. Optimal budget allocation	Static allocation of resources between two levels of simulations according to certain criteria	Easy to implement formula-based allocation; No more modeling beyond crude MC.	Existing allocation strategies depend on specific risk measures; Can be difficult to generalize.
D. Sequential allocation	Dynamic allocation of resources	Dynamically allocate budget; Ideal use of resources.	Can be slow due to conditional statements in computational algorithms.
E. Preprocessed inner loops	Preprocess inner loop results under representative scenarios and infer results under desired scenarios from those under similar representative scenarios.	Easy to understand and implement; Modest accuracy in low dimensions.	Difficult to determine boundary points to cover all points for interpolations; Difficult to select grid points in high dimensions.
F. Least-Squares Monte Carlo (LSMC)	Approximate inner-loop items by curve fitting, typically with polynomial approximations	Modest accuracy with small number of inner-loops; Can be used for extrapolation.	Little guidance on basis functions; Difficult to select cross-terms in high dimensions.
G. Least-Squares Monte Carlo with basis selection	Replace inner-loop items by exponential sum approximations with automatic bases selection.	Can be more efficient than F due to automatic selection of basis functions.	More analysis involved; Limited literature on high dimensions.
H. Numerical partial differential equation (PDE) methods	Replace inner-loop items by employing numerical PDE algorithms	Can be highly accurate and efficient; Possible reduction of dimensions to improve efficiency.	Requires expertise for stochastic analysis; Special algorithms for high-dimension PDEs.

Key takeaways

The main conclusions from this paper's analysis:

1. Analytical and numerical PDE methods (Method H) are, in general, the most efficient and accurate approaches given a small computation budget.
2. Least-Squares Monte Carlo methods (Method F) are significant improvements of the crude Monte Carlo methods (Method B).
3. The method of preprocessed inner-loops (Method E) is easiest to implement. Similar to the LSMC (Method F), it can be quite efficient in low-dimension problems but appears much less so in high dimensions.

For future work directions, all mathematical formulation and methods in this report can be further extended to work with models of multiple equity indices and combined benefits, although their comparative advantages may be different in higher dimensions. Another direction would be to include stochastic interest rate and stochastic volatility models in the analysis.

2.2 ACADEMIC PAPERS EXPLORING AIML FOR ACTUARIAL MODELING

2.2.1 EFFICIENT DYNAMIC HEDGING FOR LARGE VARIABLE ANNUITY PORTFOLIOS WITH MULTIPLE UNDERLYING ASSETS

Authors: X. Sheldon Lin, Shuai Yang

Issue year: 2020

Accessing the paper

This paper can be accessed through the link below:

[*Efficient Dynamic Hedging for Large Variable Annuity Portfolios with Multiple Underlying Assets*](#) (Lin, et al., 2020)

Paper overview and objective

In an attempt to reduce computational time in calculating VA liabilities, this paper studied and analyzed the best way to efficiently calculate the total VA liability and partial dollar Deltas for large VA portfolios with multiple underlying assets. The proposed algorithm, including the nested simulation, used small sets of selected representative policies and representative outer-loops.

In traditional VA hedging, most insurers perform in-house hedging with a two-step process. This involves a set number of outer-loop scenarios for economic scenarios on asset returns, then projecting the total VA liability in those shocked scenarios. Thus, the computational time is proportionally scaled based on the number of policies, number of outer-loops, and number of inner-loop scenarios, etc. As a result, reducing the simulation runtime is identified as a critical issue to insurance companies when managing large VA portfolios.

Approach and analysis

Due to limitations, such as hardware, software, and computer memory, full nested simulation still cannot be executed in practice. To extend from other research on this topic, this paper further discussed the below directions:

1. Studying the selection of a set of representative outer-loop scenarios for policies invested in multiple underlying assets,
2. Designing an algorithm to estimate the total VA liability distribution and other portfolio quantities such as partial dollar Deltas, and
3. Extending the algorithm to a multi-period setting and performing P&L projections of a dynamic Delta hedging strategy.

This paper demonstrates the importance of dynamic hedging for VA portfolios and illustrates the proposed method using P&L projections.

The algorithm proposed in this paper is an extension of the algorithm by Lin & Yang (2020). It analyzes the selection of representative policies using the model-assisted population sampling approach and scenarios clustering with multiple assets. Then, the spline regression model is used to estimate the VA liabilities of the representative policies from the sampling.

This paper model-assisted the population sampling framework to select a set of representative policies to

estimate the total VA liability. To each predictive VA liability for a policy of an outer-loop scenario, this paper employed a linear model with a policy's attributes. This model is expressed below as:

$$L_p(s) = b'x_{p,0} + e_p(s)$$

This paper utilized the Cube algorithm by Deville & Tille (2004) to select a balance sample. This algorithm contains two phases: a flight phase and a landing phase, supposing N to be the population size and each population unit of r dimensions. This paper uses the attribute vector, $x_{p,0}$.

The flight phase iteratively translates inclusion probabilities to a vector of at least $(N-r)$ zero or ones. The balance condition can be formed in a matrix where it implies that all balanced samples form a subspace of \mathbb{R}^N with dimension $N - r$. Thus, in each iteration of the flight phase, the inclusion probability vector is positioned randomly inside the kernel space of A until it reaches a point that is close to a vertex of N dimensional hypercube. The detailed steps in each iteration of the flight phase can be found in the paper. In the landing phase, each non-integer element resulting from the flight phase is adjusted to either zero or one by linear programming. The resulting vector with only zeros and ones gives a nearly balanced sample.

Key takeaways

The results are presented in the figures below, where each figure corresponds to a generic real world economic scenario over the 24-week period. The top and bottom three subfigures in each figure correspond to the settings of 2,000 and 4,000 representative policies, respectively. The subfigures in each row, from the left to the right, display 50 curves of the estimated partial dollar Delta with respect to the S&P 500 index, the S&P 600 index, and the estimated total VA liability. The red dashed curves are the mean trajectories of the 50 estimated curves shifted upward and downward by 5%.

The results show that the estimated total liability curves from different simulation runs are almost parallel to each other; most of the estimated liability curves fall inside the 5% band of the mean curve, implying the algorithm is robust in estimating the total liability under both settings; and the estimated partial dollar Delta's curves are relatively more volatile, but the estimated partial dollar Delta's curves of each scenario show the same overall trend throughout the time period, and they, in general, go against the total liability movements.

Figure 5
RESULTS: ESTIMATES OF DIFFERENT QUANTITIES FROM DIFFERENT RUNS (SCENARIO 1)

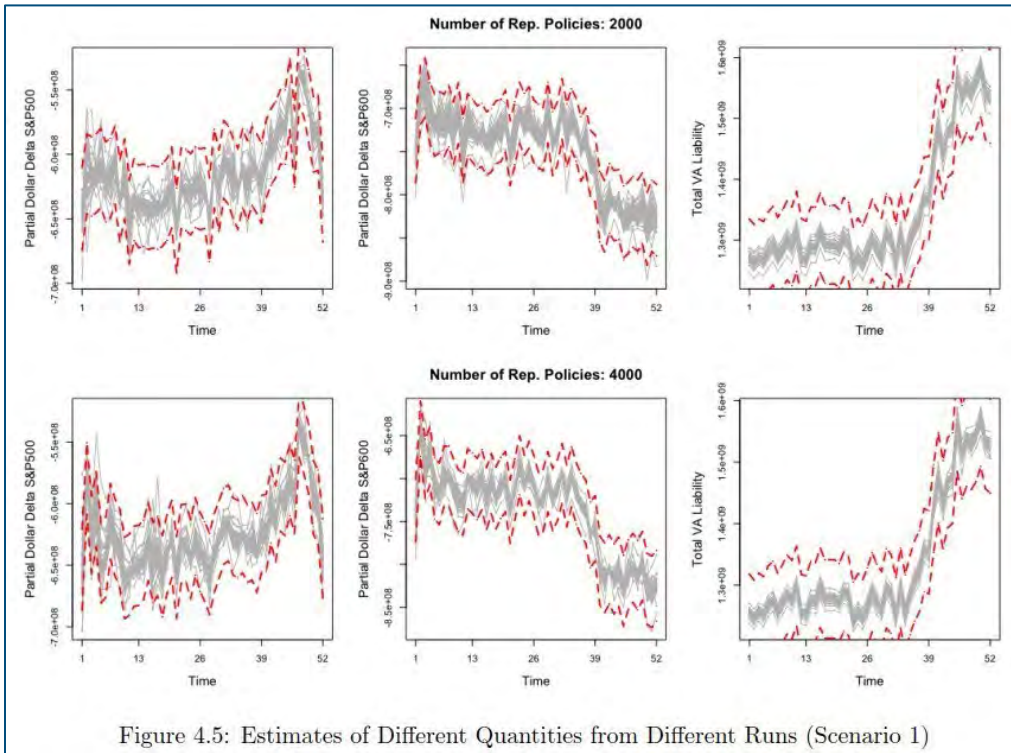


Figure 4.5: Estimates of Different Quantities from Different Runs (Scenario 1)

Figure 6
RESULTS: ESTIMATES OF DIFFERENT QUANTITIES FROM DIFFERENT RUNS (SCENARIO 2)

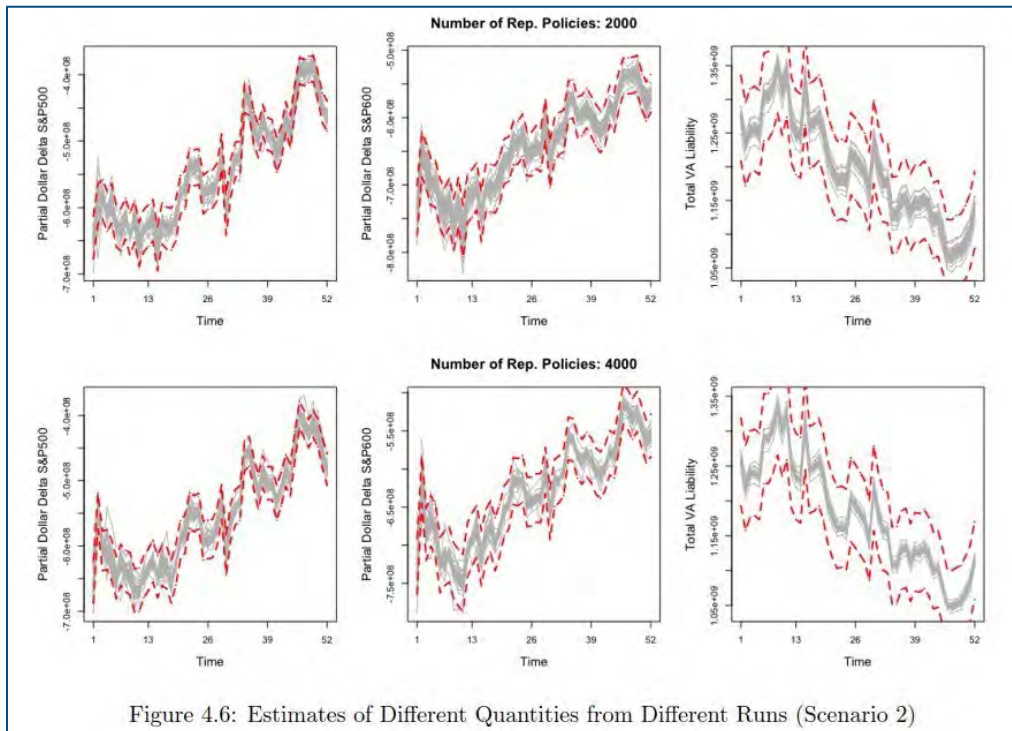


Figure 4.6: Estimates of Different Quantities from Different Runs (Scenario 2)

2.2.2 EFFICIENT NESTED SIMULATION FOR CONDITIONAL TAIL EXPECTATION OF VARIABLE ANNUITIES

Authors: Ou Dang, Mingbin Feng, Mary R. Hardy

Issue year: 2019

Accessing the paper

This paper can be accessed through the link below:

[Efficient Nested Simulation for Conditional Tail Expectation of Variable Annuities](#) (Dang, et al., 2019)

Paper overview and objective

To reduce the substantial computational burden associated with nested Monte Carlo simulations in Variable Annuities risk modeling, this paper proposed an Importance-Allocated Nested Simulation (IANS) method, using a two-stage process. The first stage is to identify the tail scenarios most likely to contribute to the CTE risk measure. There is a high level of flexibility in the choice of proxy models. The second stage allocates the entire inner simulation computational budget to the scenarios identified in the first stage.

Approach and analysis

The paper discussed the dynamic hedging for common types of VA riders, and describes the process of a standard nested simulation, presented the new approach they proposed, and performed numerical experiments to illustrate the performance of the proposed model. Below is the outline of the IANS model.

Algorithm: Importance-Allocated Nested Simulation of losses for a Delta-hedged VA contract with a single payout date T (T denotes the final expiration date of the guarantee).

input: – Underlying real-world and risk-neutral asset models with parameters.
– VA contract, term T , and fully specified dynamic hedging program.
– The risk measure and level, e.g., CTE_α :

output: CTE_α for the losses of Delta hedging the VA contract of interest.

Initialization: Simulating J outer scenarios, each is a T -period simulated stock price sample path under the real-world measure.

Stage I: Identification of proxy tail scenarios

(I.1) Select a proxy financial derivative and associated asset model that provides tractable, analytic hedge costs for which the payoff is expected to be well correlated to the VA guarantee costs.

(I.2) Calibrate the proxy asset model to the underlying risk-neutral asset model in inner-level simulations.

(I.3) Implement two-level nested simulation, but with the analytic hedge calculations for the proxy derivative and asset model replacing the inner simulation step.

(I.4) Identify $(1-\zeta) J$ proxy tail scenarios with the largest simulated loss in step (I.3) for some $\xi \in [0, \alpha]$
 ζ is the proxy confidence level.

Stage II: Nested simulation with concentrated computation budget

(II.1) Allocate remaining computational budget to the $(1-\zeta) J$ proxy tail scenarios.

(II.2) Implement the inner simulation step of the two-level nested simulation model with the original risk-neutral asset model and VA payoff, but only for the $(1-\zeta) J$ outer scenarios identified in step (I.4).

(II.3) Identify the $(1-\alpha) J$ largest liability values based on the inner simulations.

(II.4) Compute CTE_α as the output.

Key takeaways

This paper illustrated a simulation procedure for estimating the CTE of liabilities of a VA dynamic hedging strategy. The IANS method proposed takes advantage of the special structure of the CTE by first identifying a small set of potential tail scenarios from the first tier of simulation based on a proxy for liabilities calculated from a closed-form solution, then focus the simulation budget on only those scenarios. The paper conducted extensive numerical experiments on Guaranteed Minimum Maturity Benefit (GMMB) and Guaranteed Minimum Accumulation Benefit (GMAB) contracts. The numerical results showed that the proposed IANS method can be up to 30 times more efficient than a standard Monte Carlo experiment, measured by the relative mean squared errors, under the same computational budget.

This paper used GMMB and GMAB features as an example to perform the analysis. If we would like to consider using the method in this paper for our research, we need to decide what proxy asset to use for the VA block with Guaranteed Minimum Death Benefits (GMDB) and Guaranteed Minimum Withdrawal Benefits (GMWB) as needed in step 1.1 above, and the proxy confidence level. For the proxy tail scenario selection, those scenarios do not need to accurately assess the liability values. The proxy step is to ascertain a ranking of the liabilities by outer scenarios, so as long as the ranking of losses between the proxies and original models are highly related, the proxy is good enough. For the confidence level ζ , the method means the proxy tail scenarios are the $(1 - \zeta) J$ outer scenarios with the largest simulated liabilities based on the proxy calculation. We use these to identify the largest $(1 - \alpha) J$ simulated liabilities based on the inner simulations assuming that, with high confidence, the $(1 - \alpha) J$ true tail scenarios are a subset of the $(1 - \zeta) J$ proxy tail scenarios. There is a tradeoff for ζ selection between a high likelihood of including the true tail scenarios and a high concentration of simulation budget in stage II. In this paper, they used a safety margin $(\alpha - \zeta)$ of 5%.

The IANS method also inspires efficient experiment designs in other financial and actuarial applications where the CTE is estimated by Monte Carlo simulation. For future work, one direction is to consider a more rigorous and systematic approach in selecting n , the threshold for tail scenarios to be considered for nested simulations. Another direction is to improve the efficiency of nested simulation of other products such as Guaranteed Minimum Income Benefits (GMIB) and GMWB.

2.2.3 EFFICIENT SIMULATION DESIGNS FOR VALUATION OF LARGE VARIABLE ANNUITY PORTFOLIOS

Authors: Ben Mingbin Feng, Zhenni Tan, Jiayi Zheng

Issue year: 2020

Accessing the paper

This paper can be accessed through the link below:

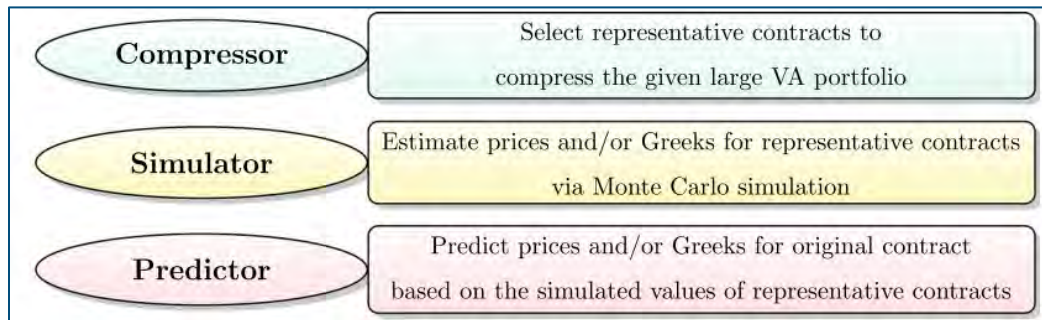
[*Efficient Simulation Designs for Valuation of Large Variable Annuity Portfolios*](#) (Feng, et al., 2020)

Paper overview and objective

The valuation of large variable annuity portfolios is an important enterprise risk management task, but is computationally challenging due to the need for simulation. To increase modeling accuracy, this paper identified three major components in an efficient valuation framework and proposed optimal experimental designs and provides analytical insights for each component. Below is the three-component Valuation Framework for Large VA Portfolios:

Figure 7

THREE-COMPONENT VALUATION FRAMEWORK FOR LARGE VA PORTFOLIOS

*Approach and analysis*

This paper provided innovative and efficient experimental designs for all three components of the valuation framework in the chart above and proposed a new method for valuation of large VA portfolios. The three components of the general framework are analyzed:

1. Proposed the SRSC (simple random sampling and clustering) compressor that selects representative contracts in a given large VA portfolio in two steps. The first step is to draw a random sample of size n from the given VA contracts. Then, apply a chosen clustering algorithm to the selected random sample.
2. Proposed the two-stage simulation budget allocation that optimally allocates a given computation budget among the representative VA contracts output by the compressor.
3. For the predictor, this paper proposed a simple benchmark metamodel, called the cluster size multiple (CSM) metamodel, which requires minimal computational requirements but has reasonable accuracy. The CSM metamodel may be too simplistic for practical applications but can serve as a worst-case benchmark for the development of new predictors within the three-component valuation framework.

Then, the paper presented some test results to demonstrate the efficiency of the proposed three-component valuation framework. The paper illustrated the efficiency of the compressor, simulator, and predictor in isolation, as well as in combination. The accuracy and speed of our proposals are compared to those in other state-of-the-art methodologies.

Key takeaways

Below are the key conclusions of the paper:

1. The SRSC compressor identifies a small set of representative contracts and has a provable performance guarantee.
2. The two-stage simulation budget allocation is shown to have satisfactory performance.
3. The CSM metamodel may be too simplistic for practical applications but can serve as a worst-case benchmark for the development of new predictors within the three-component valuation framework.

For future work, one direction is to further explore the synergies among different components when designing efficient experiments. Another direction is to address the simulator to target on minimizing the variance for estimating the original portfolio value, not the representative portfolio value by formulating and solving a new budget allocation problem.

2.2.4 FAST AND EFFICIENT NESTED SIMULATION FOR LARGE VARIABLE ANNUITY PORTFOLIOS: A SURROGATE MODELING APPROACH

Authors: X. Sheldon Lin, Shuai Yang

Issue year: 2019

Accessing the paper

This paper can be accessed through the link below:

[Fast and Efficient Nested Simulation for Large Variable Annuity Portfolios: A Surrogate Modeling Approach](#)
(Lin, et al., 2019)

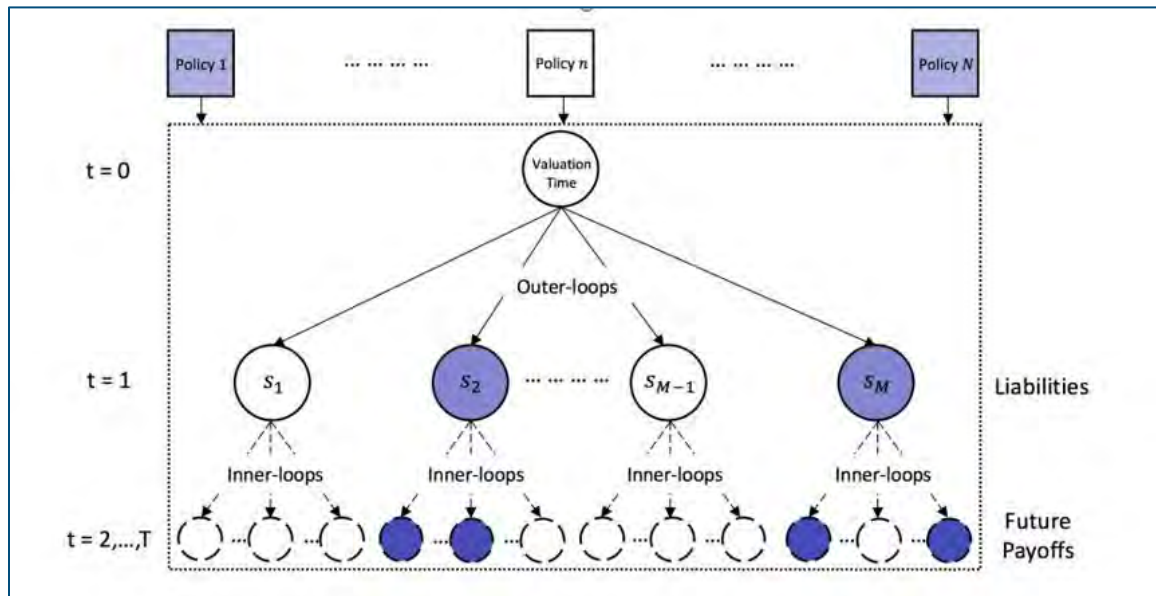
Paper overview and objective

For both regulatory and hedging purposes, the variable annuity portfolio managers are often required to compute the predictive distribution of the total VA liability in a timely basis. The heterogeneity and path-dependency of the VA guarantees cause the traditional nested-simulation to be extremely time-consuming to run. To address this important issue, this paper proposed a surrogate model-assisted nested-simulation algorithm, which incorporates several statistical tools such that the proposed algorithm can closely approximate the predictive total liability distribution at a significantly reduced computing time by running fewer numbers of inner-loops, outer-loops and policies as shown in the figure below. The method is called fast nested simulation.

This paper denoted the valuation time and the future time point by $t=0$ and $t=1$, respectively, and separated the valuation of the liability of a VA block into the steps below and proposed a way to reduce the model runtime for each step:

1. The first step is to project each policy's account value from $t=0$ to $t=1$ using many outer-loops where each outer-loop represents a scenario.
2. The second step, the inner-loops are simulated at each outer-loop to calculate the fair value of the VA liability by averaging the present values of the insurer's cash flows.
3. The third step is to sum up the liabilities from all policies in the VA portfolio to get the total liability of the entire portfolio.

Figure 8
ILLUSTRATION OF THE PROPOSED NESTED SIMULATION



Approach and analysis

First, this paper introduced a spline regression approach to approximate the relationship between a policy's account value and its liability. This is to reduce the model running time for step 2 above. Below are the detailed steps:

1. Obtain the simulated liabilities and the predicted account values in the selected scenarios of policy j , L_0 and AV;
2. Fit a spline regression using AV and L_0 ; and
3. Obtain the fitted value L from the fitted spline model to represent the liability of policy j .

Secondly, the paper proposed a clustering method to select a set of outer-loops, which are called representative outer-loops, so that the predicted liabilities are simulated only at some selected account values. This is to reduce the model running time for step 1 above. Below are the detailed steps:

1. Define the real-world asset returns $R = \{R_1, \dots, R_M\}$;
2. Decide the number of representative outer-loops m ; and
3. Run k-means algorithm with $k = m$ with R , to find the cluster centers, and find the m scenarios that are closest to the center. Those are the final m asset returns to use.

Thirdly, the paper reviewed the model-assisted finite population estimation framework, which is used to reduce the number of policies. A balanced sampling algorithm from the population sampling framework is used for selecting a set of representative policies. This is to reduce the model running time for step 3 above. The cube sampling method is used. The algorithm has two phases: flight phase and landing phase. Assume a population with N units and each unit is equipped with r attributes. The flight phase translates the given inclusion probabilities to a vector of at least $(N - r)$ zeros or ones. The landing phase then converts the non-zero/one units to either zero or one, giving an approximately balanced random sample.

Key takeaways

To test the proposed methods, this paper constructed a synthetic portfolio with 100,000 VA policies that realistically reflected a real VA portfolio. The predictive distribution of its total liability was computed using both the full simulation algorithm and the proposed simulation algorithm to compare the result of the proposed model and a full nested-simulation model. The overall distribution was well approximated with the relative error being 0.26% for the mean and 0.20% for the standard deviation. However, the runtime was reduced from six days using 60 CPU to 37 minutes using 4 CPU.

The proposed method also works well for other nonhomogeneous insurance portfolios and different scenario generators since the proposed framework does not assume any specific model for the dynamics of the underlying asset.

For future work direction, a multi-dimensional clustering technique might be considered to:

1. develop an efficient algorithm that does multiperiod prediction,
2. investigate situations where multiple underlying assets are involved application,
3. consider applying the proposed framework to calculate other quantities for large VA portfolios such as Greeks, and
4. apply the proposed algorithm to the valuation of other insurance portfolios such as universal life insurance (UL), variable universal life insurance (VUL) portfolios.

2.2.5 NESTED SIMULATION IN PORTFOLIO RISK MEASUREMENT

Authors: Michael B. Gordy, Sandeep Juneja

Issue year: 2008

Accessing the paper

This paper can be accessed through the link below:

[*Nested Simulation in Portfolio Risk Measurement*](#) (Gordy, et al., 2008)

Paper overview and objective

To reduce the computational burden of nested simulation, this paper showed that a relatively small number of trials in the inner step can yield accurate estimates and analyzed how a fixed computational budget may be allocated to the inner and outer steps to minimize the mean square error of the resultant estimator. Then, the paper introduced a jackknife procedure for bias reduction and a dynamic allocation scheme for improved efficiency.

Approach and analysis

First, the paper set out a general modeling framework for a portfolio of financial instruments.

Secondly, the paper introduced the nested simulation methodology and defined the parameters that would be tested. In the outer step, histories were drawn up to horizon H . For each trial in the outer step, there is an inner-step simulation. Let L be the number of trials in the outer step. In each of these trials, 1. Draw a single path X_t for $t = (0, H]$ under the physical measure, 2. Evaluate the accrued value at H of the interim cash flows, 3. Evaluate the price of each position at H : Closed-form price for instrument 0;

simulation with N_k “inner-step” trials for remaining positions $k = 1, \dots, K$. These paths are simulated under the risk-neutral measure, and 4. Discount back to time 0 to get the loss.

The paper then discussed the estimation of large loss probabilities, value at risk and expected shortfall. In addition, the paper relaxed the restriction that N_k is equal across k using the estimation of large loss probabilities and analyzed the optimal allocation of computational resources between the two stages that minimized the mean square error of the resultant estimator and arrived at the conclusion that higher computational resources should be allocated to a position with higher contribution to bias and lower computational effort. Furthermore, the paper derived that, as the portfolio size increases, the optimal number of inner-loops needed to become smaller using formulas.

Thirdly, the paper provided numerical illustrations of the main results. Distributions for loss Y and pricing error Z are specified to ensure that the bias and variance of the simulation estimators are in closed form. The example allowed the paper to compare the asymptotically optimal (N^*, L^*) to the exact optimal solution under a finite computational budget. This paper used simulation to perform similar exercises on the somewhat more realistic example of a portfolio of equity options. All the conclusions were robust.

Lastly, the paper proposed some refinements to further improve the computational performance of nested simulation: 1. apply simple jackknife methods to the setting to the estimation of large loss probabilities for bias reduction; and 2. apply dynamic allocation of workload in the inner-loop step to further reduce the computational effort in the inner step while increasing the bias by a negligible controlled amount or even decreasing the bias.

Key takeaways

This paper showed that nested simulation of loss distributions poses a much less formidable computational obstacle than it might initially appear. The paper also showed that large errors in pricing at the model horizon can be tolerated as long as the errors are zero mean and mainly idiosyncratic. In the aggregate, such errors have a modest impact on our estimated loss distribution. More formally, this paper quantified that impact in terms of bias and variance of the resulting estimator, and allocated the workload in the simulation algorithm to minimize the mean square error. Simple extensions of the basic nested algorithm can eliminate much of the bias at a modest cost.

