Deep calibration of financial models: Turning theory into practice

Patrick Büchel∗, Michael Kratochwil†, Maximilian Nagl‡, Daniel Rösch§

August 10, 2020

Abstract

The calibration of financial models is a laborious, time-consuming and expensive task, which needs to be performed frequently by financial institutions. Recently, the application of artificial neural networks (ANNs) for calibration has gained interest. This paper provides the first comprehensive empirical study on the application of ANNs for calibration based on observed market data. We benchmark the performance against a real-life calibration framework. We show that the results of an ANN based calibration framework are very competitive and derive guidelines for its practical implementation to enhance and accelerate managerial decisions. Furthermore, we show that our calibrated parameters are more stable over time, enabling more reliable risk reports and business decisions.

Keywords: Decision Support; Global Optimization; Deep Learning; OR in Banking; Model Calibration

JEL classification: C23, G21, G33, E43

∗ Commerzbank AG, Mainzer Landstraße 157, 60327 Frankfurt am Main, Germany, email: patrick.buechel@commerzbank.com
† Dr. Nagler & Company GmbH, Maximilianstraße 47, 80538 München, Germany, email: michael.kratochwil@nagler-company.com
‡ Corresponding author, Universität Regensburg, Chair of Statistics and Risk Management, Universitätstraße 31, 93040 Regensburg, Germany, email: maximilian.nagl@ur.de
§ Universität Regensburg, Chair of Statistics and Risk Management, Universitätstraße 31, 93040 Regensburg, Germany, email: daniel.roesch@ur.de

Electronic copy available at: https://ssrn.com/abstract=3667070
1 Introduction

The calibration of financial models is a laborious, time-consuming and expensive task performed by financial institutions on a regular basis (e.g., daily). Asset pricing models are used to determine the value of derivatives or to generate scenarios for Monte Carlo calculations in risk management. Hence, the outcomes of these models are crucial information required for investment and business decisions. The calibration of these models needs to be performed frequently to ensure the validity of their outcomes. In particular, the calibration of complex and multi-dimensional models is burdensome and requires significant computational efforts and time. The choice of an asset pricing model for a specific product involves balancing the accuracy of the model and the time required for its calibration.

The calibration of a financial model can be described as a reverse optimization task, where the inputs of a pricing function (model parameters) are determined to fit observable outputs (e.g., market prices). The solution of this problem usually requires calling a specific pricing function a huge number of times with different parameter settings. Hence, the required time and computational resources have always been limiting factors when choosing a pricing model and models with fast (semi-)analytical solutions are generally preferred. Furthermore, these limitations have led to the broad application of local optimization algorithms for calibration. The application of more advanced optimization algorithms is rarely considered. Particularly models with multiple parameters give rise to multiple minima for calibration. Hence, local optimization algorithms tend to struggle finding a robust solution.

Given the aforementioned issues and limitations, the application of machine learning for the calibration of asset pricing models has recently gained interest. In particular, the application of artificial neural networks (ANNs) for accelerating the pricing of derivatives is a topic of interest. As one of the first, Hutchinson et al. (1994) analyzed the applications of ANNs to estimate the pricing function for derivatives in a non-parametric, model-free way. This idea was resumed amongst others by Quek et al. (2008) and Culkin and Das (2017).1 Recently various papers emerged dealing with

1 Ruf and Wang (2019) provide a comprehensive review of literature on the application of neural networks for option pricing and hedging.
a model-based approximation of derivative pricing functions under advanced asset pricing models. For example, Ferguson and Green (2018) apply a forward feed network to estimate the valuation function for equity basket options. Hirsa et al. (2019) analyse the performance of ANN pricing methods for European, Barrier and American options under different mathematical regimes. Liu et al. (2019) use ANNs for the approximation of option values under the Black & Scholes and Heston model. With respect to interest rate models, Kienitz et al. (2020) analyze the application of ANNs for the approximation of swaption prices under the Hull-White and Trolle-Schwartz model.

Based on the application of ANNs for the pricing of derivatives, there are several papers on utilizing these trained ANNs for calibration. Hernandez (2017) firstly presented this idea by applying a feed forward ANN for the calibration of a single-factor Hull-White model based on real market data (Sterling ATM swaptions). Dimitroff et al. (2018) use convolutional neural networks for the calibration of stochastic volatility models. As the application of ANNs is expected to accelerate the pricing process, the application of more complex models is an intensively discussed issue. In particular, the calibration of rough volatility models is extensively analyzed by Bayer and Stemper (2018), Bayer et al. (2019), Horvath et al. (2019) and Stone (2020). The general idea behind these papers is the acceleration of the instrument valuation via the application of a neural network. The optimization itself is in most cases still based on a local optimization algorithm. Furthermore, most of the existing papers do not use real market data to assess the performance of the ANN and the associated calibration of model parameters.

We build our approach upon the calibration framework proposed by Liu et al. (2019). It involves a two-step procedure for the calibration of financial models. First, a feed forward ANN is trained based on simulated training data to approximate the valuation function under a given asset pricing model. Second, the trained ANN is utilized in a backward manner for the calibration of model parameters. We apply the calibration framework to an interest rate (IR) term structure model based on Trolle and Schwartz (2009). While Liu et al. (2019) use simulated data for the training of the ANN as well as the calibration of the model parameters, we empirically analyze the performance of this framework based on a comprehensive set of real market data provided a consecutive series of trading dates (18 months). Hernandez (2017) uses historic market data for the calibration of the Hull-White model, but the data is limited to ATM swaptions.
Furthermore, the adjustments to the Hull-White model, such as keeping the parameters constant across swaption maturities are considered as being too simplistic for practical application (Kienitz et al. (2020)). Hence, we consider our study as the first comprehensive empirical assessment that deeply examines the application of ANNs for calibration of financial models based on real market data. The purpose of the paper is to show, how to practically enhance and accelerate managerial decisions based on model calibration. To the best of our knowledge, we are the first to validate the ANN calibration results against a benchmark implementation based on real market data, including the current COVID-19 pandemic.

We extend the literature regarding the calibration of IR term structure models in three important ways. We are the first to establish an ANN for the valuation of swaptions under the Trolle-Schwartz model and validate the results based on historical market data, evaluating their performance in real-life situations. Second, we calibrate the Trolle-Schwartz model parameters for a consecutive series of trading days based on real market data for EUR swaptions using a global optimization algorithm. The calibration results are validated and benchmarked against a local optimization algorithm. We find that the resulting model parameters using a global optimizer are much more stable compared to our benchmark. This has important managerial implications as stable parameters lead to less volatile P&L figures, highly enquired by financial institutions. Furthermore, several more simplistic but widely used IR term structure models can be recovered from the Trolle-Schwartz model by using assumptions for certain parameters (Trolle and Schwartz (2009)). Therefore, we consider our results interesting not only for institutions using the TS model, but for a wide-range of market participants applying less complex IR term structure models. Finally, we derive lessons learned and guidelines for the practical application of ANNs for financial model calibration and decision making.