The method proposed in this paper can be applied to other problems in finance, such as pricing options on complex derivatives, the rating of CDOs and other structured debt instruments when model parameters are subject to uncertainty.

2.2.6 EFFICIENT RISK ESTIMATION VIA NESTED SEQUENTIAL SIMULATION

Authors: Mark Broadie, Yiping Du, Ciamac C. Moallemi

Issue year: 2010

Accessing the paper

This paper can be accessed through the link below:

[Efficient Risk Estimation via Nested Sequential Simulation](#) (Broadie, et al., 2010)

Paper overview and objective

This paper divided the risk measurement into two stages, scenario generation and portfolio revaluation. To reduce the computational burden of the nested simulation model, this paper proposed a new algorithm that sequentially allocates computational effort in the inner simulation based on marginal changes in the risk estimator in each scenario. This paper sought to exploit the fact that accurate portfolio revaluation is not equally important across all scenarios. Nested simulation can be made much more efficient by allocating computational effort non-uniformly across scenarios. Both the theoretical results and numerical results were shown to support the efficiency of the algorithm.

Approach and analysis

This paper considered the most basic risk measure, the probability that the future portfolio value falls below a pre-specified threshold, a large loss. First, the paper defined the problem and introduced the notations in the nested simulation, then reviewed the results for uniform inner-stage sampling. The uniform estimator is a function of two parameters: the number of scenarios n , and the number of inner-stage samples for each scenario m . The best choices for the parameters m and n are to minimize the mean squared error of the estimate, subject to the constraint of a limited budget of computational resources.

Next, the paper proposed a non-uniform nested simulation algorithm. The algorithm proceeds by allocating the inner-stage samples for portfolio revaluation in a sequential fashion. At each time step, it myopically selects the scenario where one additional inner-stage sample will have the greatest marginal impact on the estimated loss probability. Furthermore, the paper provided analysis that demonstrated, for the same overall number of samples, the non-uniform estimator proposed reduces bias by an order of magnitude. The paper also provided analysis that demonstrated the lower asymptotic MSE of the approach. Since non-uniform sampling provides a lower bias for the same number of inner-stage samples, some of this computational savings can be used for outer scenarios to lower variance.

Numerical results were also illustrated to show the benefits of the non-uniform nested estimation. Two examples were chosen for the numerical analysis. One is the Gaussian cash flow, where both the outer-stage scenarios and inner samples are generated from normal distribution; the other is a put option, where the portfolio consists of a single put option on an underlying asset whose price follows a geometric Brownian motion process. The examples were used to compare the uniform estimator and non-uniform threshold and sequential estimators on the basis of bias, with MSE achieved by various uniform and non-uniform estimators, given a fixed computational budget of k inner-stage samples. The comparison showed that MSE was reduced by factors ranging from four to over 100.

Key takeaways

This paper proposed using a non-uniform nested sequential simulation procedure, which allocates more resources where the effort on the risk estimation is the greatest, allowing more effort to be devoted to the generation of outer scenarios. The algorithm produced a risk estimator that converged to the true value at a faster rate.

For future research, the sequential estimation procedure proposed in this paper can be combined with previous research on variance reduction for the outer-stage scenario generation to achieve further computational savings. The proposed approach can be applied to other risk measures as well.

2.2.7 RISK ESTIMATION VIA REGRESSION

Authors: Mark Broadie, Yiping Du, Ciamac C. Moallemi

Issue year: 2015

Accessing the paper

This paper can be accessed through the link below:

[Risk Estimation via Regression](#) (Broadie, et al., 2015)

Paper overview and objective

To reduce the computational challenges in financial risk measurements, this paper introduced a regression-based nested Monte Carlo simulation method for the estimation of financial risk. An outer simulation level is used to generate financial risk factors and an inner simulation level is used to price securities and compute portfolio losses given risk factor outcomes. The paper used theoretical analysis and numerical results to demonstrate that the proposed regression method can save computational effort compared with other methods.

Approach and analysis

First, this paper proposed a global spatial method based on regression that combines information from different outer-stage scenarios to better approximate the portfolio loss function. The MSE of the regression method converges at the rate k^{-1} until reaching an asymptotic bias level, which depends on the size of the regression error.

Next, theoretical analysis was provided to highlight and quantify the effect of model error.

Thirdly, the paper demonstrated that the proposed regression method was practically implementable by providing numerical results that illustrate the computational savings on a range of examples and compared the performance of the regression method with other methods.

Finally, the paper proposed a weighted variation of the regression method that offered improved asymptotic bias. The paper described and analyzed a weighted regression algorithm for risk estimation. The paper established that the asymptotic bias of this algorithm was determined by the choice of weight function, described an idealized “optimal” choice of weight function, along with a practically implementable variation, and provided numerical results that demonstrated an improvement consistent with theory.

Key takeaways

This paper proposed a new risk estimation method based on Monte Carlo simulation and regression. The optimal choice was k outer-stage scenario and inner-stage samples. Theoretical results show that the mean squared error diminished at a rate close to k^{-1} until a non-diminishing bias level was reached. The proposed regression method outperformed standard nested simulation and the delta-gamma method with a small number of samples and a quadratic basis. In addition, before hitting the bias level, the proposed method recovered the standard k^{-1} convergence rate of non-nested unbiased simulation estimators. Finally, the global regression method can be combined with other local methods and further improved using standard simulation variance reduction techniques.

2.2.8 A COMPARATIVE STUDY OF RISK MEASURES FOR GUARANTEED MINIMUM MATURITY BENEFITS BY A PDE METHOD

Author: Runhuan Feng

Issue year: 2014

Accessing the paper

This paper can be accessed through the link below:

[*A Comparative Study of Risk Measures for Guaranteed Minimum Maturity Benefits by a PDE Method*](#) (Feng, 2014)

Paper overview and objective

This paper distinguished and compared two types of calculations, and used guaranteed minimum maturity benefits to investigate alternative numerical methods to reduce the computational burden but still achieve high accuracy and efficiency. This paper proposed a technique for assessing the overall risk of equity-linking products and provided a new approach to testing and estimating the sampling errors of simulations. Even though this paper focused on guaranteed minimum maturity benefit, the partial differential equation (PDE) method proposed can be extended for more general product designs.

Approach and analysis

First, the paper introduced two models. The first one was the individual model, the Black-Scholes model, to investigate the insurance liability of the GMMB on a standalone contract basis. The second was the average model for current market practice, and only the average contract of a typical size is considered in the reserve calculation for a variable annuities block.

Then, the paper applied PDE to the computation of risk measures for the two models through theoretical analysis. Since using high dimensional PDEs are much harder than low dimensional PDEs, this paper was more interested in a dimension reduction technique. The resulting PDEs from both models were in a form that could be easily handled in most commercially available software packages.

Finally, the paper performed numerical analysis to compare the PDE method against other analytical methods for the individual model. The analysis confirmed that the PDE method was an accurate alternative. For the average model, the advantage of the PDE method is its flexibility. The PDE method was compared with Monte Carlo simulation, and the results showed that the PDE method required at most one-fifth of the computational time with accuracy up to four decimal places.

Key takeaways

For future research opportunities, the paper compared the application of the proposed method on the two models for stochastic mortality. First, the stochastic mortality models cannot be used in an average liability model. Secondly, the comparison confirmed that the individual model was an appealing alternative of risk modeling for guaranteed products, as it is more adaptable for complex mortality structures. The topic can be extended to an aggregate liability model with stochastic mortality rates in future research.

2.2.9 VALUATION OF LARGE VARIABLE ANNUITY PORTFOLIOS UNDER NESTED SIMULATION: A FUNCTIONAL DATA APPROACH

Authors: Guojun Gan, X. Sheldon Lin

Issue year: 2015

Accessing the paper

This paper can be accessed through the link below:

[Valuation of large variable annuity portfolios under nested simulation: A functional data approach](#) (Gan, et al., 2015)

Paper overview and objective

Insurance companies currently use nested simulation for VA portfolio guarantee valuation, but efficient valuation under nested simulation for a large VA portfolio has been a real challenge. The computation is highly intensive and often prohibitive. This paper proposed a novel approach to speed up the nested simulation by reducing the number of VA contracts that go through the nested simulation. This approach combined a clustering technique with a functional data analysis technique to solve the computation burden issue. The paper used a highly non-homogeneous synthetic VA portfolio of 100,000 contracts and estimated the dollar Delta of the portfolio at each time step of the outer-loop scenarios under the nested simulation over 25 years. The results showed that the proposed approach had high accuracy and efficiency.

Approach and analysis

First, the paper gave a brief introduction to the nested simulation of VA contracts and several existing approaches that address computational issues arising from the valuation of large VA portfolios, and why some of them are not solving the problems very well.

Secondly, the paper gave a brief description of functional data analysis and the universal kriging method for functional data (UKFD). Then, the paper applied the universal kriging method to the VA Delta calculation under the nested simulation and performed the numerical tests. This paper used a clustering technique, k-prototypes algorithm, to select a small set of representative VA contracts from a large VA portfolio and calculated quantities of interest of the representative contracts under nested simulation, then applied the universal kriging to calculate those quantities of interest for the remaining VA contracts in the portfolio. The paper illustrated how to use the universal kriging to accurately estimate the dollar Deltas of any contract in the VA portfolio using the representative VA contracts. Next, to evaluate the effectiveness of the proposed method, the paper generated 100,000 synthetic VA contracts with GMDB and GMWB riders and used the proposed method to calculate the dollar Deltas at each time step in the real-world scenarios. The paper performed three test cases with a different number of representative contracts. All the test results showed that the method was able to effectively approximate the annual dollar Deltas along the

outer-loop real world scenarios. The result showed that a very small subset, such as 1%, of VA contracts in a VA portfolio might be used as representative contracts to achieve fairly satisfactory accuracy. When the number of representative contracts increases, the absolute differences between the annual dollar Deltas estimated by the UKFD method and those calculated by the simulation model decreased.

Key takeaways

This paper proposed a method based on functional data analysis to estimate the annual dollar Deltas along outer-loop scenarios under the nested simulation framework. It used the k-prototypes algorithm to select representative VA contracts from a large VA portfolio and the universal kriging for functional data to estimate the Delta along outer-loop scenarios for all contracts in the VA portfolio. Since the time-consuming nested simulation is only applied to a small number of representatives and the UKFD method is fast, the proposed method can reduce the computation time significantly. The proposed method can also be applied to estimate other quantities of interest, such as dollar Pho, cash flows and risk measures.

All the components of the proposed method can be changed or modified for future research. One direction could be to explore other methods for selecting representative VA contracts. Another direction could be to test the proposed method under a realistic nested simulation model.

Section 3: Practical Application of AIML for Actuarial Modeling

With [section 1](#) serving either as a primer or a refresher depending on the reader's prior experience with AIML and [section 2](#) providing a good overview of recent relevant research to this paper, we are now ready to start exploring practical applications of AIML for actuarial modeling with a focus on nested stochastic.

This section assumes that the reader is now familiar with the AIML concepts introduced in [section 1](#) and builds upon this knowledge to explore how AIML can be inserted into actuarial modeling methodologies to reduce challenges with running models. However, we do not assume that the reader understands the concepts covered in [section 2](#). Literature reviews, especially the academic papers found in this section, will cover similar concepts but in a simpler fashion for practitioners looking to get started.

We have structured the section as follows:

- We first provide a high-level overview of how AIML can be used to address runtime challenges and resulting applications in subsection [3.1 AIML Actuarial Modeling Applications](#).
- We then explore the end-to-end approach that readers can follow when developing such AIML models for actuarial modeling applications in subsection [3.2 Approach overview](#). We will build upon this approach for the case studies that follow later in this paper.

Lastly, as we mentioned at the beginning of [section 1](#), certain aspects of the methodology used for AIML proxy models may differ from the typical methodology to develop machine learning. We will make specific mentions when this is the case.

3.1 AIML ACTUARIAL MODELING APPLICATIONS

In this section, we explore methodologies such as clustering in [3.1.1 Clustering](#), the use of proxy models in [3.1.2 Proxy models](#), algorithms to identify subsets of representative scenarios to run in [3.1.3 Scenario selection](#) and hybrid methodologies in [3.1.4 Hybrids](#).

3.1.1 CLUSTERING

Overview

Clustering, within the life and annuity actuarial modeling field, is a technique used to reduce the overall number of model points by grouping similar points together, assuming they would produce similar results. This is also often referred to as compression or in-force compression.

There is a wide range of industry practices used by actuaries and other fields that use clustering. Clustering reduces computation complexities by finding representative points of homogenous attributes. When applying it into actuarial modeling, subject matter expertise for making appropriate assumptions of what constitutes similarity on certain attributes, and the desired resulting number of clusters, are key considerations to balance the runtime improvement versus result precision.

A traditional methodology we have seen actuaries use to cluster is to create groupings of policies based on similar in-force characteristics, such as grouping by a combination of attributes like attained age, policy year, account value, etc.

AIML can be used to approach clustering more efficiently by letting the algorithm optimize the groupings to provide the most adequate coverage across the block. This includes grouping policies based on the target output. Unsupervised machine learning techniques, such as Density Based Spatial Clustering, account for the data points within constraint from the cluster center. A cluster center would be the dense region with

the most similarities, while a lower density region indicates differences in attributes. Another method could be hierarchical based, where clusters are formed using a tree type structure based on the hierarchy. The hierarch of the tree can be structured either bottom-up, which is referred to as the agglomerative method, or bottom-down, which is referred to as the divisive method.

Users may frame the problem such that the algorithm groups policies not only based on similar characteristics such as attained age, policy age and face amount, but also based on present values, reserves or even cash flows.

Clustering can greatly reduce the runtime as it can compress a large in-force file or a granular set of new business cells into a relatively small set of clustered records. Each clustered record is typically assigned a weight and, when weighted together, the aggregate result across the clustered records provides a close approximation for the result that would have come from the entire in-force records or the entire new business cells.

When may actuaries find this approach useful?

Clustering is most efficient with applications that have a large number of model points (in-force policies or new business records) to process. Clustering can be very efficient for large blocks of business. An example is using a simple and common algorithm such as K-means to partition all model points into k clusters, where each observation belongs to the cluster with the nearest mean serving as a representative model point of the group of policies. Choosing K can be directly related to the reduction of the computation budget desired.

For actuarial models that are deployed in a Cloud-based environment that support distributed or parallel computing, clustering can also help efficiently process those large blocks to the cloud, especially when distribution by scenario strategies would be efficient if there were fewer model points.

Word of caution

Actuaries using clustering techniques, whether based on AIML or not, need to be careful to use the clustered records when it is appropriate to do so. For instance, a clustered in-force designed to provide a good approximation for statutory reserves may not perform adequately for GAAP projections. It is the responsibility of the actuary to make sure the clustering process is well tested and that controls are in place so that it is used for the right purposes.

Paper case studies

We do not explore clustering as a technique in the case studies offered in this paper. Additional consideration of what these can be applied to actuaries, and how they can be used for the purpose of principle-based approaches to reserves and capital are very well discussed by practice notes (Beeson, et al., 2010) published by the American Academy of Actuaries Modeling Efficiency Working Group, including aspects of validating the clustered results.

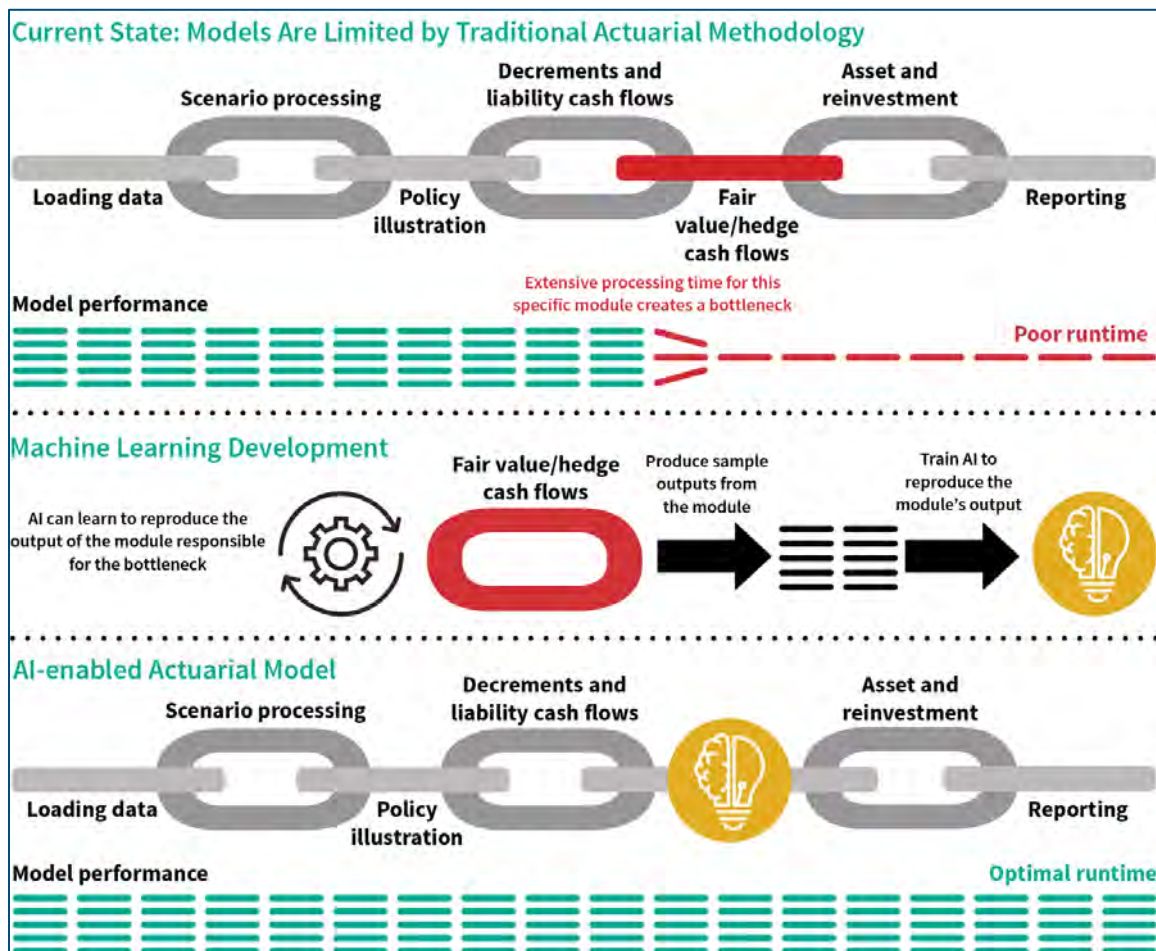
3.1.2 PROXY MODELS

Overview

As discussed previously, AIML can be used as high-fidelity proxy models to replicate actuarial calculations. This can be used strategically by actuaries to replicate certain calculations that are onerous, such as nested stochastic.

The diagram below illustrates this process:

Figure 9
ILLUSTRATION OF AIML USE IN ACTUARIAL MODELING (CZERNICKI, 2020)



In the *current state*, actuaries can use a full nested stochastic approach where the fair value and Greek calculation is based on stochastic risk-neutral valuation where the model branches off at the pivot points along the projection to calculate these measures for each policy.

While the rest of the actuarial process runs efficiently, the nested stochastic calculation proves to be a bottleneck and slows down the overall process, resulting in long model runtime.

In *machine learning development*, actuaries can use the component that is responsible for the poor runtime and produce a database of sample input-output combinations. This is done by running the actuarial calculations through multiple combination of inputs that provide reasonable coverage of the cases

the model may run into. Actuaries may then use this database to calibrate a regression AIML model, such as XG Boost or neural networks, to replicate the output given the inputs.

In an *AI-enabled actuarial model*, the calculation of fair value and Greeks can then be replaced with the AIML model, eliminating the runtime bottleneck.

When may actuaries find this approach useful?

Proxy models are particularly useful when a runtime intensive calculation needs to be run frequently. Intensive calculations often involve, but are not limited to, nested stochastic calculations.

Actuaries commonly run into this challenge when projecting reserves and capital in a pricing or forecasting setting.

Common examples include:

1. Projection of hedge income that requires the projection of fair value or Greeks (e.g., variable annuities)
2. Projection of US Statutory reserves and capital (e.g., UL principle-based reserves)
3. Projection of US GAAP reserves
4. Projection of certain IFRS17 reserves and Solvency II requirements
5. Projection of Bermuda reserves

Word of caution

Actuaries using this approach need to be careful to understand how closely the model proxies the first-principles actuarial calculations. In particular, actuaries need to take into consideration the range of inputs that were used to train the model and make sure to not use the model outside of those bounds.

Failure to control appropriately for the usage of the AIML model could result in a potentially wide inaccurate proxy. This could cause serious issues for the actuary depending on the nature of the application.

Recalibration and continuing performance monitoring with the appropriate model risk governance would be applicable for proxy models as well. Reader can refer to [section 1](#) for details on the Machine Learning Life cycle and operation of these applications. Scenarios which could cause fundamental dynamics, such as policyholder behavior in the block of business, or market environment to significantly deviate from the dynamics when the model was first established. These are often tell-tale signs that recalibration of the proxy model is needed.

Paper case studies

This application will be explored in the first and second case studies and will be a focus area of the paper.

3.1.3 SCENARIO SELECTION

Overview

Many actuarial applications require running a large number of stochastic scenarios as part of a fair value, reserve/capital calculation, profitability/risk analysis or other applications.

AIML can be used to attribute relative importance to the underlying scenarios and identify a subset that can be used as a proxy for the full set. In 2.2.2, the literature review of the paper, *Efficient Nested Simulation for Conditional Tail Expectation of Variable Annuities*, discussed the importance of the allocation of computation budget to inner-loop scenarios that are most likely to result in tail reserves, which is a good example of utilizing scenario selection.

Note that AIML is one of many techniques that can be used to perform scenario selection. Many actuaries may be currently using scenario selection techniques that do not involve AIML.

When may actuaries find this approach useful?

Actuaries may find this approach particularly useful for aggregate tail-based measures such as US life PBR stochastic reserves and VM-21 reserves where clustering alone would not resolve the runtime concerns, or may not be an option, and where the overall calculation is too runtime intensive to produce sufficient data to train a proxy model for the end result of the calculation.

Word of caution

Similar considerations apply here as to the proxy models where the AIML algorithm that recommends the most important scenarios may not perform well outside of the coverage used for training the model.

3.1.4 HYBRIDS

Actuaries could consider combining the methodologies above. While we will not explore this in this paper, this technique can be effective in solving the runtime challenge incrementally.

We can explore a hypothetical situation to showcase how this might be applied for in-force and new business variable annuity capital forecasting:

- First, start by developing a clustering model to consolidate the in-force and new business model points into a smaller compressed set of model points. This is developed by compressing the policies based on the present value of cash flows over scenarios from the American Academy of Actuaries. This will help reduce the runtime with a limited loss of precision.
- Second, develop an AIML model to recommend the tail scenario selection. This is developed based on the clustered in-force and over a large number of examples.
- Third, the combined compressed model points with the tail-scenario selection are used to produce many examples of capital results over various projected economic environments. The resulting data is used to develop a direct proxy. This was achievable given the combination of the first and second models reduced the runtime enough to produce a sizable database to train the direct proxy model.
- Now, the final proxy model can be used directly to proxy the capital.

3.2 APPROACH OVERVIEW

We will use a seven-step methodology to develop the AIML models in this paper:

1. **Problem statement, planning and business case:** Defining the objectives of the AIML model and establishing clear requirements and success criteria to evaluate the AIML model. The actuary should get a clear understanding of key considerations such as, but not necessarily limited to, the applications that will eventually use the AIML model, the range of economic environments that the model will be used on, the expectations for the precision of the proxy and how frequently the proxy is expected to be recalibrated.
2. **Environment preparation:** In this step, the actuary would prepare the AIML modeling environment and use common packages to develop machine learning and other packages that provide useful functionality.
3. **Data generation:** In this step, the actuary would prepare the actuarial model to create the training data needed to develop the AIML model. The actuary first has to evaluate the methodology used to generate the data such that the AIML model can be trained to meet the objectives that were established in the first step. Key considerations include how to vary the economic environment and policyholder information effectively. Note that well-sampled training data is one of the most important elements to develop an effective AIML model.
4. **Feature engineering and selection:** Develop additional inputs for the AIML model derived from the original Monte Carlo valuation inputs. Perform data exploration and feature importance analysis based on data generation with added feature engineering to identify which features should be used as inputs to the AIML model.
5. **Model testing and selection:** Test performance of various AIML models. Test and control for overfitting. Select the top model based on performance and judgment.
6. **Hyperparameter tuning:** Test alternative hyperparameters for the selected model to refine performance. When needed, hyperparameter tuning can improve the performance of a given model.
7. **Actuarial evaluation:** Evaluate the performance of the model from an actuarial perspective. This includes applying the model in its final use case and comparing it against the first-principles calculations. It is also recommended to test boundary cases to see how the model performs in tail or more extreme events that are in scope for its eventual use.

The development of the model will generally be done in an iterative fashion where the actuary may have to go back to previous steps depending on the learnings achieved as the actuary progresses through the seven steps above.

3.2.1 PROBLEM STATEMENT, BUSINESS CASE AND PLANNING

We recommend that actuaries first start by determining the root cause of the poor actuarial model runtime and evaluate if AIML might be the right solution to address the issue.

Actuaries should think about whether clustering ([3.1.1 Clustering](#)), a proxy model ([3.1.2 Proxy models](#)), scenario selection ([3.1.3 Scenario selection](#)) or a combination or hybrid model ([3.1.4 Hybrids](#)) would be most adequate to solve the runtime issue.

Actuaries should perform a cost-benefit analysis to determine whether the proposed approach would provide value to confirm if there is a business case.

When evaluating the cost, actuaries should consider:

1. Resource time needed to research, develop and implement the solution
2. Increased complexity on processes
3. Time and cost to generate sample data (if applicable)
4. Opportunity cost

When evaluating the benefits, actuaries should not only evaluate the runtime saved, but also the intrinsic value of being able to generate faster analysis.

Actuaries will also gain a better sense of the post development costs as they progress through the development of the AIML model and the business case should be refined and validated prior to implementation.

In planning the work, actuaries should consider:

1. Any coordination with other groups (e.g., data science)
2. How to communicate the results effectively with the ultimate model users
3. Activities, timeline and dependencies

3.2.2 ENVIRONMENT PREPARATION

While the AIML model could be developed on an individual's laptop, actuaries should consider using the cloud for more efficient collaboration and processing power.

Many companies have a cloud environment for AIML model development and actuaries may be able to leverage this for development of their own models.

Python and R are the most commonly used open-source software for machine learning development and both offer extensive open-source libraries with functionality to develop machine learning models.

3.2.3 DATA GENERATION

Many AIML applications detailed in this paper will require some form of data generation. By data generation, we refer to the activity of running actuarial models to produce sample outputs.

For instance, precise AIML proxy models require a large number of sample input-output combinations generated from the actuarial model.

In our experience, it is critical to apply actuarial judgment in designing the process to generate the data to train and test the AIML model.

In particular, most applications will require varying a combination of economic environments and coverage of the characteristics of the liabilities. This concept will be explored in detail in each of the case studies in this paper.

3.2.4 FEATURE ENGINEERING AND SELECTION

As opposed to actuarial models where all detailed information on the policyholder, yield curve, and other elements are needed to initiate a calculation, AIML models do not necessarily need all the fundamental inputs to perform in its intended design. This was explored in [1.3.3 Data transformation](#) with details on feature engineering and selection.

Further, the performance of AIML models may be significantly enhanced through the process of feature engineering. This process consists of deriving alternative potential inputs from the other information that is available in a way that provides greater predictive power to the models.

For instance, engineered features that have a close relationship to the target variable may help improve the model performance significantly.

Actuarial judgment or intuition can again be critical in the engineering of features as actuaries may have a sense of how to combine the original inputs into metrics that are more closely tied to the target variable. Again, this is something will be explored with concrete examples in the case studies.

3.2.5 MODEL TESTING AND SELECTION

Model testing and selection for these applications follow the same principles as in [section 1](#).

The major difference for proxy models is that they are designed to achieve very high R^2 metrics (generally above 99.5%), which would generally be attributed to overfitting in traditional data science applications.

3.2.6 HYPERPARAMETER TUNING

Hyperparameter tuning can help further improve the performance of the model. However, in our experience, this is generally a final step to make minor additional refinements.

The previous steps, in particular data generation, feature selection and engineering and model selection, are more critical.

[Section 1](#) provides more information on hyperparameter tuning.

3.2.7 ACTUARIAL EVALUATION

Evaluating the performance of the model from an actuarial perspective is critical. A model may perform well on the original hold-out data, but may not perform well when applied in an actual application.

Actuaries should test the model in real-life setting before formally implementing the model and using it in production settings.

For instance, if an AIML model proxy was designed to replicate a reserve or other stochastic calculation along a projection, actuaries should compare projection results with the AIML proxy against first-principles calculations. Actuaries can then analyze how the proxy fares against the true results in various types of economic environments and other situations.

Section 4: Professional, Fairness and Regulatory Considerations

Besides the technical aspects of AIML, including its applications for nested stochastic modeling introduced so far in this paper, there are a few other aspects that are important for practitioners to consider.

In particular, as a nascent field, data scientists are not a defined profession with a structured authority that could articulate standards of practice, codes of conduct and CPD requirements.

One key area that regulators and professionals alike are focusing on is the concept of fairness where there are extensive efforts and discussions on detecting and controlling for undue bias in AIML models, especially when those models influence or dictate decisions affecting individuals.

While this paper’s primary objective is to explore the technical aspects of AIML to solve life and annuity nested stochastic modeling challenges, we dedicate this section to:

1. Explore professional considerations through the Actuarial Standards of Practice (ASOPs) – see subsection [4.1 ASOP Review](#).
2. Provide an overview of the discussions on bias and fairness and its applicability to the AIML models in this paper – see subsection [4.2 Bias and Fairness](#).
3. Review recent public developments in regulations that may affect the applications discussed in this paper – see subsection [4.3 – Regulatory developments](#).

4.1. ASOP REVIEW

The Actuarial Standards of Practice, commonly abbreviated as ASOPs, are defined and maintained by the Actuarial Standard Board (ASB). As defined by the ASB:

“The Actuarial Standards Board (ASB) sets standards for appropriate actuarial practice in the United States through the development and promulgation of Actuarial Standards of Practice (ASOPs). These ASOPs describe the procedures an actuary should follow when performing actuarial services and identify what the actuary should disclose when communicating the results of those services.”

Actuaries who are looking to apply the concepts in this paper should extend the applicability of the ASOPs to these new concepts. In this section, we provide commentary on which ASOPs we found to be relevant this paper.

All ASOPs are provided on the ASB website.

[Appendix C: ASOP review](#) provides commentary on the applicability of certain ASOPs that may be relevant to actuaries looking to apply AIML for stochastic modeling.

4.2. BIAS AND FAIRNESS

As mentioned at the onset of this section, fairness is an important topic when it comes to AIML. Without the proper controls, AIML models can easily introduce undue bias that can be unfair resulting in unethical treatment of individuals for certain applications. Examples include credit or lending decisions based on AIML, AI chatbot, voice or facial recognition systems, and search and advertising placement using recommender algorithms.

Now, bias and fairness are less of a concern when it comes to the proxy modeling applications detailed in this paper. This is due to the intended purpose of the AIML proxy models to replicate the results of an actuarial model. However, actuaries using AIML in any capacity should always keep in mind the important discussions happening globally on the bias and fairness of actuarial models.

In particular, actuaries should be careful that the error introduced by the AIML proxy model is not in itself biased and introducing unfair consequences to individuals (e.g., bias or error in a pricing application affecting the price of insurance).

Regulators, such as the National Association of Insurance Commissioners (NAIC), are working towards providing guiding principles for AI adopters that address questions surrounding the use of big data, modeling, and artificial intelligence. The Financial Stability Board (FSB) published similar guidance, where they stress the importance of assessing ethical uses of AI and machine learning with an understanding of their risks, including adherence to relevant protocols on data privacy, conduct risks, and cybersecurity. Adequate testing and 'training' of tools with unbiased data and feedback mechanisms are important to ensure machine learning applications do what they are intended to do.

On the ethical and societal aspects, the European Commission has published extensive Ethics Guidelines for Trustworthy AI. The guidelines put forward a set of seven key requirements that AI systems should meet to be deemed trustworthy. A specific assessment list aims to help verify the application of each of the key requirements. The SOA published a report on the "Ethical Use of AI for Actuaries," that covers AI, the social context, and the five pillars of ethical AI: responsibility, transparency, predictability, auditability, and incorruptibility.

4.3. REGULATORY DEVELOPMENTS

As innovation and adoption in AIML continues to evolve, so do regulations. At the time this paper was written, regulators in the US had been focusing on addressing the most common applications of AIML in the insurance industry, which included underwriting and claims. The laws and regulations relating to the use of artificial intelligence and machine learning (AIML) vary by country and region. Here are some examples of global regulatory developments:

1. European Union: In April 2021, the EU released a set of regulations called the "Artificial Intelligence Act," which outlines rules for the use of AI systems. These rules apply to both public and private entities and cover four categories of AI systems: unacceptable risk, high risk, limited risk, and minimal risk.
2. Canada: In 2018, Canada released a set of guidelines called the "Ethical AI Framework," which provides guidance for the development and deployment of AI systems. The guidelines emphasize transparency, accountability, and respect for human rights and privacy.
3. United Kingdom: In 2021, the UK released a set of guidelines called the "AI Code," which provides guidance for the development and deployment of AI systems. The code emphasizes transparency, accountability, and the ethical use of data.

For the United States: Currently, there is no federal law that specifically regulates the use of AIML. However, there are several laws that govern data privacy, cybersecurity, and discrimination that may apply to AIML systems. Additionally, several states, including California and New York, have passed their own laws relating to AI and data privacy. At the time this paper was written, there were several legislative actions that would apply to the result of pricing work. Several bills have been either introduced or passed in states that are particularly focused on ensuring the avoidance of proxy discrimination in the sales and

pricing of insurance. The details of these are beyond the scope of this paper but should be kept in the mind of practitioners. There are also existing actuarial standards boards that have provided guidance on how predictive models should be tested and reviewed in ASOP 56. Actuarial, accounting, and regulatory guidance contain general principles requiring the actuary to certify the appropriateness of the models used. As such, actuaries should remain vigilant in following the spirit of the regulations, ensuring that the predictive models used are fit for purpose, reflective of the true risks facing the insurance company, contain risk margins for conservatism appropriate for the purpose, and use professional judgment in their training, deployment, and use.

Section 5: Case Study 1 – Index Crediting

5.1. INTRODUCTION

In this section, we explore the use of AIML to proxy the market value of non-traditional derivatives in the context of indexed-based products (e.g., FIA, RILA and IUL). The market value of these options is used to set the crediting parameters such that the cost of the option can be funded by the return on the supporting assets and provide an acceptable interest margin for the insurer.

Calculating the market value of non-traditional options often requires Monte Carlo or fair valuation (we will use the term Monte Carlo valuation in this section).

This use case was selected because it serves as a practical and gentle introduction to applying the AIML and actuarial concepts we have covered in the prior sections.

Background and product feature overview

Sales of indexed products, such as fixed indexed annuities, indexed universal life and registered indexed-linked annuities, have increased over the last few years and now represent a significant portion of the sales within the life and annuity insurance industry (Windsor, 2022).

Indexed products generally credit interest rates at regular intervals based on the performance of the underlying index over the index crediting period.

Typical index crediting strategies

Indexed products that are currently sold in the market generally use simple or vanilla strategies based on one or a combination of:

- **Cap:** Interest crediting is based on equity performance up to a pre-defined maximum rate (e.g., full participation in a given equity index up to X% and subject to a floor of 0% preventing losses).
- **Participation:** Interest crediting based on a proportion of the equity performance (e.g., partial participation of Y% into the equity index, with no cap, and subject to a floor of 0% preventing losses).
- **Spread:** Interest crediting based on equity performance above a spread (e.g., full participation in the equity index for any growth above Z%).

The most common strategies are based on point-to-point equity performance (e.g., equity performance over a one-year term) but strategies can also be based on average monthly returns, high equity point during the term and many other structures.

These indexed strategies can generally be replicated by combining European equity options (Hull, 2021) (vanilla point-to-point derivatives) and actuaries would typically use Black-Scholes (Black, et al., 1973) or other closed-form formulas to calculate the market value of the index crediting strategy based on the equity options that replicate the strategy. They may also employ these formulas to calculate the Greeks depending on the nature of the index crediting hedging strategy.

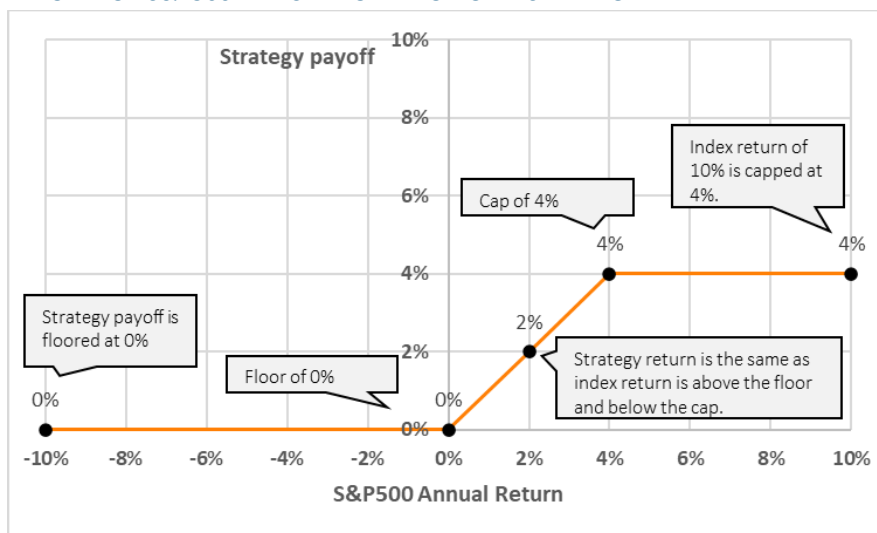
Calculating the market value and other related metrics (e.g., Delta, Gamma, Rho) are important for risk management applications, such as hedging, and for financial reporting purposes, such as producing valuation results or forecasting Statutory and GAAP balance sheet in the United States.

Below we illustrate the crediting process for an annual point-to-point cap strategy based on the S&P 500 index where the cap rate is set at 4% for this crediting term. Note that the cap rate (or other applicable crediting parameter) is generally set at the discretion of the insurance company, subject to contractually guaranteed floors, and is generally driven by the performance of the insurance company's underlying portfolio and market value of replicating options for the strategy.

The insurance company sets the cap rate at the beginning of the crediting term and the cap rate is hereafter locked in until the end of the crediting term.

The figure below captures the interest credited to the policyholder at the end of the year based on the performance of the S&P index.

Figure 10
PAYOFF FOR S&P500 ANNUAL POINT-TO-POINT STRATEGY



Non-traditional index crediting strategies

While most FIA, IUL and RILA products currently offered by life and annuity insurers are based on point-to-point with a cap, participation and/or spread, some insurers are starting to offer unique or exotic index crediting strategies to differentiate their products from the industry.

This differentiation generally comes in the following forms:

1. **Use of proprietary indices:** While many products will be based on common indices, such as the S&P500, NASDAQ and EAFE, some insurers are starting to offer new indices to manage volatility or make the indexed product more attractive.
2. **Exotic crediting terms:** Typical index crediting strategies are generally based on a combination of European options. Non-traditional crediting strategies may instead use exotic options such as Asian or lookback options.

Although the use of proprietary indices and unique crediting terms are the most common forms of differentiation, other means of differentiation may exist or surface in the future.

Valuation challenge for non-traditional index crediting strategies

The payoff of typical European index crediting strategies, such as the one provided in figure 10 above, can be replicated by purchasing an at-the-money European call option and selling an out-of-the-money call

option with a strike price that is set above the current index level based on the cap rate (4% in the example above).

In this case, the market or fair value of this index crediting strategy would be calculated as the market value of the underlying index options that can replicate the strategy (the net market value of the two options detailed above). This calculation can be performed using Black-Scholes and would not require a stochastic or Monte Carlo valuation. The same goes for the calculation of Greeks.

However, Black-Scholes does not apply to non-traditional index crediting strategies.

As opposed to typical index crediting strategies, there might not exist a precise or accurate closed form solution to evaluate the market value and the industry must often resort to Monte Carlo valuation. However, this practice is often prohibitive due to runtime concerns, especially for actuarial applications that require revaluating the market value of the strategies throughout actuarial projections or actuarial valuations (nested stochastic).

Monte Carlo valuation of index crediting strategies can even result in three layers of stochastic where:

1. **First layer:** The actuaries are performing stochastic real-world outer stochastic projections (first layer).
2. **Second layer:** The balance sheet of the company is projected along those outer projections and financial reporting standards dictate the use of stochastic valuation (e.g., VM-21). In addition, the Monte Carlo valuation methodology is required in this layer for the index crediting strategy.
3. **Third layer:** The inner-loop calculations of the balance sheet component (e.g., VM-21) require projection of the cash flow components, which require the projection and valuation of the index crediting strategy (Monte Carlo).

In the example above, the computing requirements for such products can compound and get out of hand very quickly.

Note that various closed form formulas (either exact or approximative) have been developed by researchers for various non-European options and can be applied to non-traditional index crediting strategies. We encourage actuaries to consider those formulas before evaluating the use of Monte Carlo valuation or the AIML proxy modeling methodology discussed in this paper.

5.2. PRODUCT AND ACTUARIAL METHODOLOGY SPECIFICATIONS

Product features

In this section, we explore a non-traditional index crediting strategy based on a volatility control index targeting a constant 12% volatility with crediting terms defined as a three-year lookback strategy subject to a floor and cap.

The interest credited to the account at the end of the three-year crediting term is calculated as follows:

$$\text{Interest Credited} = \left[\text{Min} \left(\frac{S_{Max}}{S_0} - 1, \text{cap} \right) \right]_{0+}$$

Where S_{Max} is the maximum level the volatility control index reached at the closing of any calendar month-end throughout the three-year index crediting period, S_0 is the index level at the time of reset and cap is the cap rate offered by the insurance company at the time of reset.

The key difference with this strategy relative to typical index strategies is the introduction of a volatility control index instead of a standard equity index and the lookback option. However, the lookback option is the product feature that prevents us from using Black-Scholes and requires the use of Monte Carlo valuation.

Valuation methodology

This case study uses Monte Carlo simulation to calculate the market value of the option. The Monte Carlo simulation used in this case study is based on stochastic equity paths following a log-normal model and fixed (deterministic) interest rates.

The equity index process under the log-normal model is calculated using the following diffusion process:

$$dS_t = r \cdot S_t dt + \sqrt{v} \cdot S_t dW_t$$

Where S_t is the current index level, r is the current risk-free interest rate (fixed deterministic interest rate), v is the index volatility (fixed at 12% based on the underlying volatility control index described in product terms) and W_t is the geometric Brownian process.

The market value is calculated as the average present value of the interest credited or payoff of the strategy across all simulated scenarios.

5.3. AIML MODEL DEVELOPMENT

Overview and problem statement

This section details the methodology used to calibrate and test an AIML proxy model for Monte Carlo valuation of the non-traditional option detailed in [subsection 5.2](#).

The following table summarizes the steps used for this case study and associated references from the subsections below.

Table 1
STEPS AND REFERENCES FOR THE CASE STUDY

#	Step	Paper Section	Description
1	Overview and problem statement	This section	Perform the valuation of the non-traditional option detailed in subsection 5.2
2	Preparation	5.3.1 Preparation	Prepare the environment, load external packages and define the various functions that will support the steps below.
3	Data generation	5.3.2 Data Generation	Define methodology to generate sample results to train and test the AIML model by varying the Monte Carlo valuation inputs.
4	Feature engineering and feature selection	5.3.3. Feature engineering and feature selection	Develop additional inputs for the AIML model derived from the original Monte Carlo valuation inputs. Perform data exploration and feature importance analysis based on data generation with added feature engineering to identify which features should be used as inputs to the AIML model.
5	Model testing and selection	5.3.4. Model testing and selection	Test performance of various AIML models. Test and control for overfitting. Select the top model based on performance and judgment.
6	Hyperparameter tuning	N/A <i>Given the simplicity of this use case, hyperparameter tuning was not performed for this initial case study.</i>	Test alternative hyperparameters for the selected model to refine performance. When needed, hyperparameter tuning can improve the performance of a given model.
7	Actuarial evaluation	5.3.5. Actuarial evaluation	Evaluate the performance of the model from an actuarial perspective.

The methodology to develop such AIML models generally requires iterating among the steps outlined above. In this section, we focus on the final methodology and analysis that was developed by the researchers.

5.3.1. PREPARATION

The first step consists in preparing the environment by loading any external packages, defining the user functions and user inputs.

Python was used to generate the training and testing data, develop the AIML model, and produce the analysis provided in this section. We used the following Python packages:

- **NumPy:** Adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- **Pandas:** Used for data transformation and analysis.
- **Matplotlib, Sweetviz and Seaborn:** Used to produce various visualizations.
- **Scikit-learn (sklearn) and Keras:** Both of these packages provide a library of machine learning models.

5.3.2. DATA GENERATION

The data generation process for this use case was performed by randomizing the features driving change in the market value of the exotic index crediting strategy. Each random combination constitutes an observation for training the AIML model.

The following table summarizes the inputs to the Monte Carlo valuation and provides details on how each input was randomized. The data generation joined random combinations of possible input values at reset and possible input values throughout the crediting term.

Table 2
INPUTS TO THE MONTE CARLO VALUATION

Input	Description	Low Bound	High Bound
Initial index level	Index level at issue or last reset of the index crediting strategy. This variable is fixed at 1,000 and all variables below are normalized based on an initial index level of 1,000.	Fixed at 1,000	
Current index level	Index level as of the date of the Monte Carlo valuation.	At issue or reset Fixed at 1,000 Between resets $8,000 \times (1 - 0.2 \times T)$ <i>T is the time since last reset</i>	At issue or reset Fixed at 1,000 Between resets $12,000 \times (1 + 0.2 \times T)$ <i>T is the time since last reset</i>
Maximum index level	Maximum index level reached at month-end since the last reset of the index crediting strategy	At issue or reset Fixed at 1,000 Between resets 1,000 <i>Set to 1000 as the maximum index level cannot be lower than the index level at any time since the reset</i>	At issue or reset Fixed at 1,000 Between resets 1.2x Current index level
Time to maturity	Remaining time (in years) until the index crediting strategy matures and the resulting interest is credited to the account.	At issue or reset Fixed at 3 Between resets 1/12	At issue or reset Fixed at 3 Between resets 3-1/12
Cap rate	Maximum interest that can be credited to the account (see subsections 5.1 and 5.2 for further details on the cap rate)	10%	30%
Risk-free rate	Current risk-free interest rate used in Monte Carlo valuation	0.01%	10%
Volatility	Volatility used in Monte Carlo valuation. The volatility is fixed at 12% given the underlying is a control volatility index that manages volatility to 12%.	Fixed at 12%	

Note that unless stated otherwise, all randomized input were generated independently of one other. A total of 26,000 observations were generated with 30% of those being at reset and 70% in-between resets.

The number of observations generated was informed by monitoring incremental model performance as models were calibrated with increasingly more data. The 30/70 split was chosen to gain greater accuracy at reset to limit the proxy error on setting the cap rate given that the cap rate is locked for the duration of the crediting period.

There is inherent variance in the calculated market value of the index crediting strategy as Monte Carlo valuation is based on simulated random paths. In theory, a fully accurate result (no variance) would be achieved with an infinite number of random paths (central limit theorem (Gnedenko, 1954)). However, in practice, the number of random paths that can be generated is limited by the computing resources available, timeline to produce the results, and potential cost considerations (e.g., if running on the cloud, if more powerful hardware is needed, etc.)

Variance reduction techniques (Pedersen, et al., 2016) can be used to reduce the variance for a given number of random paths used and can be a useful strategy to reduce errors in the samples generated to train and validate the AIML model.

In this case, antithetic variates (Botev, et al., 2017) are used. This technique reduces the variance by producing an opposite of each path used. For instance, if N paths are used (N is even) where the random numbers are based on uniform distributions:

- The first half of the paths generated (1 to N/2) are generated randomly based on the uniform distribution. The first half of the paths will be $u_1, u_2, u_3, \dots, u_{N/2}$ where $u_i \sim U(0,1)$.
- The second half of the paths (N/2+1 to N) are derived directly from the first half. The second half of the paths will be $u_{N/2+1}, u_{N/2+2}, u_{N/2+3}, \dots, u_N$ where $u_{N/2+i} = 1 - u_i$.

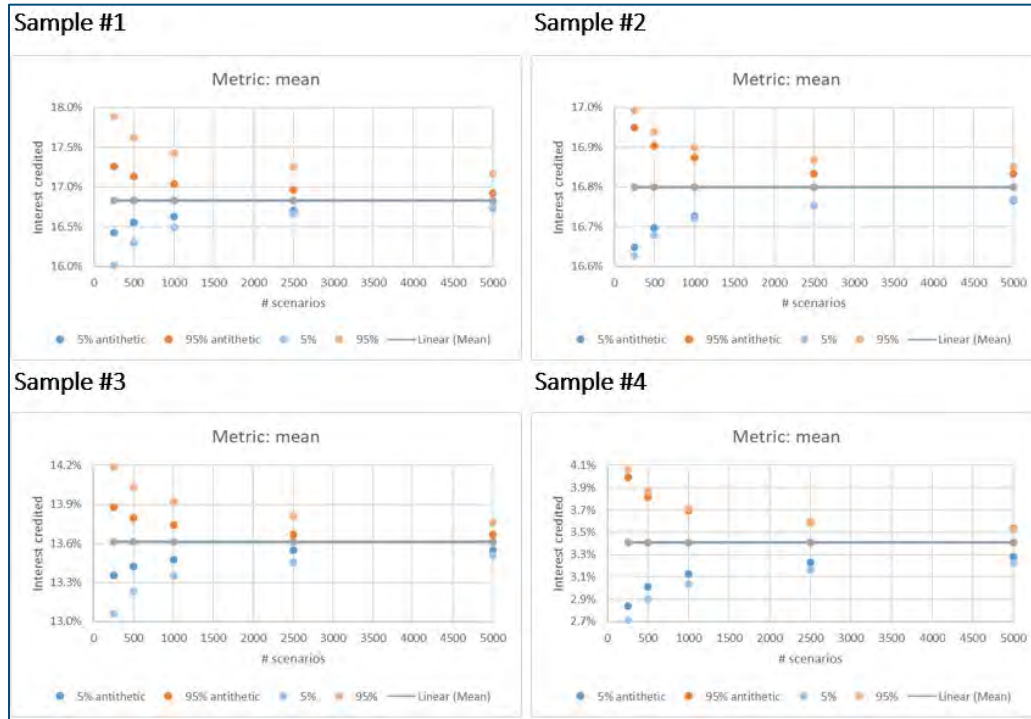
Then, bootstrapping (Efron, et al., 1954) was used to quantify the errors for various sample sizes (N). This test was done over four sample cases where Monte Carlo valuation of the exotic index strategy was performed.

Table 3
SUMMARY OF THE FOUR SAMPLE CASES

Sample #	Time to Maturity	Initial Index	Current Index	Max Index to Date	3-year Cap	Risk-free Rate
1	3 years <i>New business</i>	1,000	1,000	1,000	30%	5%
2	1.5 years <i>In-force</i>	1,000	1,100	1,150	20%	8%
3	1.5 years <i>In-force</i>	1,000	1,050	1,050	20%	3%
4	2.0 years <i>In-force</i>	1,000	850	1,000	30%	3%

The variance for each of the four above test cases is illustrated below. The variance is measured through the 5% to 95% confidence interval for increasing number of Monte Carlo simulations.

Figure 11
MONTE CARLO VARIANCE



The dark dots provide the confidence interval with antithetic, and the light dots provide the same confidence interval without antithetic. The grey line was calculated using 10,000 scenarios and used to approximate fully converged results. The scenario samples were randomly selected from a set of 10,000 based on the bootstrapping technique (subset selection of the 10,000 with replacement).

The authors opted to use 10,000 scenarios for the training data and 10,000 scenarios for the testing data. Unless computing resources allow running a sufficient number of inner-loop scenarios to reach high convergence, the number of scenarios used for training should balance the processing time to complete the generation of training data and the precision of each individual observation.

This can generally be achieved through trial and error with the objective of optimizing the model fit for given computing resources or budget allocated to data generation. Testing data, on the other hand, should seek to minimize the errors such that any difference between the AIML model results and the testing data is mostly attributable to the performance, or lack thereof, of the AIML model.

In our experience, users should avoid using training data with little convergence in the samples generated as this may introduce challenges or concerns with overfitting to individual sample error.

Experimentation with training data volume

We evaluated the relative performance of the model with both 5,000 and 25,000 data points for the training set. Under both iterations, the overall performance over the testing data (measured by R^2 and MAE) were similar. However, as expected, the model exhibited stronger performance under the actuarial tests when trained on the set of 25,000 data points. Below is a summary of the performance of the final

model trained on both sets of training data.

Table 4
PERFORMANCE SUMMARY

Model	No. of Training Scen	Scen RunTime	R ²	MSE	MAE	RMSE
Neural Network - Feat. Set 3	25K	20 hours	99.993%	0	0.00047	0.00064
Neural Network - Feat. Set 3	5K	4 hours	99.967%	0	0.00096	0.00135

The data generation time for 5,000 data points took four hours, while the 25,000 took 20 hours on a single core on our test machine. The runtime costs to generate the data and train the models for both sets is negligible, especially compared to the other case studies explored in this paper.

However, we did notice a difference in performance in our actuarial testing. We noted that the models trained on the set of 5,000 data points lacked precision over the variables that had the most minor impact on the fair value (e.g., interest rates). This was expected as the model will naturally pick up on the most impactful variables and will have to be tuned over a larger dataset before picking up the more nuanced aspect of the Monte Carlo valuation.

It is the responsibility of the modeler to evaluate the circumstance of the application to determine if the additional cost of the data generation is worth the increased performance, especially when considering the ongoing costs if the model will have to be recalibrated periodically.

5.3.3. FEATURE ENGINEERING AND FEATURE SELECTION

Feature engineering can be used to improve the model performance for a given volume of training data. With feature engineering, we can derive additional potential features or input fields for the AIML model that provide greater predictive power than the original data fields.

In our proxy modeling use cases, feature engineering is useful as it is generally used to reduce the data generation requirements and, therefore, decrease the upfront runtime investment needed from actuaries. While we could have skipped this step and achieved similar results through additional data generation, this would come at greater upfront runtime costs and, therefore, reduce the net value gained from the proxy model.

Various additional features were derived from the Monte Carlo inputs provided in the data generation step above. We have primarily used actuarial judgment in designing various potential features derived from the Monte Carlo valuation inputs.

The table below summarizes the full set of potential features, which combines the inputs and additional features engineered.

Table 5
SUMMARY OF POTENTIAL FEATURES

Variable	Source and Description	Candidate?	Selected?
Initial index level	Monte Carlo input Index level at issue or last reset of the index crediting strategy. <i>This variable is fixed at 1,000 and index variables below are normalized based on an initial index level of 1,000.</i>	No <i>Variable is fixed</i>	No
Current index level (ratio)	Monte Carlo input Index level as of the date of the Monte Carlo valuation. <i>Expressed as the ratio of the initial index level.</i>	Yes	Yes
Maximum index level (ratio)	Monte Carlo input Maximum index level reached at month-end since the last reset of the index crediting strategy. <i>Expressed as the ratio of the initial index level.</i>	Yes	Yes
Time to maturity	Monte Carlo input Remaining time (in years) until the index crediting strategy matures and the resulting interest is credited to the account.	Yes	Yes
Cap rate	Monte Carlo input Maximum interest that can be credited to the account (see subsections 5.1 and 5.2 for further details on the cap rate)	Yes	Yes
Risk-free rate	Monte Carlo input Current risk-free interest rate used in Monte Carlo valuation	Yes	Yes
Volatility	Monte Carlo input Volatility used in Monte Carlo valuation. The volatility is fixed at 12% given the underlying is a control volatility index that manages volatility to 12%.	No <i>Variable is fixed</i>	No

Variable	Source and Description	Candidate?	Selected?
Maximum index level to current (ratio)	Engineered feature Maximum index level reached at month-end since the last reset of the index crediting strategy over the current index level.	Yes	Yes
Black-Scholes valuation (call spread)	Engineered feature Rough approximation of the Monte Carlo valuation with a Black-Scholes call spread (buy call at initial index and short a call at initial index * 1 + cap rate). We used the following information for each of the inputs to the Black-Scholes function: <ol style="list-style-type: none"> 1. Stock Price: Maximum index to date 2. Strike Price: <ol style="list-style-type: none"> a. Initial Index for long call b. Initial Index * (1+ Cap rate) for short call 3. Tenor: Remaining Time to Maturity (in years) 4. Interest Rate: Current Interest Rate associated with policy 5. Volatility: Fixed Volatility of 12% 6. Cap rate: Same cap rate as the lookback cap strategy 	Yes	Yes
Black-Scholes valuation (at the money call)	Engineered feature Rough approximation of the Monte Carlo valuation with Black-Scholes. We used the following information for each of the inputs to the Black-Scholes function: <ol style="list-style-type: none"> 1. Stock Price: Maximum index to date 2. Strike Price: Initial Index 3. Tenor: Remaining Time to Maturity (in years) 4. Interest Rate: Current Interest Rate associated with policy 5. Volatility: Fixed Volatility of 12% 	Yes	Yes
Reset indicator	Engineered feature Binary Variable (0 or 1) to indicate whether policy is at reset date	Yes	Yes
Discount Factor	Engineered feature Calculated as: $(1 + \text{Interest Rate})^{-\text{Time to Maturity}/12}$	Yes	Yes

We have not included inputs or features that do not vary as those would not provide any predictive power to the AIML model. This is the case for volatility. The volatility input was held constant as the underlying index is assumed to be managed at a fixed volatility.