The rest of the paper is structured as follows. In section 2, we briefly introduce the Trolle-Schwartz model and show the procedure for calibrating the model. Section 3 provides a detailed explanation of our ANN calibration approach and its subsequent components. The data, methodology and results of our comprehensive empirical study are presented in section 4. This includes the validation and benchmarking of our results. Section 5 concludes this paper.
2 Calibration of interest rate term structure models

2.1 Model calibration

The calibration of financial models is a reverse optimization problem. We assume that we can use a given model to calculate prices of certain financial instruments. The calculation of the price estimate ($\hat{p}_{j}^{(model)}$) under a specific model for a given instrument ($j$) requires a series of inputs. This includes the properties of the instrument ($\tau_{j}$), the parameters of the model ($\Omega_{t} = (\omega_{t1}, \ldots, \omega_{tn})$) and a set of market data ($\Lambda_{t}$). By applying a calibration procedure, the model parameters are set such that the difference between the resulting model prices and the observable market prices is minimized given a specific loss function ($L$):

$$\arg\min_{\Omega_{t}} \sum_{j \in \mathcal{F}_{t}} L\left(p_{j}^{(market)}, \hat{p}_{j}^{(model)}(\Omega_{t} \mid \tau_{j}, \Lambda_{t})\right), \quad (1)$$

where $\mathcal{F}_{t}$ represents a set of financial instruments, which have observable market prices ($p_{j}^{(market)}$) at a specific point in time ($t$). The calibration requires a reasonable and thoughtful choice of calibration instruments. Instruments used for calibration should be liquid, frequently traded and inherit all relevant risk drivers of the instruments it will be applied to. Furthermore, the quality of the calibration is limited by the ability of the model to capture all relevant risk drivers and dependencies of the observable market prices. Nevertheless, the calibration of a complex and high-dimensional model might be quite burdensome from a methodological and computational point of view. Hence, the choice of an appropriate model requires balancing accuracy and computational performance. Especially, if these models are used for pricing financial instruments the ability to perform the calibration in a reasonable amount of time is a crucial prerequisite for their practical application, e.g., for investment or hedging decisions. In addition, the traceability and interpretability of the model is an important feature and considered a key aspect in supervisory oversight and validation.

Electronic copy available at: https://ssrn.com/abstract=3667070
2.2 The Trolle-Schwartz model

In this paper, we perform an empirical study for the application of an ANN based framework to calibrate an interest rate (IR) term structure model. We use a term structure model based on the work of Trolle and Schwartz (2009), the so called Trolle-Schwartz (TS) model. The TS model is an advanced stochastic volatility model based on the Heath-Jarrow-Morton framework (Heath et al. (1992)). We use the TS model in its risk-neutral setting. The TS model consists of two stochastic processes for the instantaneous forward rate and the variance of the rate process. The dynamics of the forward rate are modelled as follows (see Trolle and Schwartz (2009)):

\[
d f(t, T) = \mu_f(t, T)dt + \sum_{i=1}^{N} \sigma_{f,i}(t, T)\sqrt{v_i(t)}dW_i^Q(t)
\] (2)

\[
d v_i(t) = \kappa_i (\theta_i - v_i(t))dt + \sigma_i \sqrt{v_i(t)} \left( \rho_i dW_i^Q(t) + \sqrt{1-\rho_i^2} dZ_i^Q(t) \right)
\] (3)

Given these differential equations, the evolution of the forward rate is defined based on 2N standard Wiener processes \((W_i^Q(t), Z_i^Q(t))\). \(N\) defines the number of dimensions of the model. In equation (2), \(\mu_f(t, T)\) equals the forward drift. Under the assumption of no-arbitrage, Heath et al. (1992) have shown that this term is defined as:

\[
\mu_f(t, T) = \sum_{i=1}^{N} v_i(t) \sigma_{f,i}(t, T) \int_t^T \sigma_{f,i}(t, u)du
\] (4)

Based on this property, the evolution of the forward rate under the risk-neutral measure is solely driven by the initial forward rate curve, the volatility state variables \((v_i(t))\) and the volatility function \((\sigma_{f,i})\). Within the TS model, the volatility function is set to a specific form (see equation(5)) to ensure that the forward rate can be represented by a finite-dimensional Markov process and a time-homogeneous volatility structure.

\[
\sigma_{f,i}(t, T) = (\alpha_{0,i} + \alpha_{1,i}(T-t)) \cdot e^{-\gamma_i(T-t)}
\] (5)

\[2\] Within this paper we provide an overview of the Trolle-Schwartz model based on Trolle and Schwartz (2009). Hence, we do not provide mathematical derivation, proofs and background of the model. For additional information on the model and its methodological foundations, please refer to Trolle and Schwartz (2009) and Kienitz et al. (2020).
The TS model offers semi-analytical pricing for the most common of interest rate products. Within this paper, we use swaptions prices as input for the calibration of the TS model. Hence, we need to calculate the prices of swaptions under the TS model. The TS model provides a semi-analytical solution for an option on a zero-coupon bond. We perform the pricing of swaptions by utilizing these pricing functions and mapping the swaptions based on the stochastic duration method (Munk (1999)).

Our empirical study shall provide evidence and guidance for the practical application of ANNs for calibration issues. Hence, we decide to use a complex and multi-dimensional pricing model to assess the performance of our calibration framework. Furthermore, the TS model offers a semi-analytical solutions for pricing European Swaptions, which will be used as calibration instruments within our empirical study. Hence, we are able to generate trade and test data in a fast and efficient way. Nevertheless, the model is complex enough to capture the properties of the market-implied volatility / price cube. The TS model can be transformed into more simplistic IR term structure models by simply using specific settings for the parameters of the volatility function (see Trolle and Schwartz (2009)). Hence, our results are also relevant for the application of ANNs to calibrate more simplistic IR term structure models.

Table 1: Parameters of the Trolle-Schwartz model

| Parameter | Interpretation |
|-----------|----------------|
| κi        | Mean reversion speed of the variance process |
| θi        | Long-term variance |
| σi        | Volatility of the variance |
| ρi        | Correlation between forward rate and volatility state variables |
| α0,i      | Free parameter of the volatility function σf,i(t,T) |
| α1,i      | Free parameter of the volatility function σf,i(t,T) |
| γi        | Free parameter of the volatility function σf,i(t,T) |

Notes: This table provides an overview of the model parameters in the TS model and their interpretation.