As mentioned previously in the report, feature engineering is a subjective exercise that requires professional judgment and intuition. The objective is to select a combination of features that help AIML models make accurate prediction of the outputs. This includes features with high predictive power for the target output. The features selected should also have limited correlation.

In order to support our decision-making process, we trained an XGboost (Yuan, 2023). The feature at “Reset” is not included because it has a feature importance of Zero in the XGboost algorithm.

Figure 12
F SCORE CHART FOR FEATURES

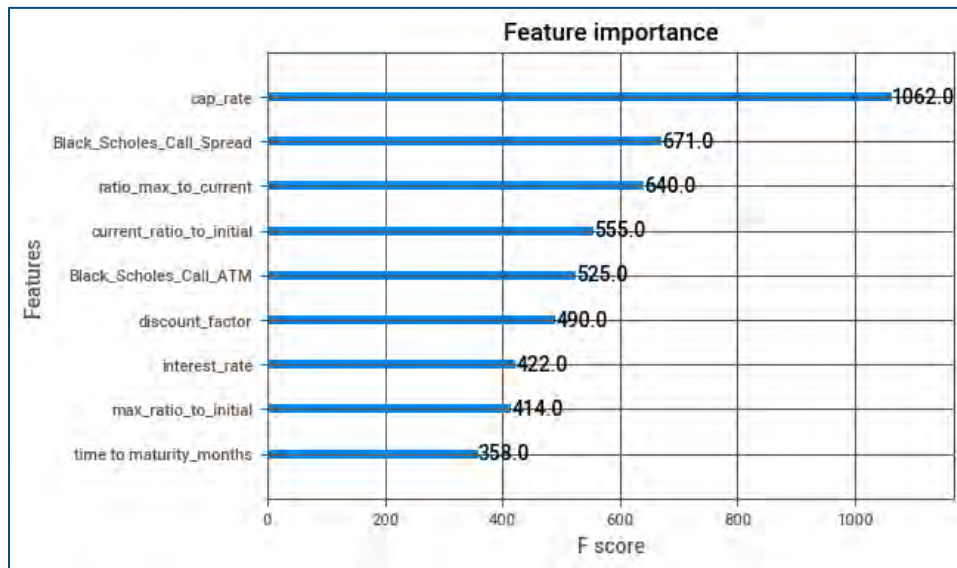


Table 6
SUMMARY OF OPTIONS TESTED FOR FEATURE SETS

Variable	Feature Set 1	Feature Set 2	Feature Set 3
Initial index level	No	No	No
Current index level (ratio)	Yes	Yes	Yes
Maximum index level (ratio)	No	Yes	Yes
Time to maturity	Yes	Yes	Yes
Cap rate	Yes	Yes	Yes
Risk-free rate	Yes	Yes	Yes
Volatility	No	No	No
Maximum index level to current (ratio)	Yes	No	Yes
Black-Scholes valuation	Yes	No	Yes
Black-Scholes valuation (at the money call)	No	No	Yes
Reset indicator	No	No	Yes
Discount Factor	Yes	No	Yes

Output definition

The output of the AIML model will be the Monte Carlo valuation of the exotic index strategy based on \$1 of account value. We did not test alternative or transformed outputs for this case study.

5.3.4. MODEL TESTING AND SELECTION

With the features identified, we have calibrated three commonly used models:

1. Multivariate regression
2. XGBoost
3. Neural Network

See section [1. Predictive analytics, artificial intelligence, and machine learning overview](#) for further information on the background of these models. All models detailed below leveraged these features.

We analyze the performance of these models below. Further actuarial testing is provided in [5.3.5. Actuarial evaluation](#).

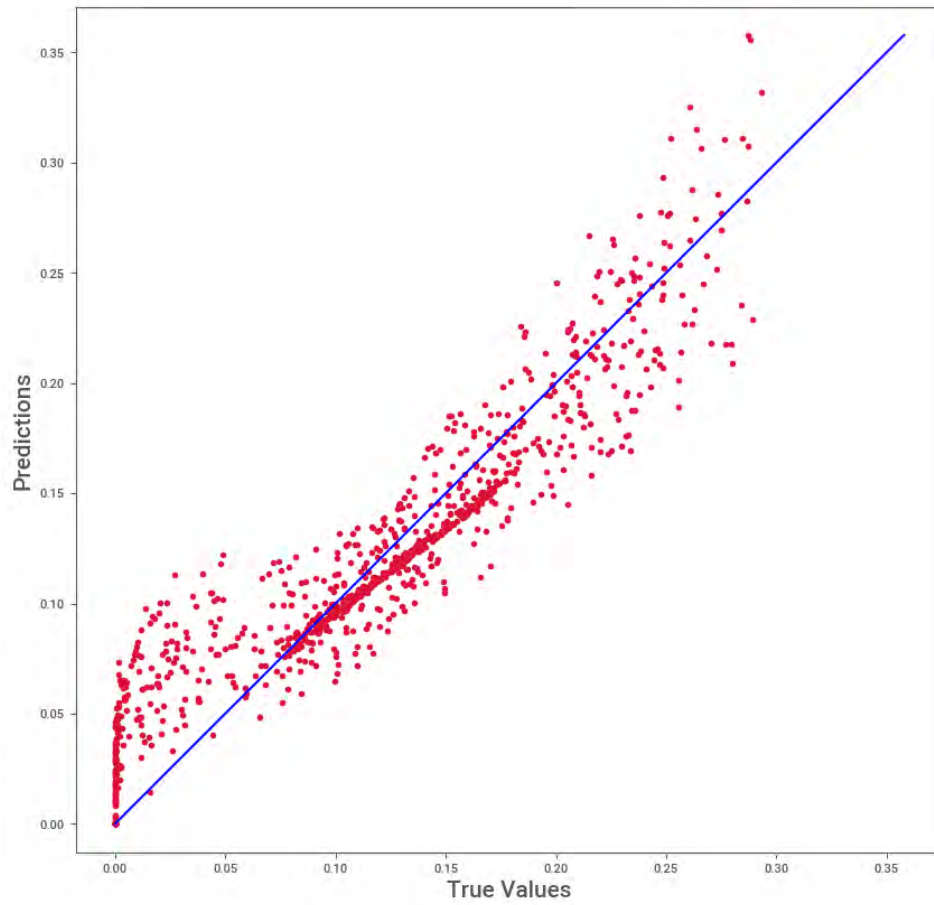
Multivariate regression

We first tested a multivariate regression model using feature set 1 from table 6 above. Our rationale for using this model was the simpler nature of this predictive model relative to other common models such as XGBoost and Neural Networks. Generally, the simplest model that can achieve the desired results is preferable, especially for models that are easier to explain.

This model achieved an R^2 of 85.80% and a mean absolute error (MAE) of .01964.

The graph below illustrates the actual against predicted across the test cases.

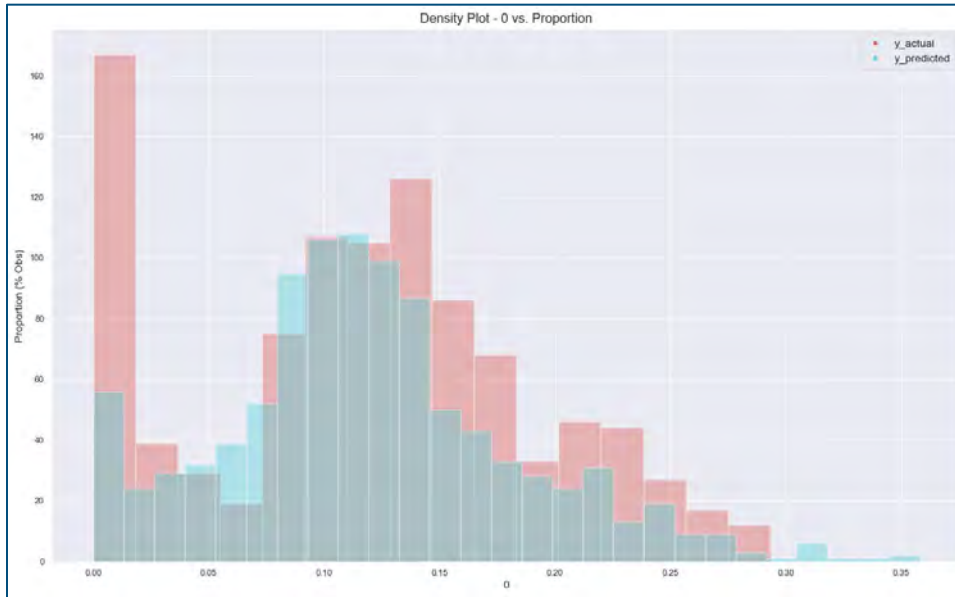
Figure 13
ILLUSTRATION OF ACTUAL AGAINST PREDICTED VALUES



The following graph illustrates the distribution of the predictions made against the actual prediction of the output.

Figure 14

DISTRIBUTION OF THE PREDICTIONS AGAINST ACTUAL PREDICTION OF THE OUTPUT



Unfortunately, the multivariate regression model did not meet our proxy modeling needs.

XGBoost

We then tested the performance of XGBoost, with feature set 1 from table 6 above. This model is often a prime candidate for proxy actuarial models. Again, we used the feature list established in the previous step and the same training and test data.

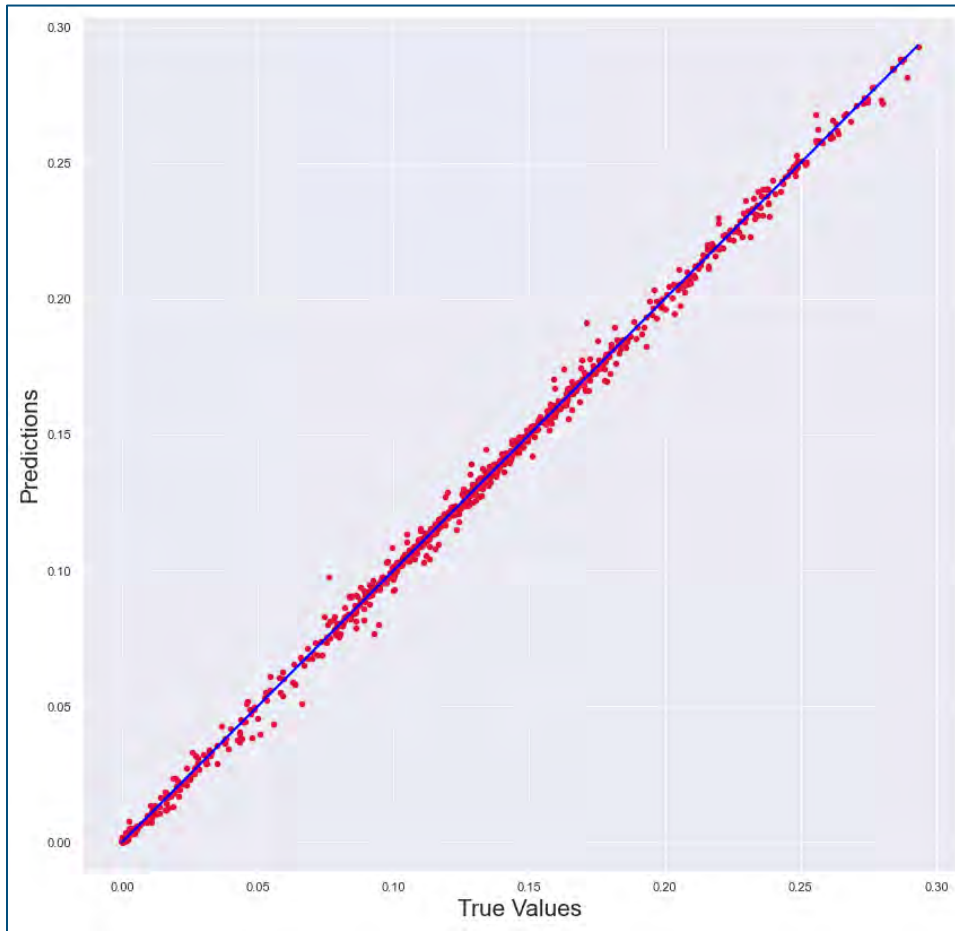
XGBoost produced a much better proxy model than the multivariate regression.

This model achieved an R^2 of 99.84% and a mean absolute error of 0.0018, offering a much more performant proxy model relative to the multivariate linear regression.

The graph below illustrates the actual against predicted across the test cases.

Figure 15

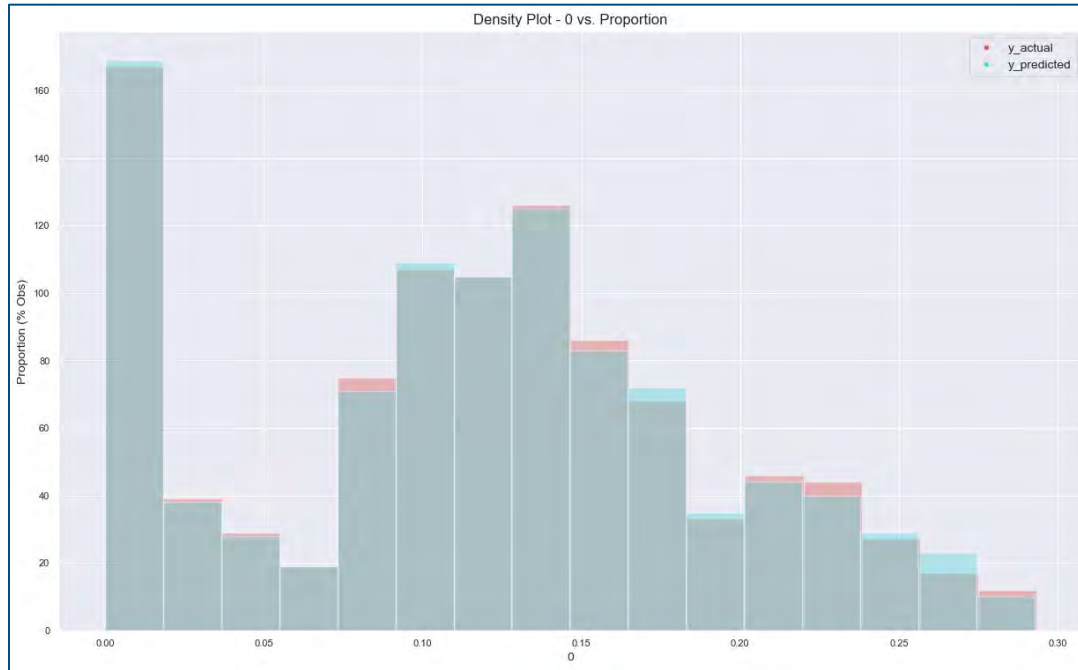
ACTUAL TO PREDICTED GRAPH



The following graph illustrates the distribution of the predictions made against the actual prediction of the output.

Figure 16

DISTRIBUTION OF THE PREDICTIONS AGAINST THE ACTUAL PREDICTION



The XGBoost model appears to be providing adequate performance as a proxy for the index crediting application. However, we decided to continue testing additional models to see if we could further refine the performance of the model.

Neural network

The final model we tested was a neural network (NN). The model chosen was a three-layer Dense NN. The first layer is an input layer with 50 nodes, with one hidden layer of 100 nodes and an output layer of one node. The NN was optimized by calculating weights that minimize the mean squared error between the observed and estimated lookback price. For training, the data was split into 100 batches and each batch was passed through the nodes 100 times.

We tested the performance with all three feature sets from table 6 above as described in [5.3.4 Model Testing and Selection](#). Below is a summary of results of the neural network with each feature set and the other two models tested.

Table 7
RESULTS OF THE NEURAL NETWORK

Model	Feature Set	R ²	MSE	MAE	RMSE
Neural Network	Feature Set 1	99.992%	0	0.00047	0.00065
Neural Network	Feature Set 2	99.988%	0	0.00062	0.00082
Neural Network	Feature Set 3	99.993%	0	0.00047	0.00064
Multiple Least-Squares Regression (MLS)	Feature Set 1	85.793%	0.0007	0.01964	0.02633
XG-Boost	Feature Set 1	99.841%	0.0000	0.00182	0.00297

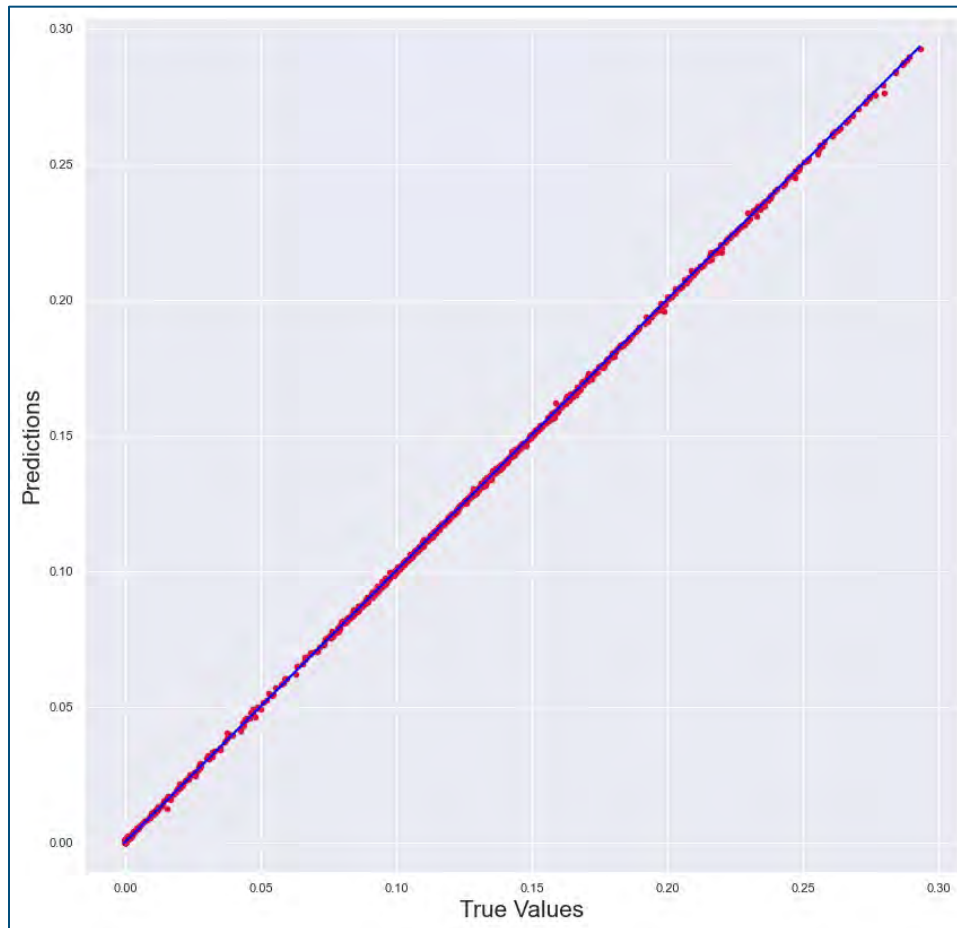
The results of the NN trained on each feature set were similar. The feature set that yields optimized error metrics depends on the kernel and number seed. Feature set 3 was selected for the final NN because it made the most sense from an actuarial perspective. It includes features, such as the Black-Scholes Valuations, which was relevant to the lookback option we were pricing.

The NN was the best performing model of the three tested. The R² was close between the NN and XG-boost but the NN had fewer errors, which was reflected by the MAE.

The improved predictive power is most noticeable on this actual to predicted graph. The predicted and actual values have the tightest fit around the line $y=x$.

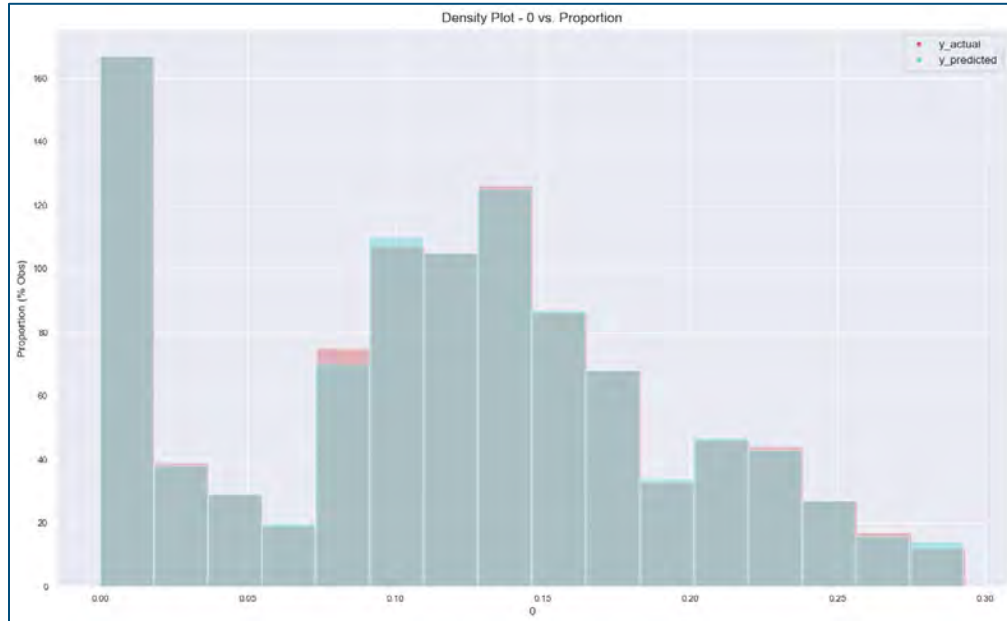
Figure 17

ACTUAL TO PREDICTED GRAPH



The following graph illustrates the distribution of the predicted lookback and the actual lookback option value. The distribution of predicted and actual values has the closest alignment for the NN. This is shown in the graph below.

Figure 18
DISTRIBUTION OF THE PREDICTED AND THE ACTUAL



The following was used to train this neural network:

```

1 early_stop = EarlyStopping(monitor = 'val_mae', patience=20,verbose = 1)
2
3 def baseline_model():
4     # create model
5     model = Sequential()
6     model.add(Dense(50, input_dim=9, kernel_initializer='normal', activation='relu'))
7     model.add(Dense(100, kernel_initializer='normal', activation='relu'))
8     model.add(Dense(1, kernel_initializer='normal'))
9
10    # Compile model
11    model.compile(loss='mean_squared_error', optimizer='adam')
12    return model
13
14    # evaluate model
15    dens_network_50 = KerasRegressor(model=baseline_model, epochs=N_EPOCHS, 16
batch_size=N_BATCH_SIZE, verbose=1)

```

The lines of code above perform the following:

- **Line 1:** Defines the early stopping parameters. Early stopping monitors the progression of the optimization metric and stops the model training once the metric has plateaued.
- **Lines 3 to 8:** Defines the structure of the model. In this case, we defined a Dense feedforward neural network with three layers.
- **Lines 11 to 12:** Compiles the model and specifies to use the mean squared error as the loss and uses the adam optimizer.
- **Line 15:** Trains the model using the function KerasRegressor from the package Keras (Keras, n.d.). It specifies the model structure to train, how many epochs to train on and the batch size.

5.3.5. ACTUARIAL EVALUATION

With the neural network selected as our model of choice based on our review of the model performance in [5.3.4 Model Testing and Selection](#), we continue to test the model performance but now from an actuarial perspective.

Our goal with the tests performed in this section was to better understand the performance of the model when running what-ifs analysis or when exploring relationships.

We compared the performance of the model by varying individual input elements of the Monte Carlo valuation. This allowed us to test whether the AIML model captured individual relationships between the input and the output of the calculation.

We analyzed whether the relationships between the market value and the following were maintained with the AIML model relative to the first principles calculations:

- Current index level
- Maximum index level
- Time to maturity
- Cap rate
- Risk-free rate

Each analysis below is based on the following baseline input values:

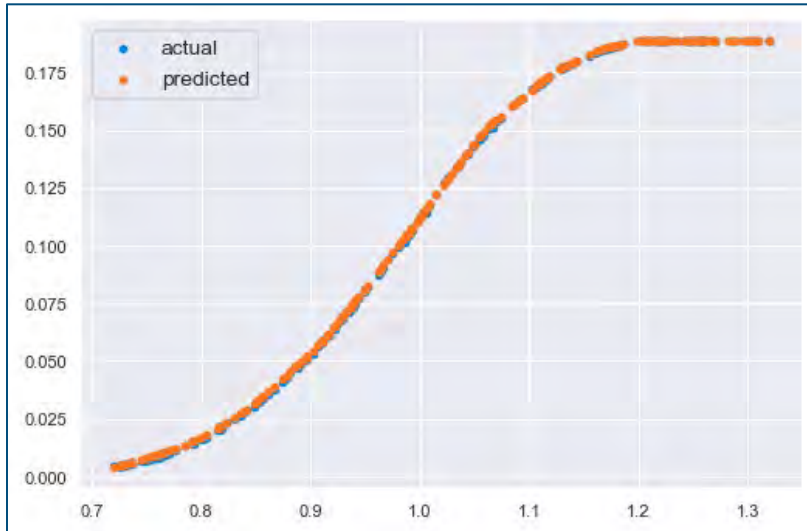
Input variable	Baseline value
Current index level	1100
Maximum index level	1.0 multiplied by the current index level
Time to maturity	24 months
Cap rate	.2
Risk-free rate	.03

Current index level

For this test, we ranged the values of the current index from \$720 to \$1,320. The Neural Network performed very strongly under this test.

Figure 19

ACTUAL TO PREDICTED GRAPH – CURRENT INDEX LEVEL

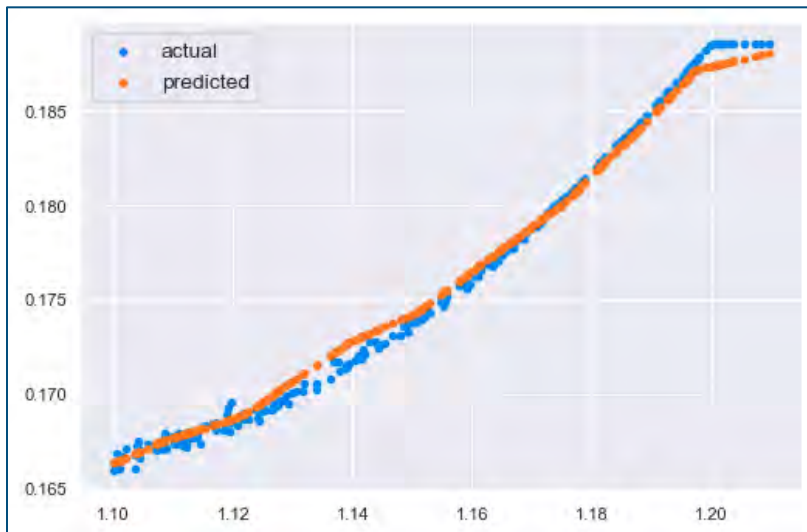


Maximum index level

For this test, we varied the maximum index level from 1 to 1.1 times the current index level of \$1,100 to see how the NN would perform under a variety of past realized high watermarks. The predictive model did a good job capturing the overall shape of the distribution.

Figure 20

ACTUAL TO PREDICTED GRAPH – MAXIMUM INDEX LEVEL



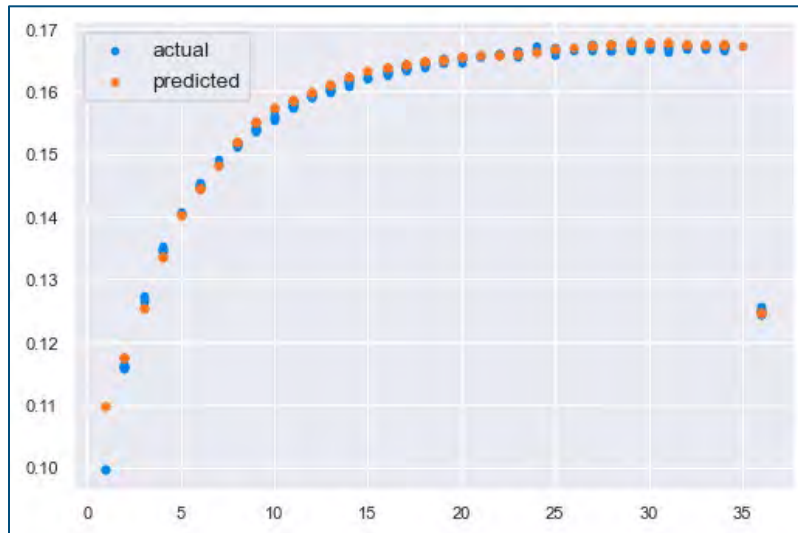
The model had a fair performance under this test, with the predicted values a little higher than the actual values.

Time to maturity

Lookback options with remaining times to maturity were simulated from a range of one month to three years. The NN performed well overall but seemed to have higher inaccuracies for very short tenors.