As discussed above, the calibration of a model requires the setting of model parameters such that the model prices fit the observable market prices. The calibration of the TS model requires the determination of Nx7 parameters (see table 1). We consider these parameters as elements of N parameter vectors Ωi. Within this paper, we set N = 1 which reduces the calibration problem to the determination of seven parameters.

3 For additional details and background on the pricing of swaptions under the TS model, please refer to Trolle and Schwartz (2009) and Kienitz et al. (2020).
Within our empirical study, we perform a daily calibration of these parameters by using the sum of squared errors over a set of observable swaption prices as loss function. Hence, the specific calibration procedure for the TS model can be written as:

$$\arg\min_{\Omega_t} \sum_{j \in \mathcal{F}_t} \left( p_j^{(market)} - \hat{p}_j^{(model)} (\Omega_t | \tau_j, \Lambda_t) \right)^2, \quad (6)$$

where $\Omega_t$ equals the parameter vector ($\Omega_t = (\kappa_t, \theta_t, \sigma_t, \rho_t, \alpha_{t0}, \alpha_{t1}, \gamma_t)$). In case of IR swaptions, $\tau_j$ equals a vector of properties describing the instrument, such as expiry date of the swaption, tenor and swap rate of the underlying swap. $\Lambda_t$ represents the yield curve (and discount factors) in the respective currency. Based on these inputs a model price is calculated. The calibration procedure optimizes $\Omega_t$ such that the loss function is minimized. The observable market prices are structured along three dimensions (expiry tenor, swap tenor, strike). Hence, the observable swaption data can be thought of as a cube of swaption prices.

3 ANN calibration approach

3.1 Methodological overview

In general, a calibration framework should be flexible, robust, fast and accurate. All these properties are combined in artificial neural networks. They became widespread in the financial domain due to their flexibility and approximation properties. Our calibration framework (CaNN) is based on the idea presented by Liu et al. (2019) and involves two consecutive components. First, we train an ANN to learn the pricing functions for swaptions under the TS model (forward pass). Second, the resulting ANN is applied within a calibration procedure, to fit the model parameters ($\Omega$) to a set of observable market prices. These two steps are explained in more detail in the subsequent sections. There are other approaches that suggest to apply ANNs to directly estimate model parameters from market prices without using a pricing function based on a traditional model. While this approach could theoretically provide a better fit to market data, it imposes several issues. As stated by Horvath et al. (2019) these approaches might not be compatible with the prevailing regulatory requirements.
due to their lack of interpretability and straightforward strategies to determine the network architecture. Furthermore, evidence for the stability and robustness of these approaches does still need to be provided. Hence, we strongly advocate a separation of pricing and calibration in an ANN based calibration framework.

ANNs are capable of approximating any continuous function that maps input variables to outputs, see Cybenko (1989) and Hornik (1991). Our approach utilizes this principle to map input features on swaption prices in a highly non-linear and complex fashion. For each swaption, the neural network starts with covariates \((\Omega, \tau_j, \Lambda) \in \mathbb{R}^p\) as inputs which are called input neurons. The network consists of stacked layers \(l = 1, \ldots, L\) whereby each layer consists of \(k_l = 1, \ldots, K_l\) neurons \(h_{l|K_l} \in \mathbb{R}^{K_l}\) that are determined by an affine combination of neurons in the previous layer which is composed with an arbitrary (non-linear) activation function \(\sigma\). Formally, the ANN is defined by:\(^4\)

\[
h_{l|K_l} = \sigma \left( W_l h_{(l-1)|K_{l-1}} + b_l \right)
\]

with \(W_l \in \mathbb{R}^{K_l \times K_{l-1}}, b \in \mathbb{R}^{K_l}\) as parameters which are usually called weights and biases. Estimates are derived from the last layer, the so-called output layer and are given by choosing the identity function for \(\sigma\), resulting in:

\[
F(y|x) = W_{L+1} h_{K_L} + b_{L+1}
\]

The weights and biases are estimated via backpropagation.

### 3.2 The forward pass: Learning the pricing function

The first part of our calibration framework consists of learning the mapping function, in our application the Trolle-Schwarz Model, via an Artificial Neural Network (ANN). Finding a suitable architecture which holds the balance between computational time, complexity and approximation accuracy is the main task in this subsection. As our goal is a highly accurate approximation, we use a rather large and complex neural

---

\(^4\) Within this paper we provide a short overview on the mathematical foundations of ANNs only. For a comprehensive summary of the most common mathematical concepts of deep learning, please refer to Kraus et al. (2020).
network, as it ensures a high approximation accuracy. As ANNs are sensitive to diverging dimensions of input parameters, we normalize all features $\xi \in (\Omega, \tau_j, \Lambda)$ to a predefined range, i.e $\xi \in [\xi_{\text{min}}, \xi_{\text{max}}]$. This makes it also easier in the backward pass to set optimization bounds. The features are normalized by:

$$\frac{2\xi - (\xi_{\text{max}} + \xi_{\text{min}})}{\xi_{\text{max}} - \xi_{\text{min}}} \in [-3, 3].$$

(7)

Usually, ANNs are prone to the problem of overfitting. Meaning, that the network is able to approximate the training data very well, but fails to approximate unseen test data. This is usually the case in out-of-time prediction in the financial context. Our approach is not designed to provide a prediction in an out-of-time fashion, as we want to approximate a specific mapping function as accurate as possible. In our case, the mapping function of training and test data is equal, as both datasets are generated via the pricing function for swaptions under the TS model. Hence, approximating the training data very well must ensure that the test data is approximated similarly good. Hence, in our opinion, the issue of overfitting can be neglected in the prevailing use case, as shown by our empirical results in Section 4.2. Furthermore, this opinion is supported by findings of previous papers, such as Liu et al. (2019) and Liu et al. (2019). These authors conduct hyper parameter searches, including techniques to reduce overfitting. In none of the final models, an overfitting reducing technique is found to be beneficial for the quality of the ANN’s approximation. Hence, these findings underline our theoretical conclusion that the problem of overfitting can be neglected when learning the mapping function within a ANN based calibration framework. Of course, this only holds if we generate a vast amount of training data, which can easily be ensured.

The ANN is trained to minimize the following loss function with respect to weights $W$ and biases $b$:

$$\arg\min_{W, b} \sum \left(p_j^{(\text{model})}(\Omega, \tau_j, \Lambda) - \hat{p}_j^{(\text{ANN})}(W, b | \Omega, \tau_j, \Lambda)\right)^2$$

(8)

As a precaution, we also generated training samples to calculate the loss of equation (8) in an out of sample task. In general, the ANN is trained over 5,000 epochs to ensure
the weights and biases are estimated as accurate as possible.

3.3 The backward pass: Calibration of model parameters

The final step of our calibration framework is to calibrate the input parameters \( \Omega_t \) given the observed market prices at a specific trading day \((t)\). After the forward pass is successfully accomplished, the weights and biases describing the relation of the input parameters \((\Omega_t, \tau_j, \Lambda_t)\) to the prices of a swaption \(p_j\) are known. This means that the mapping function is now deterministic in the sense that simple and fast matrix multiplications map the input to the corresponding swaption prices \((\hat{p}_j^{(ANN)})\). Hence, we have now a very fast way to price a swaption given \((\Omega_t, \tau_j, \Lambda_t)\). For calibration purposes, we are interested in \(\Omega_t\) which express the observed market prices \(p_j^{(market)}\) based on the TS model as good as possible. Hence, we basically invert the trained neural network by setting the values of \(\Omega_t\) as degrees of freedom in a minimization problem:

\[
\arg\min_{\Omega_t} \sum_{j \in F_t} \left(p_j^{(market)} - \hat{p}_j^{(ANN)}(\Omega_t | \tau_j, \Lambda_t, W, b)\right)^2
\]  

(9)

The optimization problem in equation (9) is essentially the calibration problem widely faced in the financial industry. To solve this minimization problem, usually local optimizers are widely used due to their speed (see Liu et al. (2019)). In our framework, several local minima exists, see i.e. Gilli and Schumann (2012). This may be a bottleneck for local optimizers. As we gain a high amount of speed by using the neural network approach, we are able to use slower, but in terms of minimization more robust optimizers. In our calibration framework, we apply a global optimizer called differential evolution (see Storn and Price (1997) for more details)\(^5\). This stochastic optimization scheme is able to find a global minimum even if the optimization problem is non-convex. We speed up our calibration framework, by using the (transformed) values of \(\Omega_{t-1}\) as initial values for the optimization.

\(^5\) Please note that we use the default values in the implementation of the Python package SciPy, except for the population size which we set to 49.
4 Empirical study

4.1 Data

Our empirical study is based on a comprehensive set of daily prices for EUR swaptions. These prices are used as input for the calibration procedure. The available market data covers 373 consecutive trading days from January 2019 to June 2020. Hence, our dataset includes the stressed market period in the context of the COVID-19 pandemic in spring 2020. The daily swaption data is available for different expiry tenor, swap tenor and strike values:

- Option Tenor: 1M, 3M, 6M, 9M, 1Y, 2Y, 5Y, 10Y, 15Y, 20Y
- Swap Tenor: 1Y, 2Y, 5Y, 10Y, 15Y, 20Y, 30Y
- Strike (ATM ± bp): 0, 12.5, 25, 50, 100, 150, 200

On each trading day, we observe valid prices for about 800 swaptions. This amounts to a total number of more than 300,000 price observations. In practical applications, financial institutions tend to use a reduced set of swaptions for the calibration of IR term structure models to reduce the calibration time. Within our empirical study, we do not further reduce the amount of swaptions entering the calibration procedure. In addition to swaption data, we obtain the yield curve (6m EURIBOR) for each trading day as well as the relevant forward rate for each swaption. The yield curve is transformed into discount factors for 53 tenors. We compare our calibration performance against a benchmark implementation, which is using a Levenberg-Marquardt optimization algorithm (see Levenberg (1944), Marquardt (1963)) by iterating the traditional pricing formula. The data for each trading day includes the model parameters and model prices estimated by the benchmark implementation. Table 2 provides an overview of the observed values for each TS parameter and the associated model prices.

As discussed in section 3.1, we do not perform the training of the neural network based on real swaption data. While our swaption dataset includes 300,000 observation, it only provides 373 combinations of TS model parameters. Hence, the number of
observations is not sufficient to ensure a satisfying performance of the ANN. To train the ANN, we need to generate a large amount of artificial (synthetic) swaption data. We generate the required dataset by sampling swaption data for 12,000 synthetic trading days. By using synthetic swaption data for training and testing, we are able to set aside the swaption prices obtained from real market data for the validation of the ANN. The properties of the synthetic swaptions are set to the discrete values shown above. The values for the TS model parameters are randomly sampled from predefined ranges (see Table 2) applying a uniform distribution. Please note that the value ranges used for sampling of parameter values exceed the observed parameter values of the benchmark implementation. Thereby, we ensure that the calibration procedure is able to provide prices for parameter values outside of observed ranges. Furthermore, the CaNN framework is able to find optimal parameter values outside the observed ranges in the calibration procedure.

| Parameter     | observed (Benchmark) | Sampling (CaNN) |
|---------------|----------------------|-----------------|
| Kappa (κ)     | [0.0031,2.80]        | [0.005,3]       |
| Theta (θ)     | [0.037,3.89]         | [0.01,4.0]      |
| Sigma (σ)     | [0.24,1.73]          | [0.1,2.0]       |
| Rho (ρ)       | [-0.047,0.60]        | [-0.50,0.80]    |
| Alpha0        | [0.00001,0.006]      | [0.00001,0.008] |
| Alpha1        | [0.0007,0.005]       | [0.0005,0.005]  |
| Gamma (γ)     | [0.048,0.089]        | [0.01,0.1]      |
| Prices \(\hat{p}_i^{(model)}\) | [0.0,0.64] | [0.0,1.06] |

Notes: This table provides observed values for Trolle-Schwartz parameters as well as the value ranges used for sampling of training data.

The yield curve for each synthetic trading day is randomly sampled from a collection of yield curve data. The yield curve dataset is constructed by a blended approach, where we combine historically observed market data with synthetic yield curve data. First, we collect yield curves for eight different currencies\(^6\) for a historic two-year time period (Apr 2018 - Apr 2020). This includes about 3,700 different yield curves. We do not include the yield curves observed in May and June 2020 to obtain a real out-of-time validation of the CaNN calibration results within our empirical analysis. Second, we enrich the dataset by adding 20,000 synthetic yield curves. These yield curves are

\(^6\) We use the historically observed yield curves for the following currencies: EUR, USD, GBP, JPY, CHF, DKK, NOK, SEK

Electronic copy available at: https://ssrn.com/abstract=3667070
generated by using an algorithm based on the Nelson-Siegel-Svensson methodology (see Nelson and Siegel (1987), Svensson (1994)). Our blended approach provides a comprehensive and representative yield curve dataset. On the one hand, we consider recent historic market environment in the training process. On the other hand, we ensure that the resulting ANN is flexible enough to cope with new unseen market data. Furthermore, this approach offers the possibility for recurring generation of training data and re-training of the CaNN framework based on newly observed yield curves.