Figure 21

ACTUAL TO PREDICTED GRAPH – TIME TO MATURITY

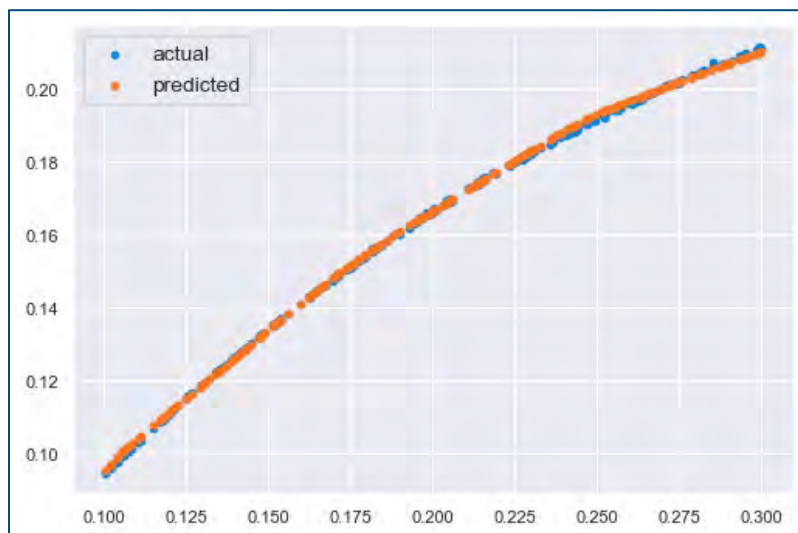


Cap rate

For this test we simulated lookback options with cap rates ranging from 10% to 30%. The NN had a very strong performance here.

Figure 22

ACTUAL TO PREDICTED GRAPH – CAP RATE



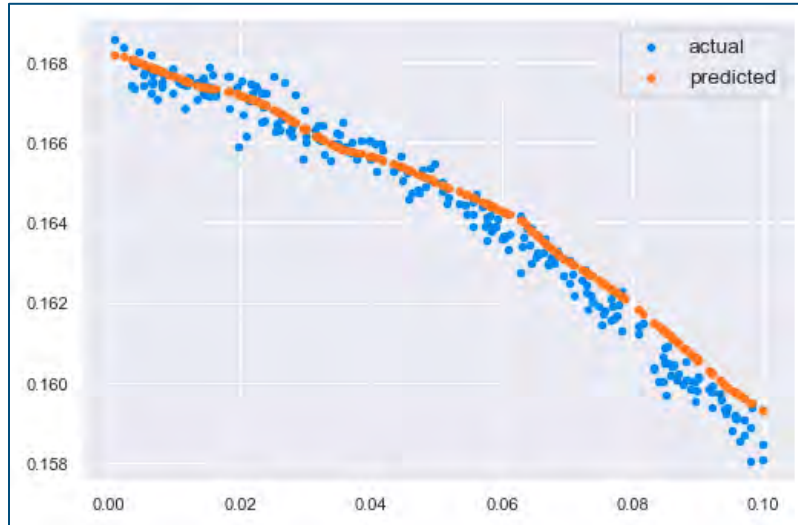
Interest rate

For this test, we ranged the values of the interest rate from .01% to 10%. The model seemed to overestimate the value of the option in a higher interest rate environment. The model generally followed

the pattern of the actual option values. This was a situation where the decision to include additional training scenarios was impactful. When we trained on 5K scenarios, the NN did not capture the pattern as well.

Figure 23

ACTUAL TO PREDICTED GRAPH – INTEREST RATE



5.4. BUSINESS CASE

This case study established that AIML can provide a highly reliable proxy for the Monte Carlo valuation of exotic index crediting strategies. However, it is also important to validate that there is a strong business case for developing such a model.

Below, we provide a few realistic situations where the AIML model could be used in a real-life setting. We use those examples to illustrate the potential return in runtime and cloud cost savings actuaries could gain from using the AIML model developed in this case study.

We note that each use case situation is unique and our goal with the discussion in this section is to provide actuaries not only a sense of the potential savings for different circumstances, but also examples so actuaries can apply this methodology to evaluate if the upfront data generation cost is worth the downstream runtime savings from the AIML model.

Further, we do not attempt to quantify the value of the time investment of the actuary and data scientists. We also stopped short of calculating technology costs such as the runtime costs and any cloud setup fees. However, all those should be factored into the costs associated in developing such a model, and the relative costs of implementing production environments and performing any additional quality controls.

Lastly, actuaries should also consider the additional value gained from being able to run more analysis as a result of the implementation of the AIML model. With the increased actuarial model throughput, it is natural for actuaries and senior management to increase the amount and complexity of the analysis performed. Actuaries can also consider additional intangible value to be able to run quick analysis for management, especially in times of crisis where companies may need to quickly run various stress test analysis.

Key assumptions

The quantification in the following pages assumes the following:

- Running a Monte Carlo simulation of 1,000 scenarios takes 0.1 core-hours on average. This corresponds to the number of scenarios we assume actuaries are running in their actual applications. This assumes that actuaries are accepting intrinsic error in the Monte Carlo valuation due to lack of convergence.
- Running a Monte Carlo simulation of 2,500 scenarios takes 0.25 core-hours on average. This corresponds to the number of scenarios used to generate the training data.
- Running a Monte Carlo simulation of 10,000 scenarios takes 1 core-hours on average. This corresponds to the number of scenarios used to generate the testing data.

We have summarized those general settings in the following tables.

Cost and Runtime Input	Value	Description
Core-hour per nested stochastic (1k scenarios)	0.1	Number of core hours required to complete the nested stochastic (actual application)
Core-hour per nested stochastic (2.5k scenarios)	0.25	Number of core hours required to complete the nested stochastic (training)
Core-hour per nested stochastic (10k scenarios)	1	Number of core hours required to complete the nested stochastic (testing)

Additionally, we summarized the training and testing data generation needs (same as what was used in subsection [5.3. AIML Model Development](#)).

Example 1: Pricing application

In this first example, we explored a case where a company uses the AIML model as part of an actuarial pricing model. The pricing model runs 100 cells over 250 pricing scenarios and a 30-year projection horizon. Further, the pricing model is run once a month.

General Inputs	Value	Description
Model points	1	We assume that all policies share the same characteristics for the Monte Carlo valuation and, therefore, a single calculation takes place for all model points
Runs per year	24	Number of times the pricing process is run in a year (twice a month)
Outer-loop scenarios	250	Number of outer-loop scenarios to be run
Inner-loop scenarios	0	Not applicable
Inner per outer	0	Not applicable
Nested stochastic per outer	360	Assumes monthly for 30 years
Nested stochastic per inner	0	Not applicable

Calculate Outer Nested Stochastic	Value	Description
Outer nested stochastic	2,160,000	Total number of nested stochastic calculations for this example

Cost Breakdown	Value	Description
Data generation – Core hour	6,250	Core hours to generate the training data
Data generation – Core hour	5,000	Core hours to generate the testing data
Annual Non-AIML Core hours	216,000	Annual core hours to run the application with Monte Carlo valuation methodology
Return on investment	1820%	Ratio of the costs over a given year

Again, this analysis is limited to the runtime improvement.

Example 2: Reserve forecasting run

In this second example, we explored a case where actuaries needed to perform quarterly balance sheet and income statement forecasts for an in-force block of FIA with the exotic index crediting strategy as part of the company's financial planning and analysis (FP&A) and risk management applications.

With this application, five pre-determined deterministic scenarios are produced every quarter. The entire in-force population is run, but the Monte Carlo valuation is shared between cohorts (36 cohorts given the three-year lookback feature).

General Inputs	Value	Description
Model points	36	We assume that all policies share the same characteristics for the Monte Carlo valuation and, therefore, a single calculation takes place for all model points
Runs per year	4	Quarterly run
Outer-loop scenarios	5	Number of outer-loop scenarios to be run
Inner-loop scenarios	1000	Number of scenarios ran to calculate VM-21
Inner per outer	10	Reserve is calculated 10 times along the outer-loop
Nested stochastic per outer	360	Assumes monthly for 30 years
Nested stochastic per inner	360	Assumes monthly for 30 years

Calculate Outer Nested Stochastic	Value	Description
Outer nested stochastic	259,200	Total number of outer nested stochastic calculations for this example

Calculate Inner Nested Stochastic	Value	Description
Inner nested stochastic	648,000,000	Total number of inner nested stochastic calculations for this example

Cost Breakdown	Value	Description
Data generation - Core hour	6,250	Core hours to generate the training data
Data generation - Core hour	5,000	Core hours to generate the testing data
Annual Non-AIML Runtime Cost	64,825,920	Annual core hours to run the application with Monte Carlo valuation methodology
Return on investment	576130%	Ratio of the costs over a given year

5.5. CONCLUSIONS

This case study demonstrated how the model development methodology detailed in this paper can be applied to develop high-fidelity AIML models to proxy the Monte Carlo valuation complex exotic index crediting strategies (to calculate the market value, option budget or Greeks).

These models can be useful to actuaries when no appropriate closed-form formulas or approximations are available.

This specific application is generally less complex given the limited set of inputs involved in this calculation, making it easier for actuaries to develop such AIML models. This application generally requires less data and does not require as extensive feature engineering or hyperparameter tuning to achieve the desired results.

This is why we selected this use case to warm up readers before tackling the next case study.

Section 6: Case Study 2 - Variable Annuity Fair Value

6.1. INTRODUCTION

In this section, we explore the use of AIML to produce proxy models for estimating the fair value of variable annuity guarantees, also commonly known as variable annuity riders, or Guaranteed Minimum Benefits (GMxBs).

Variable annuities are perhaps one of the most common, if not the most common, product to run into nested stochastic runtime challenges and is, therefore, natural for us to explore nested stochastic runtime issues with this product in both this case study and the next.

GMxB riders have allowed insurers to differentiate variable annuity products from other investment or retirement vehicles offered by banks and fund managers. Sales of variable annuities have been steadily decreasing over the last two decades as a result of the market environment, lower risk appetite and rise of alternative annuity products, such as FIA and Registered Indexed Linked Annuities (RILA), where both offer market participation to the policyholder with downside protection but generate less market exposure for the insurer. This is due to the market participation being provided through index crediting mechanisms as detailed in the first case study. These index crediting mechanisms can generally be completely offset by the insurer when purchasing replicating derivatives.

While the sales volume of variable annuities has decreased over the past few years, this product still remains a prominent offering within the roster of products offered by insurance companies. Further, insurers still need to manage the sizable blocks of variable annuities they have accumulated over the last few decades. This requires proper actuarial valuation and projection capabilities to assess and manage the risk of the block.

Note that the techniques and concepts covered in this case study are not limited to variable annuities. These concepts can be easily adapted or extended to other products with similar nested stochastic challenges, such as Market Risk Benefits (MRB) for other products, and such as FIAs or other reserve calculations that may be onerous.

6.1.1 BACKGROUND AND PRODUCT FEATURE OVERVIEW

GMxB background

Variable annuities are separate account products where policyholders can invest in a range of funds that combine exposure to various asset classes such as equities, bonds, mortgages, real estate, cash and other cash equivalents.

GMxBs are additional riders that can be attached to variable annuities. These riders provide additional valuable guarantees to policyholders to help protect against market downturns. These guarantees can take various forms. Generally, the benefits provided by these GMxBs are based on a guaranteed amount or benefit base. We will generally use the term benefit base throughout the rest of this paper.

Below is a very high-level overview of these guarantees:

1. **Guaranteed Minimum Death Benefit (GMDB):** This type of rider provides a death benefit guarantee. The guarantee provides for the greater of the account value and the benefit base upon passing of the policyholder.

2. **Guaranteed Minimum Accumulation Benefit (GMAB):** This type of rider provides account value downside projections through a guaranteed accumulation benefit. This benefit will adjust the account value at the end of the guaranteed term to the benefit base if the account value is below the benefit base at that time. In many product designs, the guaranteed accumulation benefit will expire after the guaranteed term.
3. **Guaranteed Minimum Income Benefit (GMIB):** This type of rider provides income protection through a guaranteed minimum annuitization option, based on guaranteed annuitization rates that can be applied to the guaranteed amount. If and when the policyholder decides to annuitize, he will get the greater annuity payout between applying current annuitization factors to the account balance, or the guaranteed annuitization rates applied to the benefit base. Generally, GMIB products will couple with other living benefits, such as withdrawal benefit features, until annuitization.
4. **Guaranteed Minimum Withdrawal Benefit (GMWB) or Guaranteed Lifetime Withdrawal Benefit (GLWB):** This type of rider also provides for income projection but through guaranteed withdrawals rather than a guaranteed annuitization option. The rider allows policyholders to take scheduled withdrawals at a contractually set rate based on the benefit base and age of the start of the withdrawals. The withdrawals can generally continue as long as the benefit base is active and even if the account value is depleted. As such, the goal of this rider is to provide protection against running out of funds from the account value once the policyholder has started withdrawals.

The benefit base may be the original premium, or a balance accruing at a given rate, ratcheting up periodically at the current account value, or subject to other designs.

The product concept of variable annuity allows insurers to vary product designs significantly; therefore, actuaries will see many unique product designs in the market.

Fair value of variable annuity GMxBs

The core principle of fair valuation is to leverage observable prices from the market to calibrate a model and then apply the calibrated model to derive the price of similar market instruments that do not have an observable price.

Given fair value is well documented in the literature, including in SOA papers, we will refrain from providing a detailed walkthrough of this concept and refer readers that may be new to fair value to other papers such as “How Fair Value Measurement Changes Risk Management Behavior in the Insurance Industry” from the SOA (Rosner, 2013)¹.

In this section, our objective is to develop an AIML model that can proxy the fair value of two GMxB products to a high degree of fidelity using a similar methodology to what we covered in the first case study.

Runtime challenge for variable annuity fair value

The calculation of fair value is cumbersome to actuaries as it typically requires running a large number of stochastic scenarios to reach appropriate convergence or precision in the fair value result.

¹ <https://www.soa.org/resources/research-reports/2013/research-how-fair-value/>

Actuaries typically rely on scenario reduction techniques, such as scenario selection and unique scenarios per policy, to overcome this challenge.

However, these techniques alone are generally not sufficient to circumvent the runtime challenges when actuaries need to calculate fair value in a nested stochastic or projection context. This is due to the compounding amount of calculations that need to be executed to perform such projections.

To provide an example, let's consider a case where an actuary needs to perform projections of the VA book of business of its company, which includes projection of the hedge income under a delta hedge projection.

Let's assume the following parameters:

- **Policies:** 500,000
- **Projection scenarios:** 10
- **Projection length:** 30 years
- **Hedge methodology:** Delta (shock up and down)
- **Hedge frequency:** Monthly
- **Risk-neutral scenarios:** 500
- **Risk-neutral projection length:** 30 years with monthly timesteps

This would mean that the actuary would have to perform 3.6 billion ($500,000 \times 10 \times 30 \times 2 \times 12$) fair value calculations with each calculation requiring running 500 scenarios over 30 years or $1.8E^{12}$ individual 30-year projections. We will get further into actual runtime and costs below, but we can see how the projection of variable annuity fair value can quickly get out of hand.

6.2 PRODUCT AND ACTUARIAL METHODOLOGY SPECIFICATIONS

Product features

We will focus our analysis in this section on two VA products. [Appendix A](#) provides a summary of the product specifications.

Both products share the same base product features such as the separate account funds available for investments, fees and surrender charge schedules:

- The underlying funds include a fixed allocation to equity (60% to the S&P 500) and bonds (40% to a US Long-Term Corporate Bond index).
- The fees include an annual M&E charge of 1.3% of the account balance (charged daily), an annual per policy charge of \$100 and a fund management charge of 100bps.
- The surrender charge schedule is based on a period of seven years and grades from 7% to 1% during that period.

The two products only differ in the GMxBs offered with the product:

- The first product is offered with an enhanced GMDB.
- The second product is offered with a GLWB with a return of premium (ROP) death benefit.

The product with enhanced GMDB comes with a rider fee that varies by issue age bracket and gender, ranging from 0.62% to 2.10% annually. The product with GLWB comes with a constant rider fee of 1.25%

annually. The rider fees are applied to the benefit base.

The guaranteed withdrawal rates for the GLWB rider vary between 3.30% and 5.60% annually, increasing with the age at the time of first withdrawal.

More information on the two GMxB products is provided in Appendices [A2](#) and [A3](#).

For simplicity, these products are assumed only on one life (no joint lives).

Valuation methodology

The fair value in this section will be derived according to standard actuarial practices, with the key components being the following:

- **Product features** will be reflected as-is, as in most standard actuarial projections. Product features were already detailed above and further detail can be found in [Appendix A: Variable annuity product specifications](#).
- **Risk-neutral stochastic scenarios** will be used to develop interest rates and equity paths. The risk-neutral economic scenarios will be calibrated to the current market (current yield curve and volatility). The generator behind these scenarios is the key element that drives the market consistency of the fair valuation from its calibration. The risk-neutral scenarios are detailed further down in this section.
- **Best-estimate actuarial assumptions** will be used to project the cash flows of the variable annuity product, including the claims and fees arising from the GMxB guarantee. The actuarial assumptions are detailed further down in this section.

Risk-neutral stochastic scenarios

The risk-neutral scenario generator is based on the extended Cox Ingersoll Ross model (referred to as CIR++ model) for the short rate and Heston for equities (the Heston model is equivalent to the log-normal model with stochastic volatility that follows a mean-reverting process similar to CIR).

The risk-neutral generator was designed to project the following elements at monthly timesteps:

- Short rate
- Yield curve (derived from the short rate)
- Single equity returns
- Bond fund
- Money market

While various other risk-neutral generators could have been used for this case study, and there is a wide range of industry practices, we felt that the proposed risk-neutral generator was representative of the key characteristics actuaries typically look for in a risk-neutral generator for the valuation of the fair value for variable annuities.

Short rate and yield curve

The short rate was based on the CIR++ model calibrated to the yield curve at the time of valuation (or revaluation at the future pivots). The model followed this form:

The functional form of the CIR++ model is as follows:

$$r_t = x_t + \varphi_t$$

Where r_t is the instantaneous short rate at time t , x_t is the original short rate produced from the CIR process and φ_t is the shift factor that calibrates the CIR++ model to market observable prices.

The stochastic process for $x(t)$ follows the standard CIR process:

$$dx_t = a \cdot (b - x_t) dt + \sigma \cdot \sqrt{x_t} dW_t^x$$

The calibration of the shift factors φ_t were performed against the US treasury curve. For simplicity, no other market instruments were used for the calibration process.

The CIR parameters used are as follows:

Parameter	Value
Mean-reversion speed (a)	15%
Mean-reversion level (b)	3%
Volatility (σ)	12.5%

The implied yield curve is also produced from the CIR++ model.

Equity

The Heston model was used to model the equity process. The Heston model is defined by the following stochastic process:

$$dS_t = r_t \cdot S_t dt + \sqrt{v_t} \cdot S_t dW_t^S$$

Where S_t is the index level as of time t and v_t is the instantaneous variance. Further, the instantaneous variance itself follows a stochastic process defined as follows:

$$dv_t = \kappa \cdot (\theta^2 - v_t) dt + \varepsilon \cdot \sqrt{v_t} dW_t^v$$

Where κ is the mean-reversion speed, θ is the mean-reversion level and ε is the volatility of the stochastic volatility (v_t).

Parameter	US Equity Market
Mean-reversion speed (κ)	190%
Mean-reversion level (θ)	20%%
Volatility of volatility (ε)	35%

Bond fund

Bond fund returns are proxied assuming constant rebalancing between the following bond maturities:

Term (years)	Allocation
1	1%
2	3%
3	5%
4	10%
5	20%
6	20%
7	16%
8	10%
9	10%
10	5%

Money market

The money market is assumed to earn the risk-free rate.

Correlations

The dependency between the underlying random variables is captured through the following correlation matrix:

	r_t	S	v
r_t (short rate)	100%	10%	0%
S	10%	100%	0%
v	0%	0%	100%

Best-estimate actuarial assumptions

The 2015 VBT mortality table was used for the mortality assumption, a simple dynamic function was used for lapse, and stochastic withdrawal paths were used as actuarial assumptions.

Readers can refer to [Appendix B: Variable Annuity Actuarial Assumptions](#) for detailed information on the best-estimate assumptions.

6.3 AIML MODEL DEVELOPMENT

Overview and Problem Statement

This section details the methodology used to calibrate and test AIML models to proxy the fair value for the two VA products (the VA with GMDB and the VA with GMWB) as described in subsection [6.2 Product and Actuarial Methodology Specifications](#).

We will use methodology consistent with the principles laid out in section [3. Practical Application of AIML](#)

for [Actuarial Modeling](#). Further, the rest of this section is structured similarly to how we structured the first case study from section [5 Case study 1 – Index crediting](#) as we used consistent methodology throughout this paper to develop the case studies.

The following table summarizes the steps used for this case study and associated references from the sub-sections below.

Table 8
SUMMARY OF STEPS AND REFERENCES FOR THE CASE STUDY

#	Step	Paper Section	Description
1	Problem statement	This section	Calculate the fair value for the two VA products
2	Preparation	6.3.1. Preparation	Prepare the environment, load external packages and define the various functions that will support the steps below.
3	Data generation	6.3.2. Data generation	Define methodology to generate sample results to train and test the AIML models for the VA FV case study. <i>Note that, as opposed to the first case study on index crediting, the VA FV case study and the VA capital case studies rely on an external actuarial software to generate the training and testing data and, therefore, the code or functionality to perform the data generation will not be made available to users of this report for these two case studies.</i>
4	Feature engineering and selection	6.3.3. Feature engineering, feature selection and output definition	Develop additional inputs for the AIML model derived from the original VA FV valuation inputs. Perform data exploration and feature importance analysis based on data generation with added feature engineering to identify which features should be used as inputs to the AIML model.
5	Model testing and selection	6.3.4. Model testing and selection	Test performance of various AIML models. Test and control for overfitting. Select the top model based on performance and judgment.
6	Hyperparameter tuning	6.3.5 Hyperparameter tuning	Test alternative hyperparameters for the selected model to refine performance. When needed, hyperparameter tuning can improve the performance of a given model.
7	Actuarial evaluation	6.3.6. Actuarial evaluation	Evaluate the performance of the model from an actuarial perspective. This includes performing actuarial forecasts and comparing the projected fair value using the actuarial model and using the AIML model. This analysis can help us understand the performance from an actuarial point of view.

The methodology to develop such AIML models generally requires iterating between the steps outlined above. In this section, we focus on the final methodology and analysis that was developed by the

researchers.

Only one Python notebook was developed for this case study (as opposed to two notebooks for the first case study on index crediting, again due to only providing the code for the machine learning development in this case study).

The notebook, along with the data provided in this paper, includes all the code needed to replicate the results provided in this section. The notebook contains the typical code structure to train and test AIML proxy actuarial models. This notebook starts by importing the data that we generated from our actuarial model and then provides the code to execute the steps outlined in the table above.

While we do not provide the code to produce the data generation since we used a vendor software, readers can find the full product specifications in [Appendix A: Variable annuity product specifications](#) and [Appendix B: Variable annuity assumptions](#).

The remainder of 6.3 walks the reader through each step highlighted above.

6.3.1. PREPARATION

The first step consists of preparing the environment by loading any external packages, and defining the user functions and user inputs. Preparation for the notebook for the second case study is similar to the preparation for the AIML model development notebook from the first case study (second notebook).

Instructions are provided in Appendix A (see [A1: Installing Python](#)) to setup an environment to run and use Python to replicate the results from this case study.

The environment will require the use of Python packages. A full list of packages used in this report with references is also provided in Appendix A (see [A2: Python Packages](#)). The most notable packages for this case study include:

- **NumPy:** Adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- **Pandas:** Used for data transformation and analysis.
- **Matplotlib, Sweetviz and seaborn:** Used to produce various visualizations.
- **Scikit-learn (sklearn) and Keras:** Both of these packages provide a library of machine learning models.

6.3.2. DATA GENERATION

Overview and general methodology

Given the nature of the case study, increased complexity, and limitations of the actuarial model, we decided to diverge from the methodology used for data generation from what was used in the first case study.

For this case study, we instead opted to leverage actuarial projection methodology to support the data generation process. We used economic scenarios to forecast the liabilities over a period of 30 years and then performed fair value calculations along those paths in order to generate the data for this case study.

The steps used in this case study are outlined in the following table:

Table 9
SUMMARY OF STEPS

#	Step	Description
1	Economic scenario generator selection	<p>A real-world economic scenario generator (ESG) was used to generate the economic scenarios. While other methodologies could have been used, an ESG was selected to generate the economic environments for training the AIML model as ESGs are already designed to generate a wide range of realistic yield curves and equity paths.</p> <p>The uniform methodology used in the previous case study would have had the challenge of generating realistic yield curves.</p> <p>Using a real-world economic scenario generator is reasonable because, otherwise, actuaries would have to define an entirely new process to sample realistic yield curves and paths for equities and other market components. Given the level of sophistication of today's ESGs in achieving reasonable projections of economic environments, it made sense to use it for our use case.</p> <p>We have decided to use the joint real-world ESG from the American Academy of Actuaries and the Society of Actuaries – also known as the Academy Interest Rate Generator (AIRG) (SOA, 2022). The rationale for using this ESG is its widespread use among actuaries in the United States and avoiding the need to define an additional ESG in this paper, which would add unnecessary additional complexity for the reader.</p>
2	Scenario selection	<p>However, an important characteristic of real-world economic scenario generators is that the distribution of the projected market components is probability-weighted.</p> <p>Given that the objective of our AIML proxy model is to provide a reasonable approximation of the fair value of our GMxBs across the entire distribution for the production use case, we should adapt the output from the AAA generator such that the distribution is more akin to a uniform distribution.</p> <p>Otherwise, we may oversample observations around the apex of the distribution and under sample in the tails, resulting in uneven performance of the model, with notably poor performance around the tails where the data is sparse.</p> <p>To avoid this situation, we opted to develop a scenario selection technique where we selected a set of representative scenarios from a larger set generated from the AAA generator such that we had a more level distribution across interest rates and accumulation factors for combined equity, bond and money market.</p> <p>Additional information for the scenario selection process is provided further down in this section in Scenario selection process.</p>
3	New business cell selection	<p>In order to help manage runtime, only a subset of the new business records was run over each individual scenario selected.</p> <p>The subset used was randomized independently for each scenario.</p>

#	Step	Description
4	Generation	<p>For each scenario and business cell selection combination, perform an actuarial projection of each new business cell.</p> <p>Along that projection, produce reserves for the pivot points identified. For this use case, we used at issue, every year for the first five years and then every five years from there on out until year 30.</p> <p>Similar to the first case study, we used a staggered inner-loop scenario count with 2,500 inner risk-neutral scenarios for training data and 10,000 inner risk-neutral scenarios for testing. Again, this was used to achieve as little error as possible on the testing data while accepting a little bit more error on the training data to achieve sufficient data volume for developing a model.</p> <p>Additional analysis on scenario convergence for both the VA GMDB and VA GMWB is provided further down below in Variance reduction techniques and testing for convergence.</p>

We have consolidated the documentation of the VA FV inputs with the feature engineering. See [6.3.3. Feature engineering, feature selection and output definition](#).

Scenario selection process

Efficient data generation is one of the key aspects to making the cost benefit for training machine learning models viable. The goal is to improve the performance of the AIML model for a given amount of data generation budget and provide data to the model where it can “learn” the most. For this case study, we designed and implemented a scenario selection algorithm to reduce the number of model-point cross scenarios needed for data generation. This was achieved by selecting a uniform coverage of scenarios for each training set of data generated, such that the range of predefined scenarios was covered by the least number of points.

One thousand stochastic AAA scenarios were generated for each of 50 starting yield curves for a total of 50,000 stochastic scenarios. The 50 starting curves are selected from monthly historical curves from year 1953 to 2021 with subject matter expertise to cover a range of level, slope, curvature and characteristics. The scenarios are generated using calibrated parameters from real world ESG published by the American Academy of Actuaries (AAA, n.d.). The starting yields are set to capture historical range of interest rate environments. This generated universe, therefore, casts a coverage of a range of market environments.

We then applied the clustering algorithm to select representative scenarios from the set of 50,000 based on k-nearest neighbor clustering along two dimensions that would have impacts to VA fair value:

- Equity growth rate
- Interest rates: representative points of selected 10-year term rates projected as of year 10

Fifty thousand points were used to form 5,000 clusters using k-nearest neighbor (KNN), which were the number of desired representative scenarios. Scenarios that were closest in Euclidean distance to the centroids of the formed clusters were selected as the subset. The 10-year point was chosen as a representative point for the average duration of the population. The resulting selected scenarios provided a wide coverage of the market environment while reducing the data generation cost budget to not need to generate data on similar environments repeatedly.

Variance reduction techniques and testing for convergence

Similar to the first case study, the antithetic variate variance reduction technique was used to reduce the error in the samples (i.e., fair value for each new business cell) generated to train and validate the AIML model. See [Variance reduction techniques and testing for convergence](#) in subsection [3.2 Data generation](#) for details.

Then, we used bootstrapping (Efron, et al., 2012) to quantify the error for different numbers of inner-loop risk-neutral scenarios (N). This test was done over five sample policies where we generated the fair value (average of the PV of claims minus fees across the risk-neutral scenarios) of the rider across 10,000 scenarios.

Table 10

LIST OF POLICY INFORMATION

Policy #	Rider Type	Issue Age	Withdrawal Start	Gender	Revaluation (Pivot) Year	Moneyiness at Pivot (Benefit Base / AV)
1	Enhanced GMDB	40	N/A	Male	5	109%
2	Enhanced GMDB	55	N/A	Female	10	197%
3	Enhanced GMDB	70	N/A	Male	5	119%
4	GLWB with ROP	40	After pivot	Female	10	115%
5	GLWB with ROP	65	After pivot	Female	0	100%

The variance for each of the five above test cases is illustrated below. The variance is measured through the 5% to 95% confidence intervals for an increasing number of risk-neutral scenarios.