By following the generation procedure outline above, we obtain a total number of 9.6 million synthetic swaptions. The prices of these swaptions are calculated by applying the pricing procedure outlined in section 2.2. The resulting dataset is used for training and testing the ANN within the forward pass. In general, we consider the generation of training and test data as a crucial and probably the most laborious task within the calibration framework. The composition of the dataset and its granularity are important drivers of the CaNN’s estimation power. Please note that the initial training of the ANN is time consuming and requires significant computational capacities. Nevertheless, this step has to be performed only once. The application of the CaNN framework can be accompanied by frequent re-training, which is significantly less time consuming.

4.2 ANN architecture & forward pass (pricing)

Finding a suitable ANN architecture is a major cornerstone of the successful approximation of the pricing function. As usual, one has to find the balance between approximation accuracy and computational burden. The training of the ANN with a large number of swaption has to be done only once, hence we chose a rather complex, but more accurate architecture. In total, four hidden layers with 2048, 1024, 512 and 256 neurons are used. To optimally train the ANN, we use the Adam optimizer, Relu activation function and a tailored learn rate schedule. As described above, we do not use any dropout layer or early stopping criterion. To ensure convergence with the TS pricing results, we train the ANN with 5000 epochs. An overview of the applied hyper parameters is illustrated in Table 3. For robustness, we also employed and validated the hyper parameter setting proposed by Liu et al. (2019) with 200 neurons in each of the four hidden layers. The accuracy in terms of mean squared error is 10 times
worse than with our architecture. This implies that any calibration framework and model needs a tailored set of hyper parameters to provide the a sufficiently accurate estimation of model prices.

**Table 3: Hyper parameter of the CaNN**

| Parameter                | Value                          |
|--------------------------|-------------------------------|
| Number Features (X)      | 66                            |
| Hidden Layers            | 4                             |
| Neurons per Layer        | [66, 2048, 1024, 512, 256, 1] |
| Number of parameters     | 2,891,777                     |
| Loss function            | Sum of squared errors         |
| Activation function      | ReLu                          |
| Optimizer                | Adam                          |
| Initialization           | Glorot-Uniform                |
| Batch Size               | 16,384                        |

Notes: This table provides the applied hyper parameters of the final CaNN. In total, a neural network with four hidden layers and 2,891,777 parameters is trained to approximate swaption prices under the TS model.

To train the ANN, we randomly split the 9,6 million generated synthetic swaptions into a training set (7,68 million) and a test set (1,92 million). Table 4 shows key evaluation metrics in the train and test sample. We observe only small differences, when comparing the results for the train and test set. This may imply that the ANN generalizes well and we do not encounter overfitting. Furthermore, the metrics are well in line with results of previous studies, see e.g. Liu et al. (2019) or Horvath et al. (2019). The very similar performance for the train and test data may also be attributed to the comparatively large training sample, which is imminent to approximate the mapping function.

**Table 4: Results of ANN training**

|        | CaNN  | MSE    | MAE    | RMSE   |
|--------|-------|--------|--------|--------|
| Training | 1.47e-07 | 2.38e-04 | 3.52e-04 |
| Testing  | 1.80e-07 | 2.45e-04 | 4.24e-04 |

Notes: This table provides the used hyper parameter of the final CaNN. In total, a neural network with four hidden layers and 2,891,777 parameters is trained to approximate the TS model.

In contrast to most other papers on the application of ANNs for pricing and calibration, we perform an additional validation of the forward pass based on real pricing data obtained from a benchmark implementation (BM). We call this step the “out-of-simulation validation”, as the data used to assess the ANN’s pricing performance has not been generated with the same process as the train and test sample. Thereby, we
ensure that the ANN has learned the TS pricing function correctly and performs well in a true out-of-sample evaluation. From our point of view, the validation based on results from a benchmark model is a prerequisite for the practical application of an ANN based calibration framework. To perform the out-of-simulation validation, we pass the observed parameters estimated by the benchmark implementation \((\Omega^{(BM)}_{\text{BM}})\) together with the historic market data for the respective trading day through the ANN for all real market swaptions across available trading days. Afterwards, we compare the predicted prices of the trained ANN with the model prices generated by the benchmark implementation to obtain different error measures (see equation (10) for mathematical illustration).

\[
MSE = \frac{1}{T} \sum_{t=1}^{T} \sum_{j \in \mathcal{F}_t} \left( \hat{p}_{j}^{(\text{model})} \left( \Omega^{(BM)}_{\text{BM}} | \tau_j, \Lambda_t \right) - \hat{p}_{j}^{(\text{ANN})} \left( \Omega^{(BM)}_{\text{BM}} | \tau_j, \Lambda_t, W, b \right) \right)^2
\]  

(10)

The results of this validation step are displayed in Table 5. First, we check the performance for the time period from January 2019 to April 2020. The swaption data from this period was used for setting the parameter ranges and yield curves for the simulation of synthetic swaptions. As the evaluation metrics are close to the results obtained in the training and testing, we may conclude that the ANN is robust in real-life market situations. As a next step, we use the benchmark parameters from the out-of-time period (May 2020 - June 2020). Data and information from this period, such as parameter values and yield curves, has not been used in the previous steps and is therefore completely new to the framework. The results for this period of time indicate that we achieved generalization even in an out-of-time perspective with unseen features. Given these results, we are to the best of our knowledge the first to provide comprehensive evidence that the pricing of derivatives via a neural network is not only working in simulated environments, but also in real-life applications.

| Table 5: Results of ANN training |
|-----------------------------|----------|----------|----------|
| CaNN                        | MSE      | MAE      | RMSE     |
| Out-of-simulation (Jan 2019-Apr 2020) | 5.47e-07 | 2.98e-04 | 7.24e-04 |
| Out-of-simulation (May 2020 & June 2020) | 2.56e-07 | 2.73e-04 | 5.06e-04 |

Notes: This table show key evaluation metrics in the out-of-simulation validation. We divide the samples into data building the basis of our training (January 2019 to April 2020) and true out-of-time data (May & June 2020).
Figure 1: Real fit plots of selected trading days

Note: These figures show the real fit plots of selected real trading days. Furthermore, the day specific MSE is displayed. The price estimations of the ANN are displayed on the x-axis, whereas the model prices of the benchmark implementation is shown on the y-axis.

Figure 1 provides real fit plots for selected trading days taken from the out-of-time period. The plots compare the prices estimated by the ANN (x-axis) with model prices from the benchmark implementation (y-axis). As we can see, the points are on the bisecting line which implies a very good convergence of the ANN prices to BM model prices. In each real fit plot, the MSE for the respective trading day is displayed. For some days, we obtain much better results than in training, whereas for some days we are slightly worse. In summary, we find sufficient evidence that the trained ANN generalizes very well even if confronted with unseen data. Hence, the ANN provides a very good approximation of the TS pricing function for swaptions.

4.3 The backward pass (calibration)

In the next step of our calibration framework, we utilize the trained ANN to calibrate the TS model parameters to a daily set of observable swaption prices. Furthermore, we validate our results against a benchmark implementation. To the best of our knowledge, this is the first paper that features a comprehensive empirical study based on a full set of daily market data as well as a benchmarking against an actual practical implementation. For each of the 373 trading days, we obtain two calibrated parameter sets. One parameter set is returned from the benchmark implementation ($\Omega^{(BM)}_t$), while
the other parameter set results from our ANN based calibration framework \((Ω^{(ANN)}_t)\).

For clarification, we restate and concretize the general formulation of the calibration problem in equation (9) and provide a specific notation for both calibration processes:

\[
\begin{align*}
\arg\min_{Ω^{(BM)}_t} & \sum_{j ∈ F_t} \left( p^{(\text{market})}_j - \hat{p}^{(\text{model})}_j(Ω^{(BM)}_t | τ_j, Λ_t) \right)^2 \quad (11a) \\
\arg\min_{Ω^{(ANN)}_t} & \sum_{j ∈ F_t} \left( p^{(\text{market})}_j - \hat{p}^{(\text{ANN})}_j(Ω^{(ANN)}_t | τ_j, Λ_t, W, b) \right)^2 \quad (11b)
\end{align*}
\]

Both calibration approaches aim to minimize the sum of squared errors for each trading day. By minimizing the loss function, an optimal set of TS model parameters is selected. The benchmark approach performs the calibration by applying a local optimization algorithm (Levenberg-Marquardt) and repeatedly calls the traditional implementation of the semi-analytic pricing formula (see equation (11a)). The benchmark implementation involves parameter restrictions for the TS parameters to ensure that the optimizer returns a result. For this empirical analysis, the resulting benchmark model parameters \((Ω^{(BM)}_t)\) are obtained from the historical calibration results of the benchmark implementation. The CaNN framework utilizes the forward pass by frequently estimating swaption prices based on the trained neural network for different parameter settings (see equation (11b)). Please note that the weights and biases of the ANN have already been set in the training phase (forward pass) and are not altered during the calibration procedure.

Within the CaNN framework, a global optimization algorithm (Differential Evolution) is used to minimize the loss function given by equation (11b). The application of the differential evolution (DE) algorithm shall avoid the problem of stopping at local minima and offers the advantage that no starting values are required (see Liu et al. (2019))^7. However, we use the parameter values of the previous trading day as starting values for the DE algorithm. We have observed that using starting values leads to a faster convergence and significantly accelerates the calibration process. In practical applications, such as the referred benchmark implementation, the parameter values of the previous trading day are commonly used as starting point for the optimization process. This could potentially lead to a deterioration of the minimization, when

^7 Please note, that we set the bounds for the optimization to \([-3,3]\) and use a population size of 49.
applying local optimizers, but should not be an issue for global optimization algorithms. Hence, we are confident that there is no downside in setting starting values for the DE algorithm in the CaNN framework. On the contrary, we observed that setting starting values speeds up the ANN calibration by roughly 50 times. Thereby, the calibration for each trading day can be performed in about 30 seconds, making the global optimizer competitive to local optimization algorithms.

Table 6: Calibration results

| Period         | daily MSE (BM) | daily MSE (ANN) | daily SSE (BM) | daily SSE (ANN) |
|----------------|----------------|-----------------|----------------|-----------------|
| 01/2019-04/2020| 1.35e-06       | 1.41e-06        | 1.09e-03       | 1.14e-03        |
| 05/2020-06/2020| 1.71e-06       | 1.79e-06        | 1.38e-03       | 1.44e-03        |

Notes: This table show key evaluation metrics of the ANN and benchmark calibration result. We divide the samples into data building the basis of our training (January 2019 to April 2020) and true out-of-time data (May & June 2020).

Table 6 provides an overview of the calibration results equal to the average daily values of the loss function calculated by equations (11a) and (11b) as well as the daily mean squared error (MSE) for both calibration approaches. The results show that the CaNN framework provides calibration results that are very close to the benchmark implementation for both time periods. Nevertheless, there might be a concern that only these results do not provide sufficient evidence for the practical applicability of the CaNN framework. We expect that supervisory authorities will have a critical view on the application of ANNs for pricing and calibration as the ANN pricing function constructed in the forward pass is not considered traceable given the high amount of parameters in the neural network.

To prove that the CaNN provides accurate calibration results, an additional validation of the resulting parameter values is required, where the CaNN parameter set ($\Omega_t^{(ANN)}$) is used as input for the pricing formula for swaptions under the TS model. By comparing the resulting prices with observable market prices, we are able to prove that the CaNN calibration results hold true in the Trolle-Schwartz model framework. Hence, we apply equation (12b) to validate the ANN solution for each trading day. The result will provide insights with respect to the true quality of the CaNN calibration results.

$$SSE^{(BM)}(t) = \sum_{j \in F_t} \left( p_j^{(market)} - \hat{p}_j^{(model)}(\Omega_t^{(BM)} | \tau_j, \Lambda_t) \right)^2$$  \hspace{1cm} (12a)

Electronic copy available at: https://ssrn.com/abstract=3667070
\[
SSE^{(ANN)}(t) = \sum_{j \in F_{t}} \left( p_{j}^{(market)} - \hat{p}_{j}^{(model)}(\Omega_{t}^{(ANN)} | \tau_{j}, \Lambda_{t}) \right)^{2}
\]  \hspace{1cm} (12b)

Figure 2 illustrates the daily performance measure (SSE) for both calibration approaches over time. The black line represents the benchmark result (equation (12a)), while the grey line represents the performance measure for the CaNN framework (equation (12b)). In general, we find that the performance of both calibration approaches significantly varies over time. In the early months of 2019 the losses are comparatively low whereas in the fourth quarter of 2019, we observe a considerable increase. A remarkable spike can be observed after the break-out of the COVID-19 pandemic, meaning that the calibrated TS model prices strongly deviates from market prices. These results clearly indicate that a thorough assessment of ANN calibration approaches should be done in different market environments to ensure their practical applicability.