Figure 24
FAIR VALUE VARIANCE



The grey line was calculated using 10,000 scenarios and used to approximate fully converged results. The scenario samples were randomly selected from a set of 10,000 based on bootstrapping technique (subset selection of the 10,000 with replacement).

We opted to use 2,500 scenarios for the training data and 10,000 scenarios for the testing data. This decision was made based on professional intuition. Readers should consider testing various convergence levels to identify an efficient balance between convergence and runtime.

6.3.3 FEATURE ENGINEERING, FEATURE SELECTION AND OUTPUT DEFINITION

Feature engineering and feature selection

As introduced previously in this paper, feature engineering can be used to improve the model performance for a given volume of training data. With feature engineering, additional potential features or input fields for the AIML model that provide greater predictive power than the original data fields can be derived.

This is important for approaching more complex applications, especially as it becomes more difficult or costly to generate data due to runtime limitations. This is the case as we move from the index crediting to the VA FV case study.

In our proxy modeling use cases, feature engineering is useful as it is generally used to reduce the data generation requirements and, therefore, decrease the upfront runtime investment needed from actuaries.

The table below summarizes the full set of potential features, which combines the inputs and additional features engineered. Various additional features were derived from the original inputs to the VA FV calculations and summarized in the table below.

Table 11
SUMMARY OF POTENTIAL FEATURES

Variable	Source and Description	Candidate?
Economic scenario	Economic scenario generator Scenario ID from the subset of the AAA scenarios as selected by the scenario selection process defined above.	No <i>Information only</i>
Issue age	Information Age of the new business record at issue.	Yes
Issue date	Information Date of policy issue (same for all records given this is a new business / pricing application).	No <i>Information only</i>
Account value	Fair value input Total amount accumulated in the account value for this variable annuity policy at the time of the fair value calculation.	Yes
Attained age	Fair value input Attained age of the policyholder at the time of the fair value calculation.	Yes
Life expectancy	Engineered feature Expected remaining life expectancy of the policyholder based on the mortality assumption at the time of the fair value calculation.	Yes
Lives remaining	Information How many partial lives are remaining on the policy based on decrements to date in the projection. All other applicable variables will be adjusted to 1 full life at the valuation point.	No
Projection year (fractional)	Fair value input How many years the fair value valuation point is in the projection at the time of the fair value calculation. While not necessarily a direct input to the fair value calculation, this information may be a key predictor.	Yes
Policy year	Information Current age of the policy in years at the time of the fair value calculation. Same value as the projection year given we are in an actuarial pricing context	No

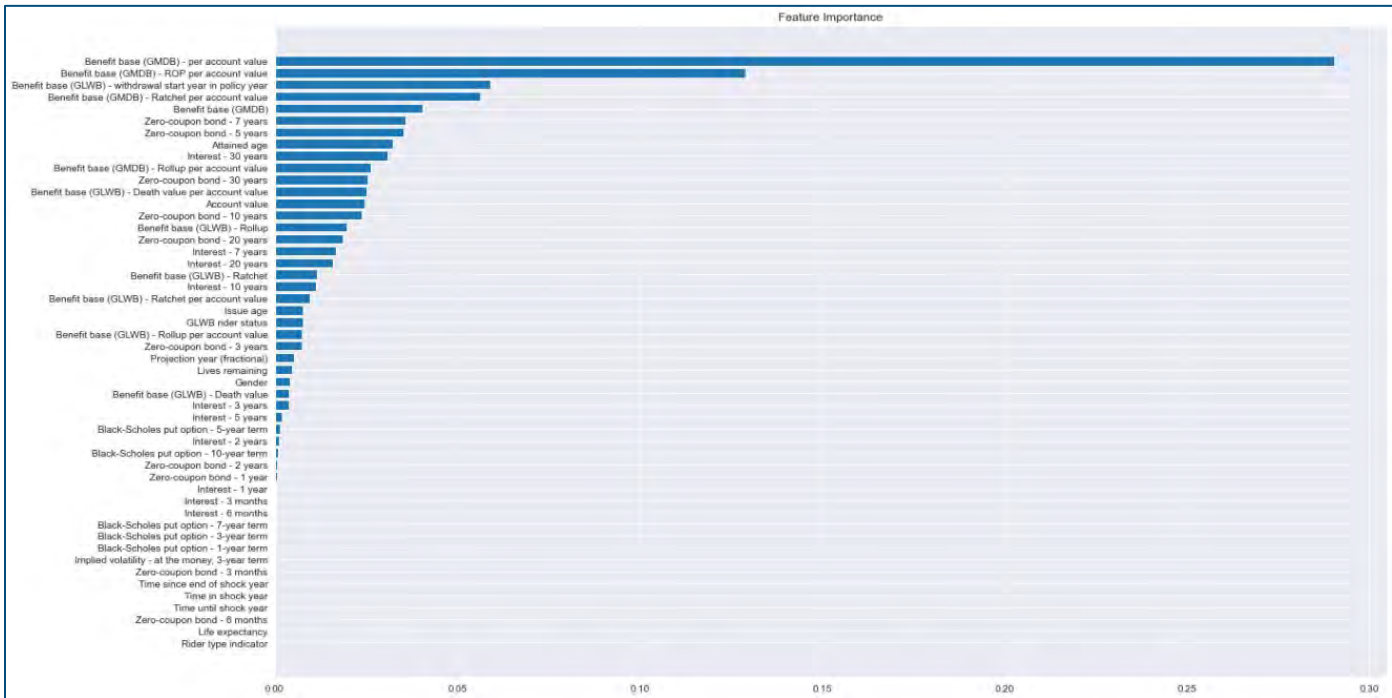
Variable	Source and Description	Candidate?
	so this one would be excluded.	
Time until shock year	Engineered feature Transformation of policy year. Captures the time remaining until the shock lapse year (floored at 0) at the time of the fair value calculation.	Yes
Time in shock year	Engineered feature Transformation of policy year. Captures the fractional year that passed since the start of the shock lapse year at the time of the fair value calculation.	Yes
Time since end of shock year	Engineered feature Transformation of policy year. Captures the fractional years since the end of the shock lapse year at the time of the fair value calculation.	Yes
Attained age	Fair value input Current age of the policyholder at the time of the fair value calculation.	Yes
Benefit base (GMDB)	Fair value input Current benefit base amount for the GMDB guarantee at the time of the fair value calculation (all benefit base features that follow are also at the time of the fair value calculation).	Yes
Benefit base (GMDB) – ROP	Fair value input Current return of premium portion of the benefit base amount for the GMDB guarantee.	Yes
Benefit base (GMDB) – Rollup	Fair value input Current rollup portion of the benefit base amount for the GMDB guarantee.	Yes
Benefit base (GMDB) – Ratchet	Fair value input Current ratchet portion of the benefit base amount for the GMDB guarantee.	Yes
GLWB rider status	Fair value input Current status of the GLWB rider (accumulation or withdrawal).	Yes
Benefit base (GLWB) – Rollup	Fair value input Current rollup portion of the benefit base amount for the GLWB guarantee.	Yes
Benefit base (GLWB) – Ratchet	Fair value input Current ratchet portion of the benefit base amount for the GLWB guarantee.	Yes
Benefit base (GLWB) – Death value	Fair value input Current death benefit under the GLWB guarantee.	Yes
Benefit base (GLWB) – withdrawal start year in policy year	Fair value input Policy year when GMWB withdrawal begins.	Yes
Benefit base (GMDB) – per account value	Feature engineering Ratio of this benefit base component over the account value.	Yes
Benefit base (GMDB) – ROP per account value	Feature engineering Ratio of this benefit base component over the account value.	Yes
Benefit base (GMDB) – Rollup per account value	Feature engineering Ratio of this benefit base component over the account value.	Yes
Benefit base (GMDB) – Ratchet per account value	Feature engineering Ratio of this benefit base component over the account value.	Yes
Benefit base (GLWB) – Rollup per account value	Feature engineering Ratio of this benefit base component over the account value.	Yes
Benefit base (GLWB) – Ratchet per account value	Feature engineering Ratio of this benefit base component over the account value.	Yes

Variable	Source and Description	Candidate?	
Benefit base (GLWB) – Death value per account value	Feature engineering Ratio of this benefit base component over the account value.	Yes	
Interest – 3 months	Fair value inputs Modeled point on the risk-free yield curve.	Yes	
Interest – 6 months	Fair value inputs Modeled point on the risk-free yield curve.	Yes	
Interest – 1 year		Yes	
Interest – 2 years		Yes	
Interest – 3 years		Feature engineering Price of a zero-coupon bond at the specified duration based on the risk-free yield curve.	Yes
Interest – 5 years			Yes
Interest – 7 years			Yes
Interest – 10 years			Yes
Interest – 20 years			Yes
Interest – 30 years			Yes
Zero-coupon bond – 3 months		Yes	
Zero-coupon bond – 6 months		Feature engineering Price of a zero-coupon bond at the specified duration based on the risk-free yield curve. TBD	Yes
Zero-coupon bond – 1 year			Yes
Zero-coupon bond – 2 years	Yes		
Zero-coupon bond – 3 years	Yes		
Zero-coupon bond – 5 years	Yes		
Zero-coupon bond – 7 years	Yes		
Zero-coupon bond – 10 years	Yes		
Zero-coupon bond – 20 years	Yes		
Zero-coupon bond – 30 years	Yes		
Implied volatility – at the money, 3-year term	Yes		
Black-Scholes put option – 1-year term	Feature engineering Price of an at-the-money put option issued as of the time of the fair value calculation and of the specified duration.	Yes	
Black-Scholes put option – 3-year term	Feature engineering Price of an at-the-money put option issued as of the time of the fair value calculation and of the specified duration.	Yes	
Black-Scholes put option – 5-year term		Yes	
Black-Scholes put option – 7-year term		Yes	
Black-Scholes put option – 10-year term		Yes	
Rider type indicator	Input Indicates whether the record is a GMDB or a GLWB. This has been converted to a binary for the AIML model (0 for GMDB and 1 for GLWB)	Yes	
Gender	Input Indicates the gender of the policyholder (0 for Male and 1 for Female)	Yes	

As mentioned previously in the report, feature engineering is a subjective exercise that requires professional judgment and intuition. The objective is to select a combination of features that help AIML models make accurate predictions of the outputs. This includes features with high predictive power for the target output. The features selected should also have limited correlation.

To support our decision-making process, we trained an XGboost. Here is the ranking summary table for all candidates noted above.

Figure 25
RANKING SUMMARY TABLE



As the illustration shows, many of the features showed insignificant predictive power to target and support our professional judgment and intuition. We further performed analysis on three difference feature sets based on:

1. Feature importance score above 25bps.
2. Most common VA actuarial model inputs.
3. Subjective feature set based on our judgment.

Below is the feature set summary table:

Table 12
SUMMARY OF FEATURE SET

Variable	Option 1	Option 2	Option 3
Account value	Yes	Yes	Yes
Attained age	Yes	Yes	Yes
Benefit base (GLWB) - Death value	No	Yes	Yes

Variable	Option 1	Option 2	Option 3
Benefit base (GLWB) - Death value per account value	Yes	Yes	Yes
Benefit base (GLWB) - Ratchet	Yes	Yes	Yes
Benefit base (GLWB) - Ratchet per account value	Yes	Yes	Yes
Benefit base (GLWB) - Rollup	Yes	Yes	Yes
Benefit base (GLWB) - Rollup per account value	Yes	Yes	Yes
Benefit base (GLWB) - withdrawal start year in policy year	Yes	Yes	Yes
Benefit base (GMDB)	Yes	Yes	Yes
Benefit base (GMDB) - per account value	Yes	Yes	Yes
Benefit base (GMDB) - Ratchet per account value	Yes	Yes	Yes
Benefit base (GMDB) - Rollup per account value	Yes	Yes	Yes
Benefit base (GMDB) - ROP per account value	Yes	Yes	Yes
Black-Scholes put option - 10-year term	No	No	Yes
Gender	No	Yes	Yes
GLWB rider status	Yes	Yes	Yes
Implied volatility - at the money 3-year term	No	Yes	Yes
Interest - 1 year	No	Yes	No
Interest - 10 years	Yes	Yes	No

Table 13

SUMMARY OF INTEREST RATE AND BOND RETURN

Variable	Option 1	Option 2	Option 3
Interest - 2 years	Yes	Yes	No
Interest - 20 years	Yes	Yes	No
Interest - 3 months	No	Yes	No
Interest - 3 years	Yes	Yes	No
Interest - 30 years	No	Yes	No
Interest - 5 years	Yes	Yes	No
Interest - 6 months	No	Yes	No
Interest - 7 years	Yes	Yes	No
Issue age	Yes	No	Yes

Variable	Option 1	Option 2	Option 3
Lives remaining	Yes	No	Yes
Projection year (fractional)	Yes	Yes	Yes
Rider type indicator	No	Yes	No
Zero-coupon bond - 10 years	Yes	No	Yes
Zero-coupon bond - 20 years	Yes	No	Yes
Zero-coupon bond - 3 months	No	No	Yes
Zero-coupon bond - 3 years	Yes	No	Yes
Zero-coupon bond - 30 years	Yes	No	Yes
Zero-coupon bond - 5 years	Yes	No	Yes
Zero-coupon bond - 7 years	Yes	No	Yes

Output definition

The table below summarizes the various outputs that were considered for this use case.

Table 14

SUMMARY OF OUTPUTS

Variable	Source and Description	Candidate?	Selected?
Present value of claims	Fair value output Average of the PV of claims across the risk-neutral scenarios for a given valuation.	Yes	No
Present value of fees	Fair value output Average of the PV of fees across the risk-neutral scenarios for a given valuation.	Yes	No
Net present value	Fair value output Average of the PV of claims minus fees across the risk-neutral scenarios for a given valuation.	Yes	No
Present value of claims – scaled	Fair value output Present value of claims divided by the maximum of the account balance, GMDB and GMWB benefit base, as applicable.	Yes	No
Present value of fees – scaled	Engineered output Present value of fees divided by the maximum of the account balance, GMDB and GMWB benefit base, as applicable.	Yes	No
Net present value – scaled	Engineered output Net present value divided by the maximum of the account balance, GMDB and GMWB benefit base, as applicable.	Yes	Yes

The net present value – scaled as selected as the model output for the following reasons:

1. Training models separately for the claims and fees portions of fair value would require two models. Further, each model would contribute error to the net calculation.
2. Training the model for the net present value after scaling for the account value and benefit base

transforms the output to remove the direct impact of the size of the policy on the net present value.

Therefore, the engineered output “net present value – scaled” was selected for this case study.

6.3.4 MODEL TESTING AND SELECTION

With the features identified, we calibrated three commonly used models:

1. Multivariate regression
2. XGBoost
3. Neural Network

See section [1. Predictive analytics, artificial intelligence, and machine learning overview](#) for further information or background on these models. All models detailed below leveraged the features. We analyze the performance of these models below. Further actuarial testing is provided in [6.3.6. Actuarial evaluation](#).

All illustrations provided below can be produced with the Python code provided in Appendix A.

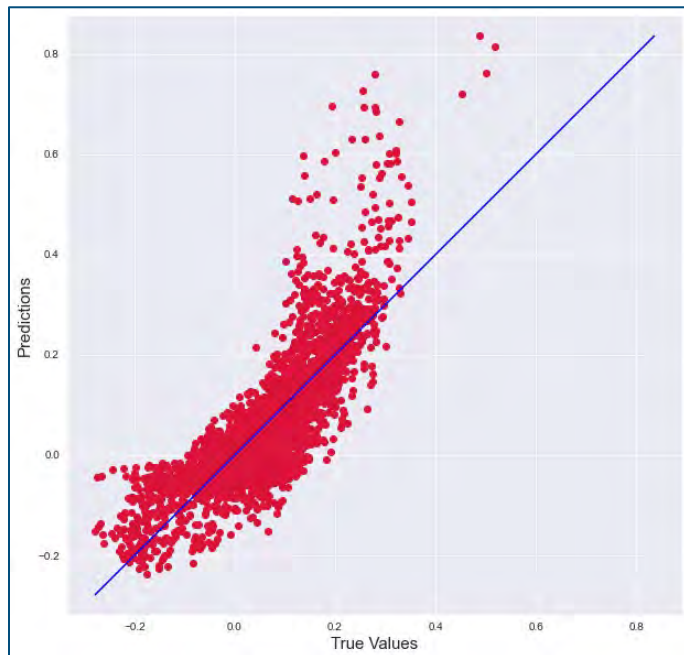
Multivariate regression

Similar to case study 1, we began by testing a multivariate regression model using feature set 1. This model achieved a R^2 of 66.45% and a mean absolute error (MAE) of .04380.

The graph below illustrates the actual against predicted across the test cases:

Figure 26

ACTUAL TO PREDICTED GRAPH FOR MULTIVARIATE REGRESSION



XGBoost

We then tested the performance of XGBoost with feature set 1. This model is often a prime candidate for proxy actuarial models. Again, we used the feature list established in the previous step and the same training and test data.

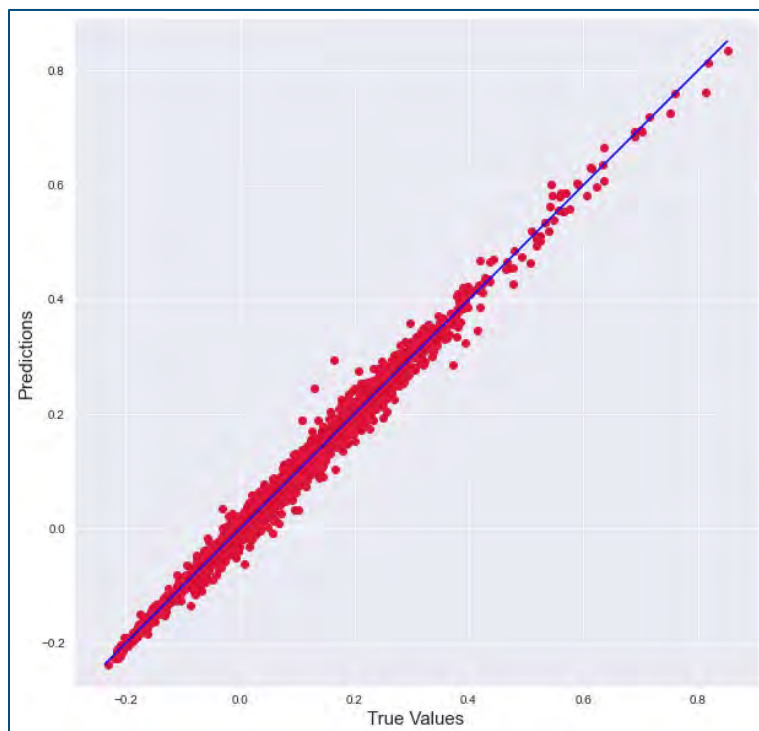
XGBoost produced a much better proxy model than the multivariate regression.

This model achieved a R^2 of 98.84% and a mean absolute error of 0.00014, offering a much more performant proxy model relative to the multivariate linear regression.

The graph below illustrates the actual against predicted across the test cases:

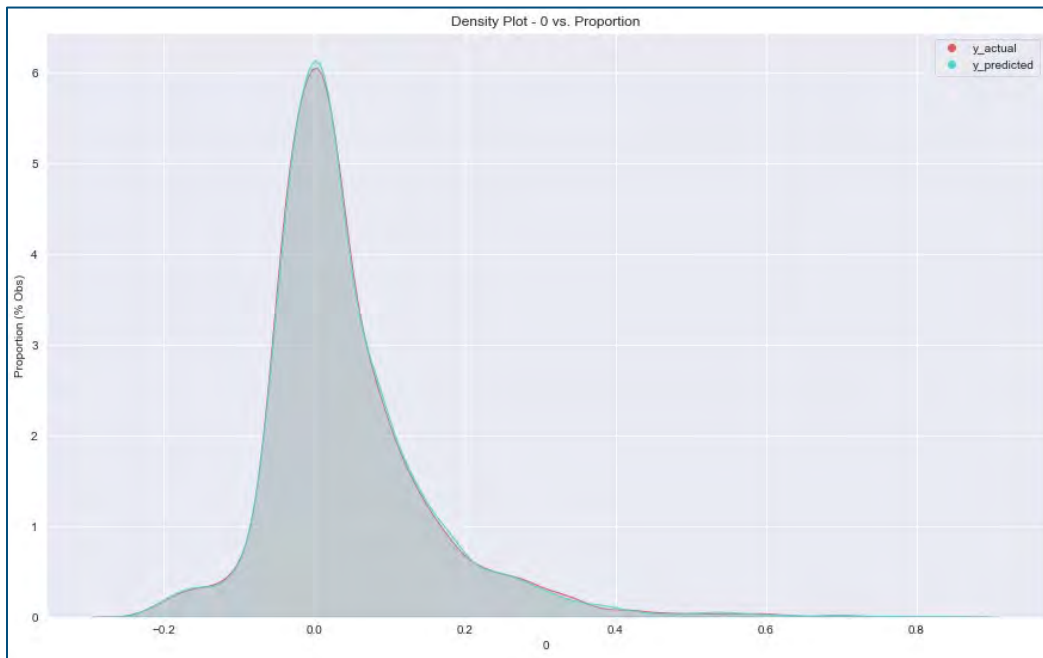
Figure 27

ACTUAL TO PREDICTED GRAPH FOR XGBOOST



The following graph illustrates the distribution of the predictions made against the actual prediction of the output:

Figure 28
THE DISTRIBUTION OF THE PREDICTIONS OUTPUT VS ACTUAL OUTPUT



The XGBoost model appears to be providing adequate performance as a proxy for the index crediting application. However, we decided to continue testing additional models to see if we could further refine the performance of the model.

Neural network

The final model we tested was a neural network (NN). The model chosen was a three-layer Dense NN. The first layer is an input layer with 100 nodes, with one hidden layer of 200 nodes and an output layer of one node. The NN was optimized by calculating weights that minimize the mean squared error between the observed and estimated lookback price. For training, the data was split into 100 batches and each batch was passed through the nodes 100 times.

We tested the performance with all three feature sets as described in [6.3.4 Model Testing and Selection](#). Below is a summary of the results of the neural network with each feature set and the other two models tested.

Table 15
SUMMARY OF THE RESULTS

Model	Feature Set	R ²	MSE	MAE	RMSE
Neural Network	Feature Set 1	99.116%	0.00011	0.00714	0.01033
Neural Network	Feature Set 2	99.010%	0.00012	0.00762	0.01093
Neural Network	Feature Set 3	99.357%	0.00008	0.00592	0.00881
Multiple Least-Squares Regression (MLS)	Feature Set 1	66.454%	N/A	0.04380	N/A
XG-Boost	Feature Set 1	98.842%	0.00014	0.00795	0.01183

The results of the NN trained on each feature set are similar. The feature set that yields optimized error metrics depends on the kernel and number seed. Feature set 3 was selected for the final NN because it made the most sense from an actuarial perspective. It includes features, such as the Black-Scholes Valuations, relevant to the lookback option we were pricing.

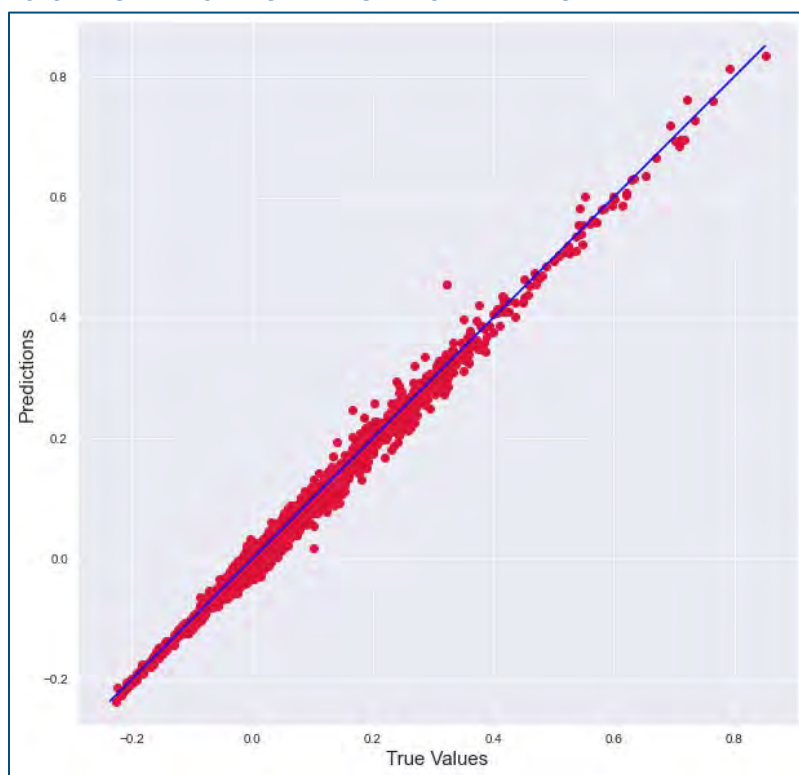
The NN was the best performing model of the three tested. The R^2 were close between the NN and XG-boost, but the NN had fewer errors, which is reflected by the MAE.

The improved predictive power is most noticeable on this actual to predicted graph. The predicted and actual values have the tightest fit around the line $y=x$.

The graph below illustrates the actual against predicted across the test cases:

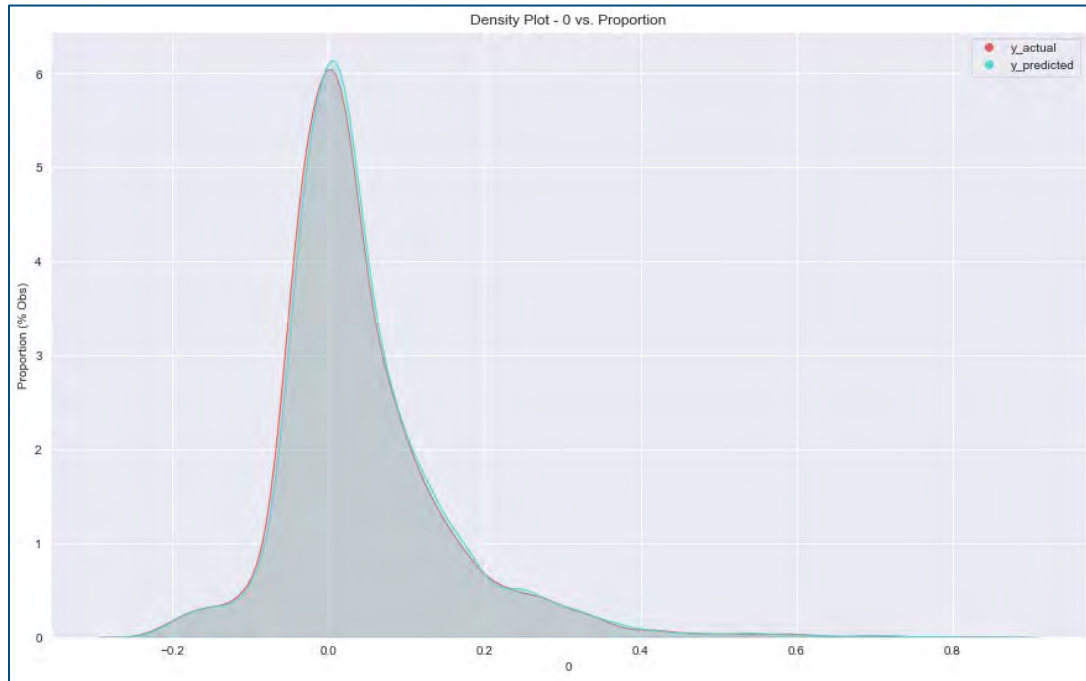
Figure 29

ACTUAL TO PREDICTED GRAPH FOR NEURAL NETWORK



The following graph illustrates the distribution of the predictions made against the actual prediction of the output:

Figure 30
DISTRIBUTION OF THE PREDICTIONS AGAINST THE ACTUAL



6.3.5 HYPERPARAMETER TUNING

The model selection above clearly shows the advantage of using the neural network model for VA fair value problems. As mentioned in [subsection 1.3](#), there are a wide range of fine adjustments that will impact the performance of the model and predictions. Therefore, the Variable Annuities Fair Value problem outlined above resulted in three hidden layers of ResNet model with 256 nodes on each layer. The resulting model structure and parameters are in the summary table below.

Table 16
PARAMETER SUMMARY

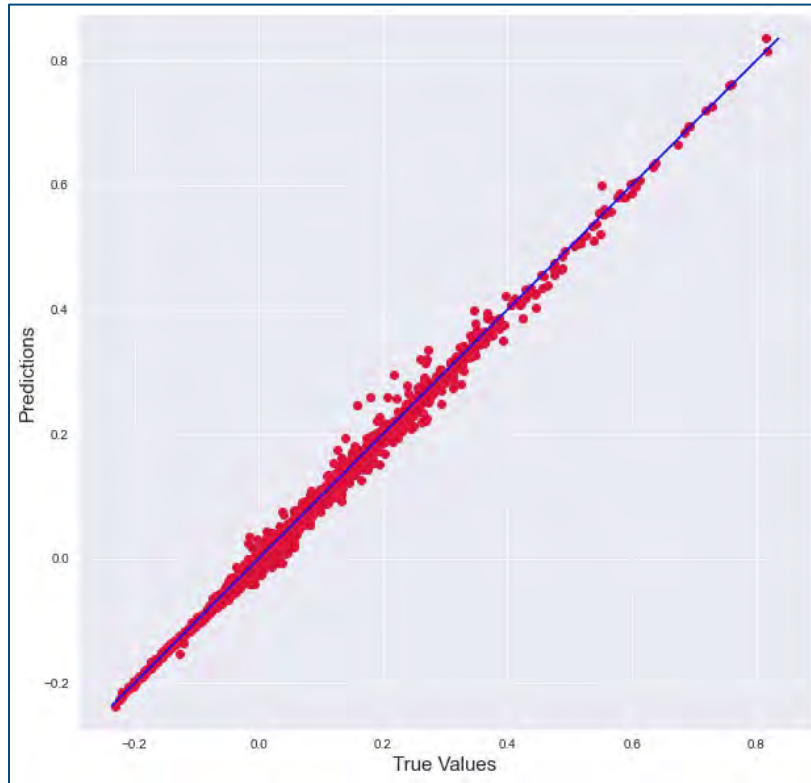
Parameter Tunings	Parameters
Model type	ResNet
Number of hidden layers	3
Residuals layer	Gaussian Noise
Activation	Relu
Loss function	MSE
Learning rate	0.0003
Batch size	1000
Epochs	2000
Neurons	256
Validation split	0.3
Callback - early stop monitor	RMSE
Callback - early stop min delta	0.001
Callback - early stop patience	25

Using feature set 3 and the tuned model parameters above, we compared the performance with a different combination of the model types, and it resulted in a model with better performance or fit against the test data. This model achieved a R^2 of 99.49%, MSE of 0.00006, MAE of 0.0511, and RMSE of 0.00784.

The graph below illustrates the actual against predicted across the test cases:

Figure 31

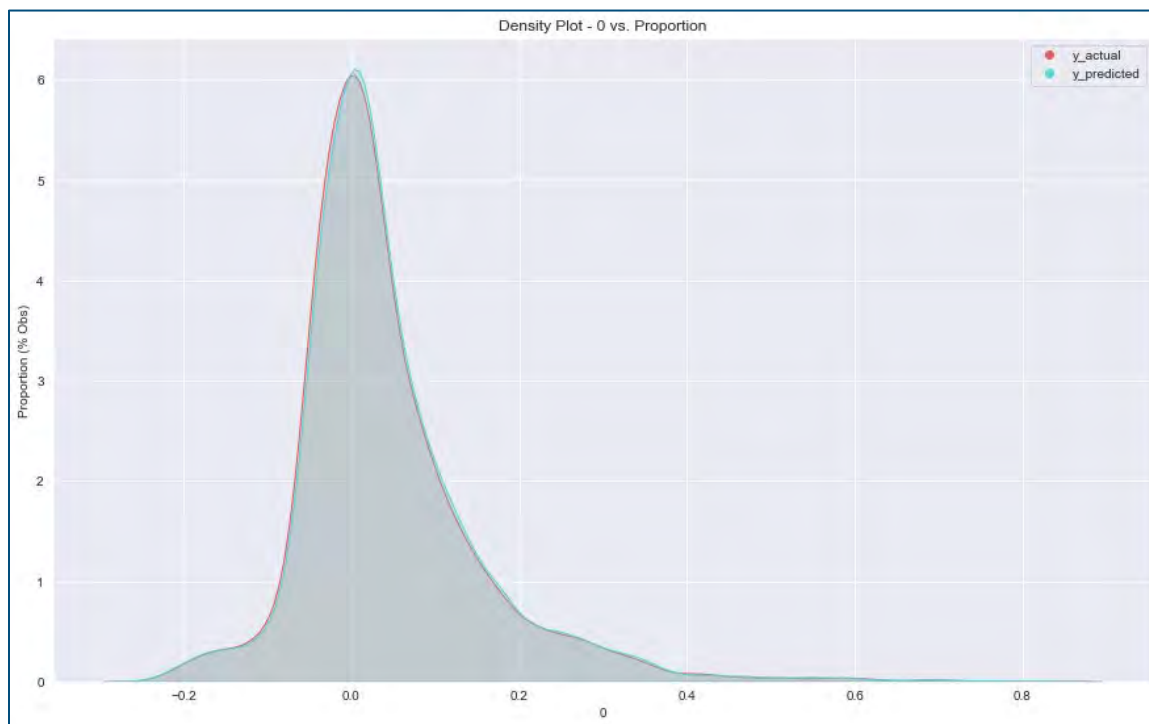
ACTUAL TO PREDICTED GRAPH



The following graph illustrates the distribution of the predictions made against the actual prediction of the output:

Figure 32

DISTRIBUTION OF THE PREDICTION AGAINST THE ACTUAL



6.3.6 ACTUARIAL EVALUATION

To evaluate the performance of the model, we generated five independent stochastic scenarios using the AAA generator. We projected pricing cells over each scenario using 1000 unique risk-neutral scenarios for each pricing cell.

Key findings

Across a variety of variable annuity fair value time series progression patterns, we observed that the predicted fair values tracked the actual results well. Please note that the fair value shown here is the difference between claims and fees, the net impact.

The accuracy of the results was compared across 30 years of fair value projections. Each time point was predicted using the trained Neural Network against the actual results produced by the actuarial model.

Figure 33
SCENARIO 1

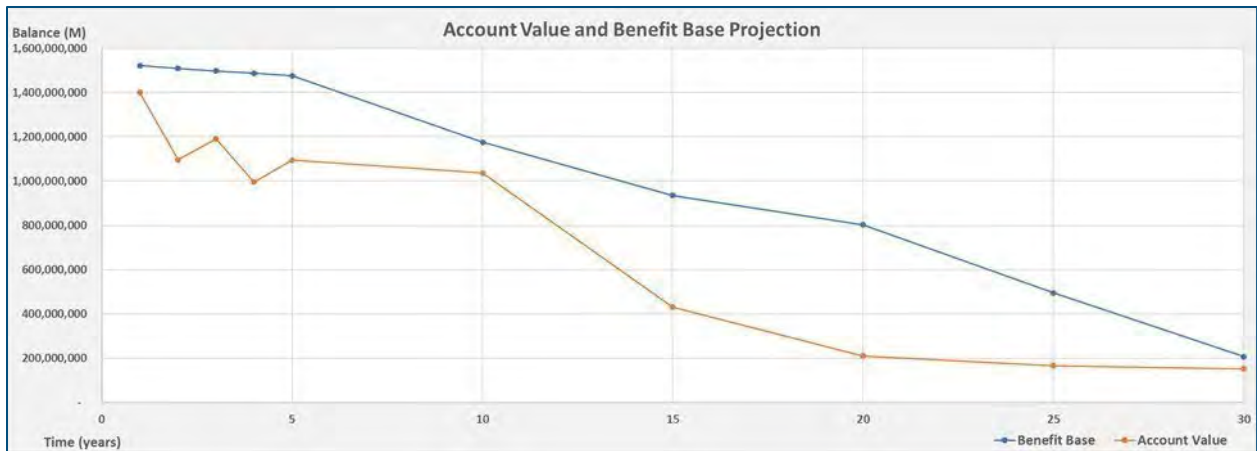
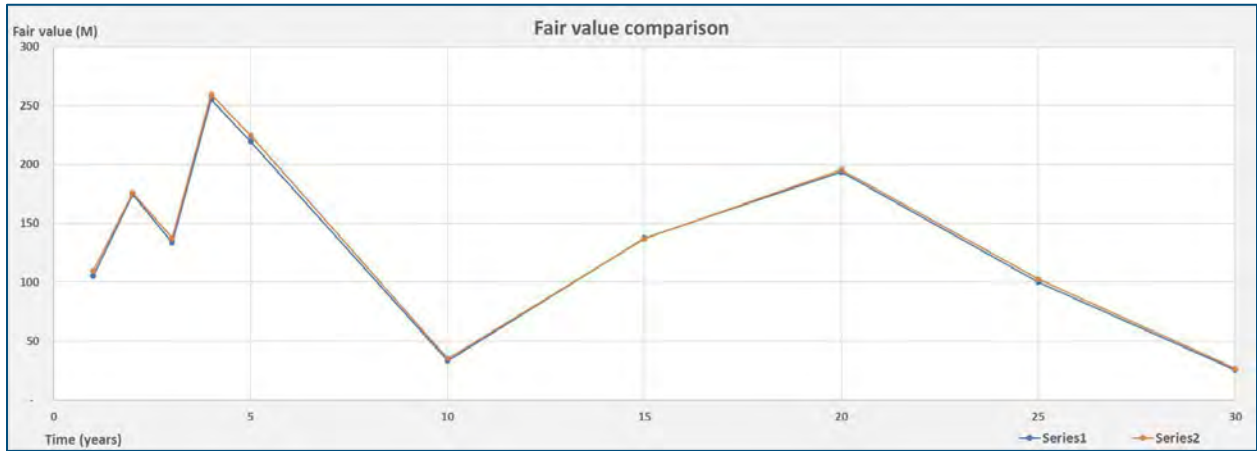


Figure 34
SCENARIO 2

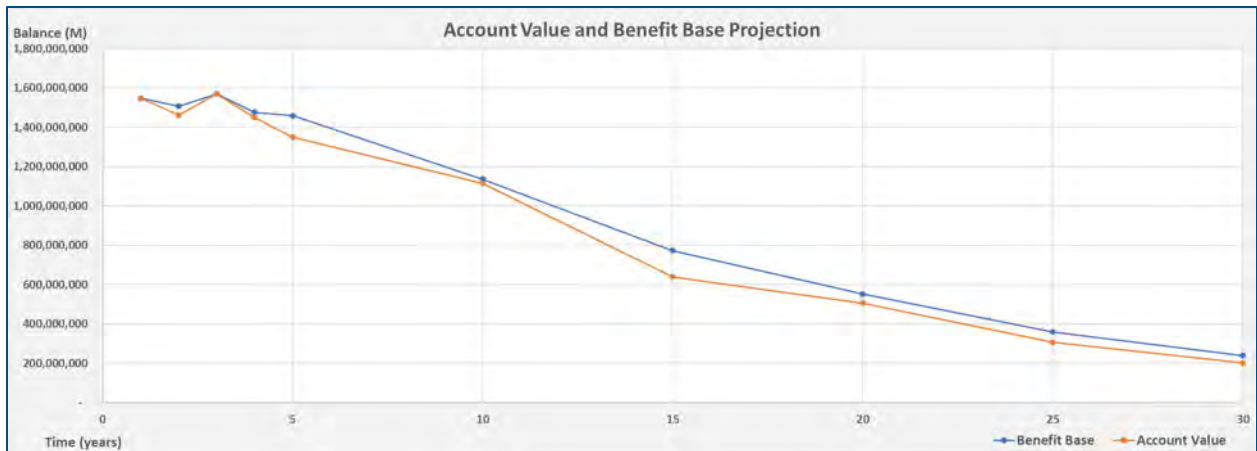
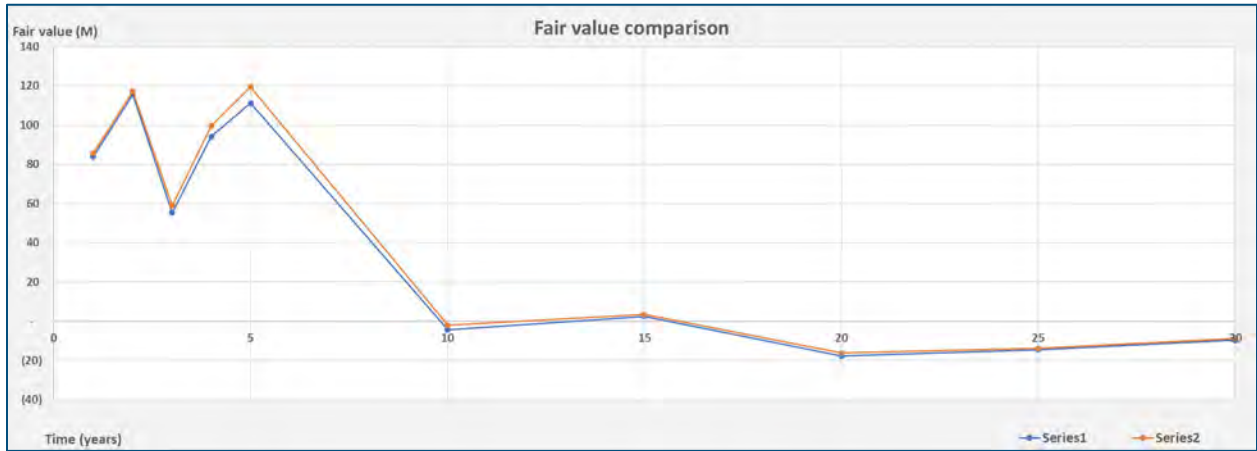


Figure 35
SCENARIO 3

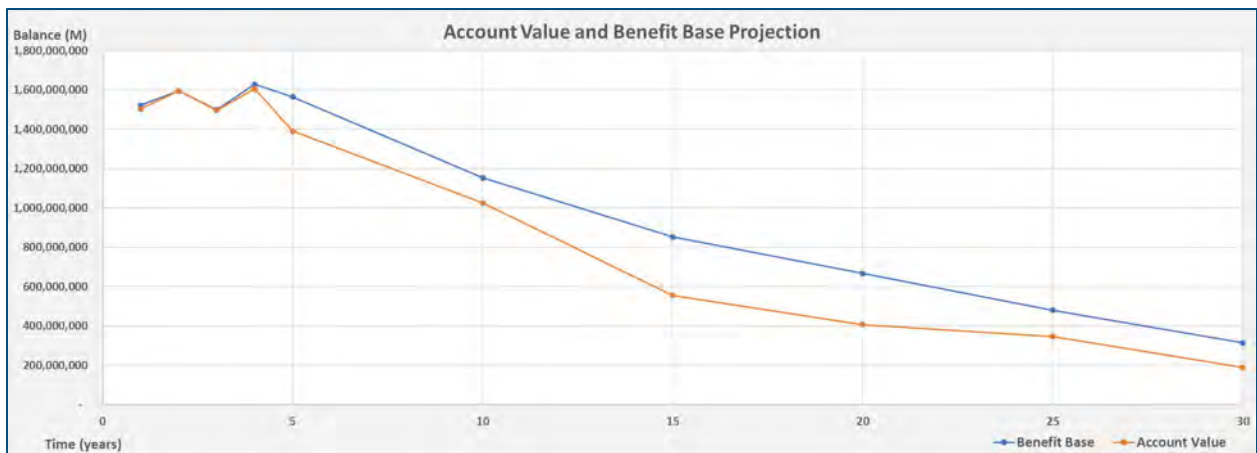


Figure 36
SCENARIO 4

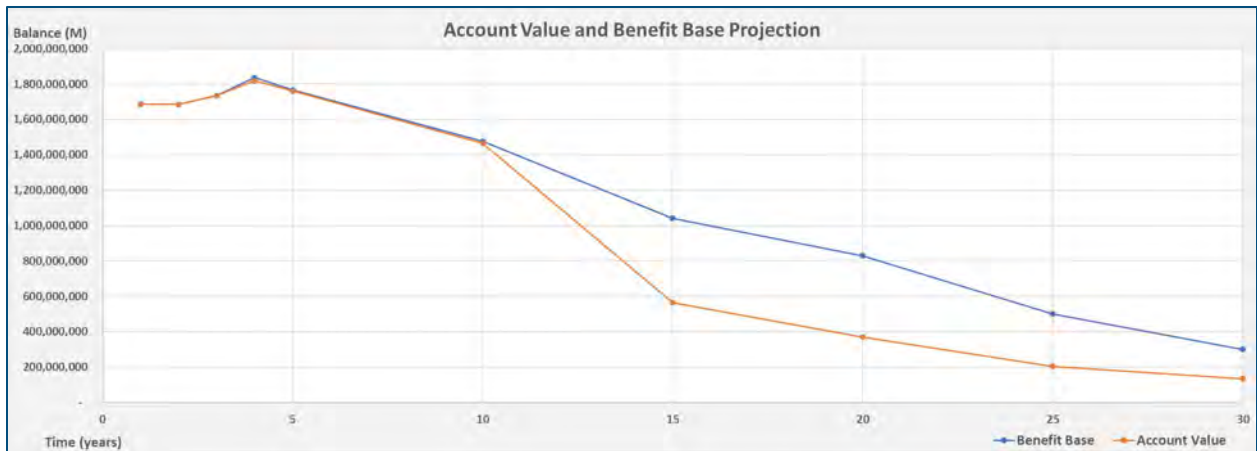
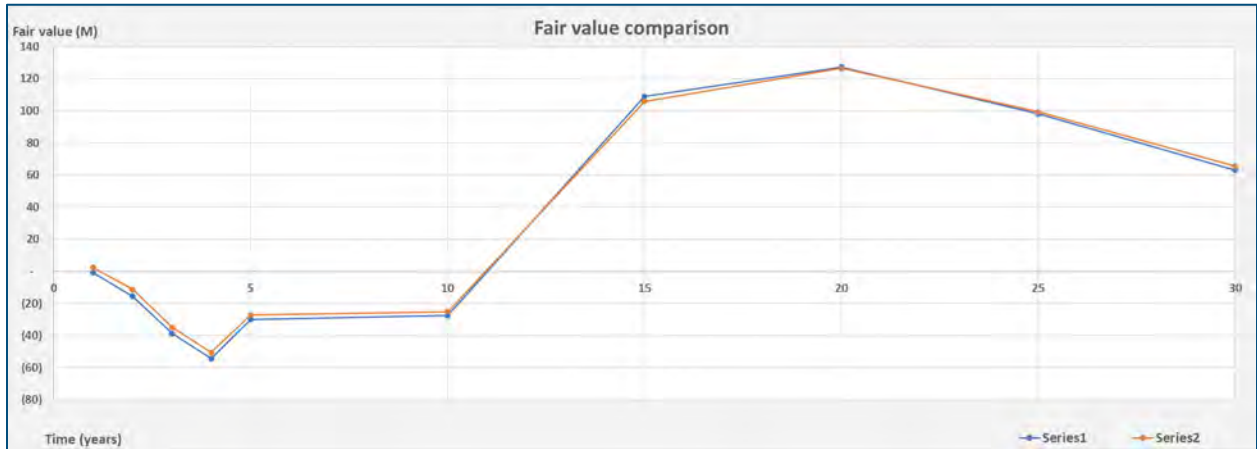
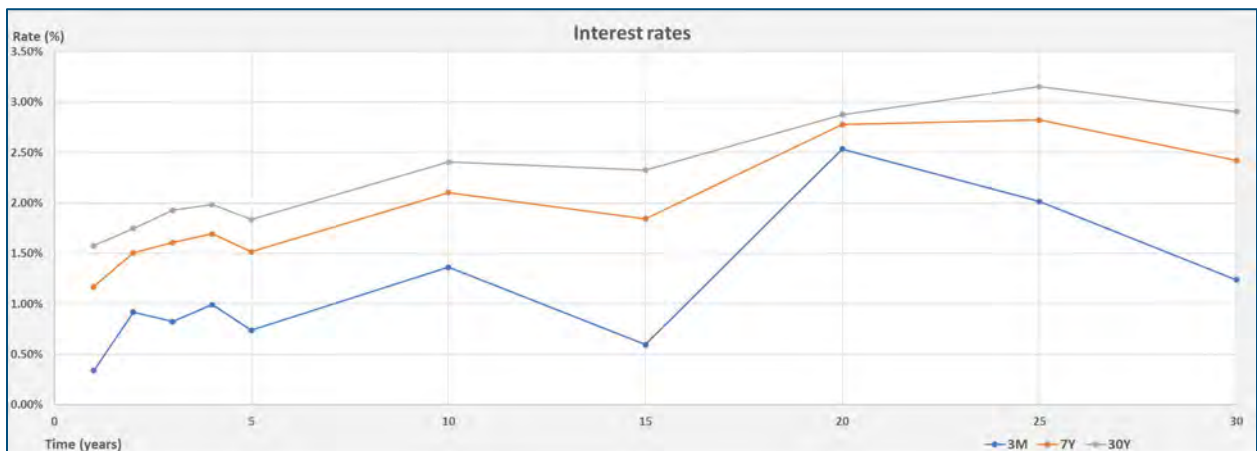
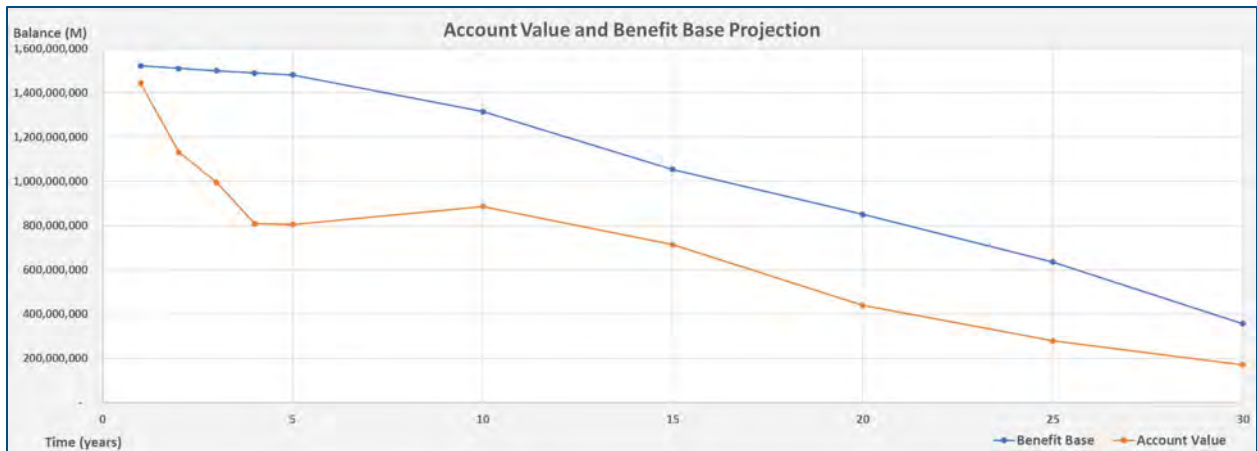


Figure 37
SCENARIO 5



6.4 CONCLUSIONS

This case study demonstrated how the model development methodology detailed in this paper can be applied to develop high-fidelity AIML models to proxy the Monte Carlo valuation variable annuity fair values.

These models can be useful to actuaries for hedging and fair value reserve FP&A, with significant runtime savings once the AIML is trained. Throughout this case study, we demonstrated the importance of feature engineering and architecture of the structure of neural network, both of which achieved and improved the desired results with the same amount of training costs. With the right balance of curated training data generation, data science techniques, and model experiments, the case study can provide a high return on investment.

Section 7: Acknowledgments

The researchers' deepest gratitude goes to those without whose efforts this project could not have come to fruition: the Project Oversight Group for their diligent work overseeing, reviewing and editing this report for accuracy and relevance.

Project Oversight Group members:

Carlos Brioso, FSA, CERA

Alex Hookway, FSA, MAAA, CERA

Karen Jiang, FSA, CERA

Hezhong (Mark) Ma, FSA, MAAA

Sambhaji Shedale, FSA

Andy Smith, FSA, MAAA

Feng Sun, FSA, CERA

Andrei Titioura, FSA, FCIA, MAAA

At the Society of Actuaries Research Institute:

Korrel Crawford, Senior Research Administrator

David Schraub, FSA, MAAA, CERA, AQ, Senior Practice Research Actuary

Appendix A: Variable Annuity Product Specifications

Two variable annuity (VA) products are modeled for this exercise:

- **VA with enhanced guaranteed minimum death benefit (GMDB):** Provides for a minimum benefit upon death. The minimum death benefit is defined by the benefit base, whereas the beneficiary will receive the greater of the account balance and the benefit base upon death.
- **VA with guaranteed lifetime withdrawal benefit (GLWB) and return of premium (ROP) death benefit:** Provides for longevity protection through lifetime income. The lifetime income is calculated based on the benefit base and annual guaranteed withdrawal rates. This feature also provides a basic ROP death benefit guarantee. This document will simply refer to this product as a GLWB.

A1: BASE PRODUCT FEATURES

The table below summarizes the base product features:

Product Feature	Specifications	Commentary
Expense charges	Per policy – annual total (\$): \$100 M&E Charge: 1.3% annually (charged daily)	Expense charges are deducted from the account balance over time.
Guaranteed minimum annuitization feature ²	N/A	This feature is typically priced far out of money and does not generate claims. For simplicity, this feature was ignored for this exercise.
Surrender charges	7-year schedule: <ul style="list-style-type: none"> • 1st Policy Year: 7% • 2nd Policy Year: 6% • 3rd Policy Year: 5% • 4th Policy Year: 4% • 5th Policy Year: 3% • 6th Policy Year: 2% • 7th Policy Year: 1% <p><i>No charges on surrender starting at the 8th policy year.</i></p> <p><i>Surrender charges calculated on fund value – free partial withdrawal amount.</i></p>	Surrender charge schedule representative of the industry

² This benefit is universal to all variable annuity contracts. The policyholder can purchase an income annuity with guaranteed interest rate and mortality parameters.

Product Feature	Specifications	Commentary
Annual free partial withdrawals	10% free partial withdrawal per year based on policy year start account value.	Common product feature
Commissions	10% upfront commission with 25bps trail commission after the surrender charge period.	

In addition, the following fund will be available for the policyholder:

Fund	Specifications
60/40 balanced fund	Fund expense ratio: 100bps (charged daily) Equity indices (60%): S&P 500 Bond funds (40%): US Long-Term Corporate Bond index

A2: GUARANTEED MINIMUM DEATH BENEFIT

The GMDB benefit upon death is calculated as follows for the GMDB only product:

$$GMDB\ Benefit = (Benefit\ Base - Account\ Value)_{0+}$$

For VA with GMDB only, the benefit base is equal to a maximum of:

1. Initial premium
2. Annual ratchet (up to age 75 or 20 years into the contract, whichever comes first)
3. Rollup at 3% (up to age 75 or 20 years into the contract, whichever comes first)

The benefit base is reduced dollar for dollar by net withdrawals below the free partial withdrawal amount. Excess withdrawals above the free partial withdrawal amount typically reduce the benefit base proportionally. For simplicity, we are not modeling excess withdrawals for VA with GMDB only.

The fees are structured as follows:

Issue Age	Male	Female
40 and under	0.65%	0.62%
41 to 45	0.85%	0.82%
46 to 48	0.95%	0.92%
49 to 52	1.10%	1.06%
53 to 55	1.40%	1.36%
56 to 58	1.50%	1.46%
59 to 62	1.70%	1.65%
63 to 67	1.95%	1.89%
68 to 72	2.00%	1.94%
73 and over	2.10%	2.03%

A3: GUARANTEED LIFETIME WITHDRAWAL BENEFIT

The GLWB guaranteed income is calculated as follows:

$$\text{Guaranteed Income} = (\text{Benefit Base} \times \text{Guaranteed Withdrawal Rate})_{0+}$$

The benefit base is defined as the greater of:

1. Initial premium
2. Annual ratchet (up to age 75, 20 years into the contract, or first lifetime withdrawal, whichever comes first)
3. Rollup at 3% (up to age 75, 20 years into the contract, or first lifetime withdrawal, whichever comes first)

The fee is structured as follows:

1. 1.25% (applied to benefit base quarterly where each quarter (1/4) of the fee is deducted from the account value)

The ROP death benefit base is reduced dollar for dollar by net withdrawals.

The guaranteed withdrawal rates vary between 3.30% and 5.60% annually, increasing with the age at the time of the first lifetime withdrawal:

Age at First Withdrawal	Male	Female
44 and under	0.00%	0.00%
45 to 59	3.30%	3.30%
60 to 64	3.80%	3.80%
65 to 69	4.50%	4.50%
70 to 74	4.60%	4.60%
75 to 80	4.80%	4.80%
81 to 84	5.20%	5.20%
85 and over	5.60%	5.60%

Required minimum distributions (RMD) were ignored for simplicity.

Appendix B: Variable Annuity Actuarial Assumptions

The following actuarial assumptions were used for this exercise:

- Appendix B1: Premiums
- Appendix B2: Mortality
- Appendix B3: Lapse
- Appendix B4: Partial withdrawal behavior
- Appendix B5: Maintenance expenses
- Appendix B6: Reinvestment strategy

Other actuarial assumptions such as company expense were not needed for the exercise.

B1: PREMIUMS

We assumed that the policies were based on a single premium of \$125,000 at issue. No other premiums were paid.

B2: MORTALITY

The 2012 IAR table was used to model mortality rates. The 2012 IAR table includes both base mortality and mortality improvement rates.

The 2012 IAR table (SOA, 2012) is an industry table commonly used as a basis for annuity products. It is common for companies to apply additional adjustments to this base table to accurately model company-specific mortality experiences; we will ignore such adjustments for the ease of use and understanding.

The guaranteed minimum death benefit rider described in [A2: Guaranteed minimum death benefit](#) was further adjusted by a multiplier of 125%. This adjustment reflects the expectation that enhanced death benefit riders would generally experience higher mortality than a living benefit rider such as the one described in [A3: Guaranteed lifetime withdrawal benefit](#).