**Figure 2: Sum of squared errors over trading days**

Note: These figures show the sum of squared errors of trading days until end of April 2020. The grey line corresponds to the SSE using our CaNN approach, whereas the black line coincides with the SSE of the benchmark implementation.

The results presented in Figure 2 show that the CaNN framework produces competitive results compared to the benchmark implementation in terms of daily performance. For some market periods we can even find better solution for the parameters, see e.g. the period from June 2019 to August 2019 or the early months of 2019. The largest deviation between the CaNN and the benchmark implementation can be observed during the COVID-19 period in the March 2020. Nevertheless, the daily performance of both approaches does not differ significantly even in this stressed market environment.
Hence, the CaNN framework does provide comparable calibration results even in extreme and unusual market situations. Furthermore, the very good results for the out-of-time period (May/June 2020) indicate that the performance of the CaNN framework does not depend on including current market data during training.

In addition to analyzing the performance of the CaNN framework, we are interested in a comparison of the parameter estimates for both calibration approaches. Figure 3 illustrates the different estimates for all elements of $\Omega_t$ over time. The black line represents the parameter estimated by the benchmark implementation, while the grey line represents the respective element of $\Omega_t^{(ANN)}$.

Overall, the analysis reveals that parameter estimates from both calibration procedures are quite close to each other and have a similar evolution over time. However, the results indicate that the CaNN parameters are more stable over time and therefore more robust against taking extreme values. For example the BM estimates for $\Theta$ show four considerable peaks in the analyzed period, while the CaNN estimates show a relatively smooth evolution over time. On some days, the benchmark implementation obtains extreme values for certain parameters, which are equal to a boundary of the parameter restrictions. This may imply that the local optimizer used by the benchmark implementation ended up in a different local minimum on the respective trading days, leading to a compensation of the high $\Theta$ value by extreme settings for other parameters.

A similar issue can be observed for the parameter $\kappa$. In the period from September 2019 to mid January 2020, the estimated parameter of the benchmark implementation starts with values from 0.777 to 2.11 in early September, decrease to 0.56 mid September and then plumbs to 0.04 in mid January 2020. In contrast, the CaNN parameter fluctuates from September 2019 with values around 1.5 to end of January 2020 with values of 1.07 with considerably less fluctuations within this period. The same behaviour can be observed for $\sigma$ in the aforementioned time period. The evolution of CaNN estimates for different parameters show significantly lower fluctuation and that the parameters are less likely to take extreme values.

Based on these observations, we conclude that the CaNN framework generally provides more stable parameter estimates over time. From our point of view, the stability of
parameter estimates over time is a desirable property of a calibration procedure. The estimated model parameters are not only required as inputs for the pricing function, but also to specify stochastic processes in Monte-Carlo simulations for the purpose of calculating P&L components, such as Credit Valuation Adjustments (CVA), and risk measures. Hence, more stable parameters could significantly contribute to a reduction of day-to-day P&L volatility and costs of hedging in the trading business. Furthermore, more stable calibration results will lead to less volatile and more reliable risk measures, which enables managers to take more profound business decisions. This makes the
CaNN approach highly relevant for managers of financial institutions.

In summary, the results of our empirical study give rise to the conjecture that an ANN based calibration framework does not only provide competitive results compared to traditional approaches, but does also offer further benefits and advantages with respect to the stability and reliability of results. Hence, we conclude that there is indeed a practical applicability for ANN based calibration frameworks. However, we recognize that the practical application of a CaNN framework might involve significant challenges with respect to the fulfilment of regulatory requirements. In our opinion, the framework proposed in this paper is generally compliant with supervisory expectations as we offer a staggered approach involving additional and separate validation steps for the ANN based pricing as well as calibration procedure. Unfortunately, neural networks are often considered black boxes as it is somewhat difficult to explain and track the mapping function due to the high complexity and high amount of parameters. Hence, regulators may not be fully convinced and could obstruct a full replacement of traditional calibration frameworks with ANN based calibration procedures. Nevertheless, we strongly believe that the implementation of our proposed CaNN framework will provide significant added value for traditional calibration approaches, validation activities and business processes in several ways.

First, the calibration results from the CaNN framework provide valuable insights and an additional validation of the parameter estimates obtained from the currently applied calibration process. This information is useful for discussions with various internal and external stakeholders and might help to improve the calibration process and results. Second, the CaNN framework can be applied to assess the impact of (stress) scenarios on calibration parameters and subsequent measures. Third, we argue that our framework can be utilized to generate initial values for the currently implemented calibration procedures, which should lead to a faster and more robust calibration process. As the initial calibration is performed by calling the ANN, financial institutions are able to reduce dependencies between pricing and calibration procedures in daily production. In summary, financial institutions could be able to realize the benefits of ANN based calibration without replacing traditional approaches. Based on our results and observation this could increase the stability of results over time and reduce the probability of a local optimizer getting stuck in a local minimum.

Electronic copy available at: https://ssrn.com/abstract=3667070
Additionally, we find that the number of function evaluations required for the local optimizer can be reduced by more than one third using the start values obtained from the CaNN calibration instead of values of the previous day. We are able to provide empirical evidence for the latter aspect in the following case study, where we repeat the calibration process of section 4.3 in two different settings.

In the first setting we only use the local Levenberg-Marquardt (LM) optimization algorithm (see Levenberg (1944), Marquardt (1963)) to calibrate the parameters. In the second setting, we first use the differential evolution algorithm and afterwards pass these values to the LM optimization as initial values. We measure the performance over the out-of-time period based on the function evaluations required by the LM algorithm to arrive at the optimum on each day. Both optimizations are performed in the CaNN framework and on the same hardware to ensure comparability. On average the stand-alone LM algorithm (with previous day start values) requires 253 evaluations per trading day, while the combined optimization only requires 161 evaluations. Hence, we were able to decrease the number of function evaluations by about 36%, while keeping the level of accuracy. This is a considerable reduction leading to a faster calibration process and reduces the computational capacities required. The generation of the daily start values with the DE algorithm does not take longer than 30 seconds, which probably is considerably less than the potential speed up due to less function evaluations. These results support our conclusion that the implementation of a CaNN framework provides added value, even if traditional calibration procedures are not fully replaced.

5 Conclusion

This paper provides the first comprehensive proof of concept regarding the application of artificial neural networks (ANNs) for the calibration of asset pricing models. We propose a CaNN framework based on Liu et al. (2019) and integrate several enhancements to ensure practical applicability. First, we provide an enhanced concept for the generation of train and test data. Second, we introduce additional validation procedures based on real data to ensure that results of the CaNN are conform with observed pricing and
calibration results. Third, we perform a real out-of-time validation to provide evidence that the CaNN framework can cope with unseen data.

Based on a comprehensive time series of real market data, we are able to show that our calibration framework is able to produce competitive calibration results for a complex IR term structure model compared to a benchmark implementation. Our empirical analysis covers 1.5 years of swaption data, including the stressed market environment following the break-out of the COVID-19 pandemic. Hence, our calibration approach is suitable for real-life calibration problems and the CaNN framework performs well in different market environments. We find that the parameters estimated by the CaNN framework are more stable over time compared to the results of the benchmark implementation. More stable parameter estimates could help to reduce the P&L volatility from valuation adjustments and lead to a more reliable risk assessment for the derivatives trading business. Hence, a CaNN framework will provided added value, beyond a potential acceleration of the calibration process.

We are aware that our empirical analysis is limited to one IR term structure model for a single currency (EUR). The decision to use the Trolle-Schwartz model was based on the aspiration to analyze the performance of the calibration framework for a rather complex, but practically relevant model. Furthermore, the TS model can be easily reduced to more simplistic term structure models. However, we believe that the application of our framework to further currencies, models and asset classes will provide further findings regarding the performance of ANN based calibration frameworks. Furthermore, our future work will also focus on obtaining additional insights with respect to the calibration procedure from the CaNN framework, such as information on parameter sensitivity or importance of different inputs.

Further conclusions for the practical implementation of an ANN based calibration framework are as follows. First, the composition and quality of train and test data is a major driver of the CaNN’s performance. Real swaption data should not be used for training and testing as the data is more valuable for validation. Hence, we propose a blended approach, which produces synthetic data by combining information from historic market data with an algorithm that simulates synthetic datasets. Second, we recommend to set start values for the global optimizer based on the previous day’s...
results as this significantly accelerates the CaNN calibration process. Third, a tailored schedule for the learning rate is beneficial for the accuracy and performance of the training process.

Although we believe that our framework generally adheres to regulatory requirements, its practical application might be viewed critical by supervisory authorities as the training process and resulting ANN pricing function is not fully traceable. To counteract this, we offer a staggered approach involving additional and separate validation steps for the ANN based pricing as well as calibration procedure. However, regulators might still obstruct the replacement of traditional implementations with the CaNN framework. Nevertheless, the implementation of our framework and the subsequent integration of its results could significantly improve traditional calibration procedures in terms of accuracy, robustness and speed and provide additional insights for validation processes. Taking this aspect into consideration, we are certainly justified in saying that a CaNN based framework is of high practical relevance and has the potential to improve model calibration, risk assessment and business decisions.
References

Bayer, C., B. Horvath, A. Muguruza, B. Stemper, and M. Tomas (2019). On deep calibration of (rough) stochastic volatility models. *Working Paper*. Available at https://arxiv.org/pdf/1908.08806.pdf.

Bayer, C. and B. Stemper (2018). Deep calibration of rough stochastic volatility models. *Working Paper*. Available at https://arxiv.org/abs/1810.03399.

Culkin, R. and S. R. Das (2017). Machine Learning in Finance: The Case of Deep Learning for Option Pricing. *Journal of Investment Management* 15(4), 92–100.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems* 2, 301–314.

Dimitroff, G., D. Röder, and C. P. Fries (2018). Volatility Model Calibration With Convolutional Neural Networks. *Working Paper*.

Ferguson, R. and A. Green (2018, October). Deeply Learning Derivatives. *Working Paper*.

Gilli, M. and E. Schumann (2012). Calibrating Option Pricing Models with Heuristics. In A. Brabazon, M. O’Neill, and D. Maringer (Eds.), *Natural Computing in Computational Finance: Volume 4*, Studies in Computational Intelligence, pp. 9–37. Berlin, Heidelberg: Springer.

Heath, D., R. Jarrow, and A. Morton (1992). Bond Pricing and the Term Structure of Interest Rates: A New Methodology for Contingent Claims Valuation. *Econometrica* 60(1), 77–105.

Hernandez, A. (2017, June). Model calibration with neural networks. *Risk*. https://github.com/Andres-Hernandez/CalibrationNN.

Hirsa, A., T. Karatas, and A. Oskouii (2019, February). Supervised Deep Neural Networks (DNNs) for Pricing/Calibration of Vanilla/Exotic Options Under Various Different Processes. *Working Paper*. arXiv: 1902.05810.
Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks* 4(2), 251–257.

Horvath, B., A. Muguruza, and M. Tomas (2019). Deep Learning Volatility. *Working Paper*. Available at https://ssrn.com/abstract=3322085.

Hutchinson, J. M., A. W. Lo, and T. Poggio (1994). A Nonparametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks. *The Journal of Finance* 49(3), 851–889.

Kienitz, J., S. K. Acar, Q. Liang, and N. Nowaczyk (2020). Deep option pricing - term structure models. *Journal of Machine Learning in Finance* 1.

Kraus, M., S. Feuerriegel, and A. Oztekin (2020). Deep learning in business analytics and operations research: Models, applications and managerial implications. *European Journal of Operational Research* 281(3), 628–641. Available at https://doi.org/10.1016/j.ejor.2019.09.018.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics* 2(2), 164–168.

Liu, S., A. Borovykh, L. A. Grzelak, and C. W. Oosterlee (2019). A neural network-based framework for financial model calibration. *Journal of Mathematics in Industry* 9(1), 9.

Liu, S., C. W. Oosterlee, and S. M. Bohte (2019). Pricing Options and Computing Implied Volatilities using Neural Networks. *Risks* 7(1), 16.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics* 11(2), 431–441.

Munk, C. (1999). Stochastic duration and fast coupon bond option pricing in multifactor models. *Review of Derivatives Research* 3, 157–181.

Nelson, C. R. and A. F. Siegel (1987). Parsimonious modeling of yield curves. *Journal of Business* 60(4), 473–489.
Quek, C., M. Pasquier, and N. Kumar (2008, October). A novel recurrent neural network-based prediction system for option trading and hedging. *Applied Intelligence* 29(2), 138–151.

Ruf, J. and W. Wang (2019, November). Neural networks for option pricing and hedging: a literature review. *Working Paper*. Available at https://arxiv.org/abs/1911.05620.

Stone, H. (2020). Calibrating rough volatility models: a convolutional neural network approach. *Quantitative Finance* 20(3), 379–392.

Storn, R. and K. Price (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization* 11(4), 341–359.

Svensson, L. E. (1994). Estimating and interpreting forward interest rates. *IMF Working Paper*.

Trolle, A. B. and E. S. Schwartz (2009). A General Stochastic Volatility Model for the Pricing of Interest Rate Derivatives. *The Review of Financial Studies* 22(5), 2007–2057.