B3: LAPSE

Dynamic lapse rates for GLWB and GMDB contracts are calculated based on the in-the-moneyness (ITM) of the contract value, where the ITM is defined as a ratio between the guaranteed value and the contract account value. For GMDBs, the guaranteed value is the minimum guaranteed death benefit. For GLWBs, the guaranteed value is the total guaranteed withdrawal base.

We used the following assumptions consistent with the full surrender actuarial assumptions from VM-21:

ITM	In Surrender Charge Period (first 7 policy years)	Shock Lapse (policy year 8)	After Surrender Charge Period (policy years 9 and over)
Under 50%	4.0%	25.0%	15.0%
50-75%	3.0%	18.0%	10.0%
75-100%	2.5%	12.0%	7.0%
100-125%	2.5%	8.0%	4.5%
125-150%	2.5%	6.0%	3.0%
150-175%	2.5%	5.0%	2.5%
175-200%	2.5%	4.5%	2.0%
Over 200%	2.5%	4.0%	2.0%

The initial ITM ratio of the contract was further modified by adjustment factors depending on the contract type. The adjustment factors are defined as follows:

- For GMDB contracts, the ITM is 90% of the ratio between the guaranteed value and the contract account value.
- For all GLWBs, the ITM is calculated as 75% of the ratio between the guaranteed value and the contract account value.

For example, a VA contract with a GMDB rider entering the last year of the surrender charge period has an account value of \$100,000 and a minimum guaranteed death benefit of \$90,000. The ITM is then equal to:

$$ITM = 90\% * \left(\frac{90,000}{100,000} \right) = 81\%$$

Based on the standard table, the VA's full surrender / lapse rate for the current policy year is 2.5%.

B4: PARTIAL WITHDRAWAL BEHAVIOR

Partial withdrawal assumptions control how much policyholders withdraw from their account each year. This differs from the lapse assumption above, which indicates the probability that the policyholder closes their contract in return for the cash surrender value. Note that a withdrawal of the full account balance would generally be equivalent to a lapse.

The proposed partial withdrawal actuarial assumption generally follows the guidance of VM-21's standard scenario prescribed assumption. Overall, VM-21 standard scenario assumptions are aggressive (tend to result in lower CTEs) because they are intended to serve as a reserve floor. We used these assumptions for simplicity reasons.

For VAs with GMDB rider only, a partial withdrawal assumption is not a significant driver of risk. For those contracts, we used a partial withdrawal rate of 2% of account value per year.

For VAs with GLWB rider, partial withdrawal behavior is an important assumption for GLWB reserve calculation. The assumption has two parts:

- Time to first withdrawal (withdrawal cohorts)
- Rate of withdrawal (and efficiency of withdrawal)

We used VM-21's standard scenario prescribed methodology to build initial withdrawal cohorts, each representing a different time to first withdrawal. The details of this calculation are included in the attachment below and the

following simplifying assumptions are made (NAIC, 2023):

- Policy is non-tax qualified, Male
- Use annuity factor as a proxy for guaranteed actuarial present value (GAPV)
- Rounded to nearest 5%
- No immediate withdrawals for issue ages less than 65

The valuation model splits the policy into different withdrawal cohorts such that the total account value and benefit base are preserved. Actuarial calculations were done for each cohort separately and aggregated at the end. The table below specifies the weights of each withdrawal cohort by issue age.

GLWB Issue Age	Years to First Withdrawal							
	Never	0	5	10	15	20	25	30
40	0.20		0.20	0.30	0.05	0.10	0.10	0.05
45	0.20		0.15	0.30	0.15	0.10	0.05	0.05
50	0.20		0.15	0.40	0.15	0.05	0.05	
55	0.20		0.30	0.35	0.10	0.05		
60	0.20		0.45	0.25	0.05	0.05		
65	0.20	0.30	0.25	0.20	0.05			
70	0.20	0.20	0.40	0.15	0.05			

For example, a VA with GLWB issued to a policyholder, aged 65, will have the following withdrawal cohort assumptions:

- 20% of the cohort will never take a withdrawal
- 30% of the cohort will start taking withdrawals immediately
- 25% of the cohort will start taking withdrawals in year 5
- 20% of the cohort will start taking withdrawals in year 10
- 5% of the cohort will start taking withdrawals in year 15

Once a policy starts taking withdrawals, we assume 100% withdrawal efficiency meaning the policyholder will continue taking withdraws indefinitely at the guaranteed withdrawal rates specified in [A3: Guaranteed lifetime withdrawal benefit](#).

B5: MAINTENANCE EXPENSES

The internal maintenance expenses assumption consists of:

- Per policy expense of \$125 as of December 31st, 2020. The expense is assumed to increase by 2% each year thereafter.
- Twelve basis points of the projected account value for each year in the projection.

B6: REINVESTMENT STRATEGY

The reinvestment strategy is modeled as a rolling portfolio yield where the portfolio yield (P) at any month M is calculated as follows:

$$P_M = P_{M-1} \cdot (1 - W)^{1/12} + NMR_M \cdot [1 - (1 - W)^{1/12}]$$

where W is the annual portfolio turnover rate and NMR_M is the new money rate this month. The new money rate is calculated by weighting the 10-point US Treasury curve based on the weights below and a spread of 2%:

3 Month	6 Month	1 Year	2 Year	3 Year	5 Year	7 Year	10 Year	20 Year	30 Year
5.00%	5.00%	7.50%	7.50%	7.50%	10.00%	35.00%	10.00%	7.50%	5.00%

Appendix C: ASOP Review

ASOP 56 – Modeling

Actuarial models are the fundamental components of machine learning solutions. As such, actuaries should ensure models used in the solutions suffice the standard to the extent of the actuary’s responsibilities with respect to designing, developing, selecting, modifying, using, reviewing, or evaluating them.

ASOP 56 is directly applicable to the development, training, and validation of any models used for actuarial purposes and this would extend to the AIML models discussed in this paper.

Excerpt from ASOP 56:

This standard applies to actuaries in any practice area when performing actuarial services with respect to designing, developing, selecting, modifying, or using all types of models. For example, an actuary using a model developed by others in which the actuary is responsible for the model output is subject to this standard.

Modeling actuaries should be intimately familiar with the requirements and guidance contained in it. The ASOP contains specific provisions related to predictive models and predictive modeling techniques, which proxy models are, such as withholding a subset of data generated from the calibration process to use for later testing.

The ASOP provides useful guidance for all models, most of which are directly applicable to AIML models, in particular, the following subsections in [section 3](#). Analysis of Issues and Recommended Practices apply to this paper’s use cases:

Section	Name	Commentary
3.1	Model Meeting the Intended Purpose	See breakdown below
3.1.1	Designing, Developing, or Modifying the Model	<ul style="list-style-type: none"> Actuaries should use their professional judgment to confirm that the capability of the model is consistent with the intended purpose. Actuaries using AIML proxy models should confirm that the model used and as calibrated is consistent with the ultimate purpose or context in which the model will be used.
3.1.2	Selecting, Reviewing or Evaluating the Model	<ul style="list-style-type: none"> <i>As-is from ASOP: “When selecting, reviewing, or evaluating the model, the actuary should confirm that, in the actuary’s professional judgment, the model reasonably meets the intended purpose.”</i> Similar comment as above for AIML proxy models.
3.1.3	Using the Model	<ul style="list-style-type: none"> Actuaries should make reasonable efforts to confirm that the use of the model is consistent with its intended purpose. The AIML proxy models are calibrated to perform over a specific range in a specific context. It is important that actuaries make a reasonable effort to confirm that the AIML model is used in the appropriate context.

Section	Name	Commentary
		<ul style="list-style-type: none"> Actuaries who are not familiar with the model run the risk of using the AIML model outside of its intended purposes if the right controls are not in place.
3.1.4	Model Structure	<ul style="list-style-type: none"> Actuaries should assess whether the structure of the model (including judgments reflected in the model) is appropriate for the intended purpose. Notable items for AIML proxy models include: <ul style="list-style-type: none"> Approximate design of the AIML proxy model within the end-to-end architecture of the process. Appropriate form of the model – in particular, using a model that provides the precision needed with the minimum level of complexity. Level of detail or aggregation level of the proxy (e.g., seriatim or aggregated) Representing options – including not only features that have predictive power, but also features that represents the inputs that would reasonably be changed or shocked by the actuary.
3.1.5	Data	<ul style="list-style-type: none"> <i>As-is from ASOP: The actuary should use, or confirm use of, data appropriate for the model's intended purpose and should refer, as applicable, to ASOP No. 23, Data Quality, when selecting, reviewing, or evaluating data used in the model, either directly or as the basis for deriving, estimating, or testing assumptions used in the model.</i> Actuaries should pay attention to the data used to train and test the proxy model. In particular, the performance of the AIML proxy model is heavily dependent on the quality of the data generated. This includes the appropriate range and distribution of the samples generated. Actuaries should also check that the data provided to the model when generating the proxy after calibration is adequate.
3.1.6	Assumptions Used as Input	<ul style="list-style-type: none"> This section is less applicable as the proxy models detailed in this paper will generally be free of actuarial assumptions. However, some exceptions may arise. Actuarial assumptions are applicable to the source model used to generate the training and testing data.
3.2	Understanding the Model	<ul style="list-style-type: none"> When expressing opinions or communicating results of the model, actuaries should understand the important aspects of the model, known weaknesses or limitations and limitations in data. This is applicable to AIML models as these proxy models

Section	Name	Commentary
		<p>will generally (some exceptions when it offsets lack of convergence), by nature, introduce additional error in the model output relative to the first principles calculation.</p> <ul style="list-style-type: none"> Actuaries should also be aware of the calibration range limitations and be able to communicate the uncertainty in the output if such a situation is ever needed from the actuary.
3.3	Reliance on Data or Other Information Supplied by Others	<ul style="list-style-type: none"> AIML proxy models are reliant on the data provided by the actuarial model.
3.4	Reliance on Models Developed by Others	<ul style="list-style-type: none"> <i>Fully applicable with no unique considerations for AIML proxy models – refer to ASOP. When actuaries are relying on models developed by others, the actuary should make practical efforts to comply with other applicable sections of this standard.</i>
3.5	Reliance on Experts	<ul style="list-style-type: none"> <i>Fully applicable with no unique considerations for AIML proxy models – refer to ASOP. Actuaries may rely on experts for these models as well.</i>
3.6	Evaluation and Mitigation of Model Risk	<ul style="list-style-type: none"> <i>As-is from ASOP: The actuary should evaluate model risk and, if appropriate, take reasonable steps to mitigate model risk.</i> Per the ASOP, the type and degree of model risk depends on the intended purpose; nature and complexity of the model; operating environment and governance and controls related to the model; whether there have been changes to the model or its operating environment; and the balance between the cost of mitigation efforts and the reduction in potential model risk. Actuaries should realize that the introduction of AIML proxy models, as with any other proxy model, introduces additional model risk. The additional model risk is due to the need to monitor the model performance and the risk that the proxy model is not well understood by ultimate users and used outside of the intended purpose, resulting in actuaries relying on inaccurate proxies. The AIML proxy models discussed in this paper are complex in nature and can be difficult to interpret, which adds to the overall model risk. AIML proxy models require establishing an appropriate control and governance structure to reduce the risk that the model is used outside of its intended purpose.
3.6.1	Model Testing	<ul style="list-style-type: none"> <i>Combined commentary on 3.6.1 and 3.6.2 as they are</i>

Section	Name	Commentary
3.6.2	Model Output Validation	<p><i>closely related for AIML proxy models.</i></p> <ul style="list-style-type: none"> • The actuary should perform appropriate testing and output validation to get comfortable with the performance of the AIML proxy model. • This testing should include detailed out of sample validation against the actuarial model. The out of sample testing should cover the range of intended uses of the model, as well as running sample use cases with full comparisons to the original first principles calculations from the actuarial model.
3.6.3	Review by Another Professional	<ul style="list-style-type: none"> • <i>Fully applicable with no unique considerations for AIML proxy models.</i>
3.6.4	Reasonable Governance and Controls	<ul style="list-style-type: none"> • As discussed in the points above, the actuary should establish strong governance and control for the AIML model to mitigate the risk of using the AIML model outside of the pre-defined range. • This commentary applies to 3.6.4 and 3.6.5.
3.6.5	Mitigating Misuse and Misinterpretation	
3.7	Documentation	<ul style="list-style-type: none"> • <i>Fully applicable with no unique considerations for AIML proxy models.</i>

ASOP 21 – Responding to Auditors or Examiners

To the extent that machine learning models become a component of the valuation or financial reporting processes, the financial planning process that results in public projections, or becomes a material part of significant financial or operational trade decisions, actuaries are required to be prepared to share with auditors the information necessary for them to conduct their audit or examination. The actuary should be prepared to provide evidence and demonstrate that they followed the guidance given in ASOP 56. This may include providing the data used to calibrate the models, the assumptions used and judgment applied, the methods used, the models used, the controls around the models, processes, and procedures, and the reasoning to support the results and conclusions. Since these models are often used to replace portions of other models, the actuary should be prepared to share the testing performed, including the results, approaches and methods, scope of testing, and raw data with the auditor or examiner.

Additionally, the actuary should be prepared to discuss with the auditor or examiner any change in circumstances that, in the judgment of the actuary, affect the use of the machine learning models or the value or accuracy of the output in the process being examined.

ASOP 22 – Statements of Actuarial Opinion Based on Asset Adequacy Analysis

Machine learning proxy models could offer great value to insurance companies in performing the calculation intensive elements of AAT, particularly in helping project out future portfolio trades and management actions. Since these projections are done, by nature, over time, the actuary should perform appropriate testing to be confident in the robustness of the model over time and, as the book of business ages and changes, to opine on the reasonableness of the results. Following ASOP 21, they should also be prepared to disclose to the auditor the

validation and testing done to ensure the robustness of the model, any assumptions or approximations used with the output in the AAT process, and the controls to ensure the completeness and accuracy of the output calculations.

ASOP 23 – Data Quality

Data quality is a key area of focus for the actuary as they look to incorporate machine learning models. Since these models are one step away from the actuarial models, tracing the calculations is more difficult. As such, it is incumbent upon the actuary to ensure the quality of the data used in the actuarial models when calibrating the proxy model, as well as the quality of the output data being consumed by the proxy model during the calibration process. We would argue “wholly hypothetical dataset” exception doesn’t really apply as the actuary may be required to develop data outside of the scope existing within in-force policies to ensure the robustness of the model for future use. The actuary will need to opine on whether the data used to calibrate the model is appropriate and fit, robust enough, and any restrictions in the data that would create limitations on the use of the proxy model. The actuary should be prepared to disclose any reviews of the data performed in ensuring the above.

ASOP 24 – Compliance with NAIC Life Illustrations Model Regulation

To the extent that proxy models are used as inputs into life illustrations, or used to produce life illustrations, the actuary needs to be prepared to show that the outputs meet all of the requirements under the NAIC Life model regulation, or an specific actuarial guidelines related to life illustrations. As such, the actuary should take steps to ensure that the datasets used to calibrate and train the model or the outputs from the model incorporate all of the constraints placed on the output by the model regulation (e.g., illustrated scale cannot be more favorable than current payable scale, interest crediting for indices are limited by AG49). The actuary should understand how the underlying models used to train the proxy model respond to significant changes in market conditions, policyholder demographics, and other dimensions the proxy model is trained on to understand any limitations on the use of the proxy model for this purpose.

ASOP 41 – Actuarial Communications

The actuary is responsible for all assumptions and methods used in actuarial communications. Because proxy models can sometimes be trained for specific purposes, the actuary should discuss any limitations on the analysis presented or constraints on its use. To the extent that proxy models built by others were relied upon for the analysis of actuarial communications, the actuary must state their reliance upon other qualified individuals, disclose if they were not able to perform sufficient analysis to endorse the robustness or accuracy of the proxy models used, or if the output, assumptions, or methods used conflict with their professional judgment on what is reasonable and appropriate.

ASOPs 46 and 47 – Risk Evaluations for Enterprise Risk Management and Risk Treatment for Enterprise Risk Management

Proxy models can be of great value in reducing the runtime of enterprise risk and economic capital models. When proxy models are incorporated into the process, the actuary needs to consider all of the normal review elements for models used in an enterprise risk context (reproducibility, adaptability to new risks, useability, reliability, limitations, etc.) for both the proxy model and the models used to train the proxy model. The actuary also needs to review the appropriateness and robustness of the approach, design, data used, methodologies and simplifications used in the training of the proxy model. The actuary needs to fully understand those limitations as they start to use that information for other purposes, including hedging, ALM, or other risk mitigating activities.

ASOP 52 – PBR for Life

Principles-based reserving contains a number of modeling elements meant to accurately reflect how the policy and assets backing the policy will perform throughout the policy lifecycle. Proxy models can serve as a valuable tool replacing those elements to significantly shorten runtimes required to accurately model the business. The actuary needs to validate the robustness and accuracy of the proxy models for the purpose intended. In addition to the procedures performed above, ASOP 52 recommends the actuary perform dynamic validation of the model by comparing cash flows produced by the model to actual historical data, evaluating consistency of the results with other systems producing similar calculations, and evaluating the reasonableness of the results, including the ability to explain why the results are changing.

ASOP 54 – Pricing of Life Insurance and Annuity Products

Proxy models can reduce the calculation intensity of pricing models, thereby potentially enabling analysis to be performed more granularly and for a longer time period than other approaches. ASOP 54 requires the actuary to validate that the calculations from the model are consistent without outside calculations, that they reasonably reflect the real financial impact of the product and associated management actions over the life of the policy, and that the models, assumptions, methodologies, and data used are fit for purpose.

ASOP 55 – Capital Adequacy Assessment

ASOP 55 is meant to be an extension of ASOPs 46 and 47 specific to modeling capital adequacy in response to regulatory requirements around ORSAs. Proxy models would provide similar benefits and require similar considerations to those discussed in ASOPs 46 and 47. Because the cost of running the models (in time and computing costs) are lower with proxy models, the actuaries can produce more analysis. The actuary needs to ensure that they understand all of the limitations inherent in the training of the proxy models, including assumptions and methodologies used in the underlying actuarial models and the approach and design of the calibration, to ensure that they can rely on the models for the purposes intended, and that they are still appropriate to model the more extreme scenarios used in ORSAs. For example, during the financial crisis, liquidity in options markets dried up and bid/ask spreads significantly widened, resulting in disconnects in option valuation and the inability to trade assets, some situations that may not have been contemplated or incorporated into the training of the underlying proxy models, and which may require judgment and adjustments to the results for them to be relied upon.

References

- AAA. Principle-Based Reserving: A New Way to Insure for Life. AAA.org, Jul 2016. URL: <https://www.actuary.org/node/13473>
- AAA. Economic Scenario Generators. AAA.org, n.d.. URL: <https://www.actuary.org/content/economic-scenario-generators>
- ASB. Actuarial Standards Board. URL: <http://www.actuarialstandardsboard.org/standards-of-practice/>
- Beach, Van, Alexandre Boumezoued, and Josh Dobiac. Cloud Computing and Machine Learning Uses in the Actuarial Profession. SOA.org, July 2019. URL: <https://www.soa.org/globalassets/assets/files/resources/research-report/2019/cloud-computing.pdf>
- Beeson, Michael, Stephen Blaske, and Nathan Campbell. Scenario and Cell Model Reduction. AAA.org, Sep 2010. URL: <https://www.actuary.org/sites/default/files/files/Scenario%20and%20Cell%20Model%20Reduction%20PN%20Final%20092210.4.pdf/Scenario%20and%20Cell%20Model%20Reduction%20PN%20Final%20092210.4.pdf>
- Black, Fischer, and Scholes Myron Scholes. 1973. The Pricing of Options and Corporate Liabilities. Journal of Political Economy. 81 (3): 637–654.
- Botev, Zdravko, and Ad Ridder. Variance Reduction. Wiley online library, 2017. URL: <http://doi:10.1002/9781118445112.stat07975>
- Broadie, Mark, Yiping Du, and Ciamac Moallemi. Efficient Risk Estimation via Nested Sequential Simulation. Moallemi.com, Dec 2010. URL: <https://moallemi.com/ciamac/papers/segrisk-2010.pdf>
- Broadie, Mark, Yiping Du, and Ciamac Moallemi. Risk Estimation via Regression. Moallemi.com, Jul 2015. URL: <https://moallemi.com/ciamac/papers/regression-risk-2011.pdf>
- Burns, Eileen, Gene Dan, and Anders Larson. Considerations for Predictive Modeling in Insurance Applications. SOA.org, May 2019. <https://www.soa.org/49bcd8/globalassets/assets/files/resources/research-report/2019/considerations-predictive-modeling.pdf>
- Columbia AI. Artificial Intelligence (AI) vs. Machine Learning. Columbia.edu, n.d.. URL: <https://ai.engineering.columbia.edu/ai-vs-machine-learning/>
- Czernicki, Dave, Peter Carlson, and Jean-Philippe Larochelle. AIML: Cut Through the Noise. Theactuarmagazine.org, Spring 2020. URL: <https://www.theactuarmagazine.org/cut-through-the-noise/>
- Dang, Ou, Mingbin Feng, and Mary Hardy. Efficient Nested Simulation for Conditional Tail Expectation of Variable Annuities. SSRN, Jun 2019. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3045430
- Efron, Bradley, and Robert Tibshirani. 1993. An Introduction to the Bootstrap. Boca Raton, FL: Chapman & Hall/CRC.
- EIOPA. Solvency II. Jan 2016. URL: https://www.eiopa.europa.eu/browse/regulation-and-policy/solvency-ii_en#ref-2020-solvency-ii-review
- FASB. FASB in Focus Insurance ED Rev (9-16). FASB.org, September 2016. URL: [https://www.fasb.org/Page/ShowPdf?path=FASB%20in%20Focus%20Insurance%20ED%20Rev%20\(9-16\).pdf](https://www.fasb.org/Page/ShowPdf?path=FASB%20in%20Focus%20Insurance%20ED%20Rev%20(9-16).pdf)
- Feng, Runhuan. A Comparative Study of Risk Measures for Guaranteed Minimum Maturity Benefits by a PDE

Method. SSRN, Jun 2014. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2432640

Feng, Runhuan, Zhenyu Cui, and Peng Li. Nested Stochastic Modeling for Insurance Companies. SOA, Nov 2016. URL: [Nested Stochastic Modeling for Insurance Companies Report \(soa.org\)](https://www.soa.org/globalassets/assets/files/resources/research-report/2016/nested-stochastic-modeling-for-insurance-companies-report-soa.org)

Feng, Ben, Zhenni Tan, and Jiayi Zheng. 2020. Efficient Simulation Designs for Valuation of Large Variable Annuity Portfolios. North American Actuarial Journal. Volume 24, 2020 - Issue 2: Predictive Analytics

Gan, Guojun, and Sheldon Lin. Valuation of large variable annuity portfolios under nested simulation: A functional data approach. Insurance: Mathematics and Economics. SSRN, Dec 2015. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2358231

Gnedenko, B.V., and A. Kolmogorov. 1954. Limit distributions for sums of independent random variables. Cambridge: Addison-Wesley.

Goodfellow, Ian, Yoshua Bengio and Aaron Courville. Deep Learning. n.d.. URL: <https://www.deeplearningbook.org/>

Gordy, Michael, and Sandeep Juneja. 2008. Nested Simulation in Portfolio Risk Measurement. Federal Reserve Board.

Hull, John. 2021. Fundamentals of Futures and Options Market. Pearson.

IFRS. IFRS 17 Insurance Contracts. IFRS.org. 2023. URL: <https://www.ifrs.org/issued-standards/list-of-standards/ifrs-17-insurance-contracts/#about>

Keras. Deep Learning for Humans. Keras, n.d.. URL: <https://keras.io/>

Koissi, Marie-Claire, Herschel Day, and Vicki Whitledge. Emerging Data Analytics Techniques with Actuarial Applications. Society of Actuaries, July 2019. URL: <https://www.soa.org/4a1369/globalassets/assets/files/resources/research-report/2019/emerging-analytics-techniques-applications.pdf>

Lin, Sheldon, and Shuai Yang. Fast and Efficient Nested Simulation for Large Variable Annuity Portfolios: A Surrogate Modeling Approach. SSRN, Mar 2019. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3342701

Lin, Sheldon, and Shuai Yang. Efficient Dynamic Hedging for Large Variable Annuity Portfolios with Multiple Underlying Assets. SSRN, Mar 2020. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3550106

NAIC. Valuation Manual. NAIC.org 2023. URL: https://content.naic.org/sites/default/files/pbr_data_valuation_manual_current_edition.pdf

Pedersen, Hal, Mary Campbell, and Stephan Christiansen. Economic Scenario Generators: A Practical Guide. SOA.org, Jul 2016. URL: <https://www.soa.org/globalassets/assets/Files/Research/Projects/research-2016-economic-scenario-generators.pdf>

Raden, Neil. Ethical Use of Artificial Intelligence for Actuaries. SOA.org, Sep 2019. URL: <https://www.soa.org/globalassets/assets/files/resources/research-report/2019/ethics-ai.pdf>

Rosner, Bruce, and Mark Freedman. How Fair Value Measurement Changes Risk Management Behavior in the Insurance Industry. SOA.org, Jan 2013. URL: <https://www.soa.org/49387b/globalassets/assets/files/research/projects/research-how-fair-value-report.pdf>

Smith, Logan, Emma Pirchalski, and Ilana Golbin. Avoiding Unfair Bias in Insurance Applications of AI Models.

SOA.org, Aug 2022. URL: <https://www.soa.org/4a36e6/globalassets/assets/files/resources/research-report/2022/avoid-unfair-bias-ai.pdf>

SOA. Individual Annuity Reserving Report and Table. SOA.org, 2012. URL: <https://www.soa.org/resources/experience-studies/2011/2012-ind-annuity-reserving-rpt/>

SOA. Economic Scenario Generators. SOA.org, Aug 2022. URL: <https://www.soa.org/resources/tables-calcs-tools/research-scenario/>

SOA. Predictive Analytics Certificate Program. SOA, 2023. URL: <https://www.soa.org/programs/predictive-analytics-certificate/program-dates/>

Windsor, Conn. Secure Retirement Institute: Total Annuity Sales Jump 16% in 2021 — Marking Highest Sales Since 2008. LIMRA.com, Jan 2022. URL: <https://www.limra.com/en/newsroom/news-releases/2022/secure-retirement-institute-total-annuity-sales-jump-16-in-2021--marking-highest-sales-since-2008/#:~:text=27%2C%202022%E2%80%94Total%20U.S.%20annuity,U.S.%20Individual%20Annuity%20Sales%20Survey.>

Yeo, Nicholas, Raymond Lai, and Jyeh Ooi. Literature Review: Artificial Intelligence and Its Use in Actuarial Work. SOA.org, Dec 2019. URL: <https://www.soa.org/globalassets/assets/files/resources/research-report/2019/ai-actuarial-work.pdf>

Yuan, Jiaming. xgboost: Extreme Gradient Boosting. Cran.r-project.org, Mar 2023. URL: <https://cran.r-project.org/web/packages/xgboost/xgboost.pdf>

Zhang, Victoria. A Tour of AI Technologies in Time Series Prediction. SOA.org, Jul 2019. URL: <https://www.soa.org/globalassets/assets/files/resources/research-report/2019/tour-ai-technologies.pdf>

These third-party links are being provided for informational purposes only. The content in these third-party links do not necessarily reflect the opinions of the Society of Actuaries or the Education and Research Section. Neither the Society of Actuaries nor the Education and Research Section are responsible for the reliability, accuracy or content of the third-party site(s). If you have questions about the content on such sites, please contact the site directly.

About The Society of Actuaries Research Institute

Serving as the research arm of the Society of Actuaries (SOA), the SOA Research Institute provides objective, data-driven research bringing together tried and true practices and future-focused approaches to address societal challenges and your business needs. The Institute provides trusted knowledge, extensive experience and new technologies to help effectively identify, predict and manage risks.

Representing the thousands of actuaries who help conduct critical research, the SOA Research Institute provides clarity and solutions on risks and societal challenges. The Institute connects actuaries, academics, employers, the insurance industry, regulators, research partners, foundations and research institutions, sponsors and non-governmental organizations, building an effective network which provides support, knowledge and expertise regarding the management of risk to benefit the industry and the public.

Managed by experienced actuaries and research experts from a broad range of industries, the SOA Research Institute creates, funds, develops and distributes research to elevate actuaries as leaders in measuring and managing risk. These efforts include studies, essay collections, webcasts, research papers, survey reports, and original research on topics impacting society.

Harnessing its peer-reviewed research, leading-edge technologies, new data tools and innovative practices, the Institute seeks to understand the underlying causes of risk and the possible outcomes. The Institute develops objective research spanning a variety of topics with its [strategic research programs](#): aging and retirement; actuarial innovation and technology; mortality and longevity; diversity, equity and inclusion; health care cost trends; and catastrophe and climate risk. The Institute has a large volume of [topical research available](#), including an expanding collection of international and market-specific research, experience studies, models and timely research.

Society of Actuaries Research Institute
475 N. Martingale Road, Suite 600
Schaumburg, Illinois 60173
www.SOA.org