NEXIALOG CONSULTING

Fast Calibration of Implied Volatility Model

Ernesto LOPEZ FUNE, Jonathan VONGDARATH
Nexialog Consulting, Paris, France

22 décembre 2023

Abstract

Calibrating financial models becomes increasingly challenging as they becomes more and more complex. This step is, nevertheless, crucial for guiding professionals in minimizing the risks of inaccurate pricing or hedging results, which can lead to significant financial losses. In the framework of the SVI model, its complexity can be time consuming due to the non-convex nature of the problem, which is why in this article we investigated the performance of several optimization algorithms, including machine learning regression ones, in terms of accuracy and computational efficiency. Furthermore, our off-line computations enable us to adjust the parameters in real time, without the need for time-consuming recalibrations. This provides greater flexibility and adaptability to changing market conditions, which is crucial for financial institutions seeking to stay ahead of the curve.

Keywords — svi, smile volatility, machine learning, catboost, random forest

NEXIALOG CONSULTING

Summary

1 Introduction

Stochastic Volatility Inspired models (hereafter referred to as SVI models) have become increasingly popular in financial engineering due to their ability to efficiently model the volatility of financial assets [1, 2, 3]. These models are designed to capture the time-varying nature of volatility and are often used in options pricing and hedging. However, due to the often non-convexity of the loss function [3, 4], fitting these models to financial data can be challenging, as they have many parameters that need to be estimated directly from market data. To all this is added the algorithmic complexity of the optimizer used, as well as the time it takes to perform the calculations in case one needs to perform a dynamic optimization with market data in real time.

SVI models were developed to address some of the challenges associated with traditional Stochastic Volatility models (hereafter SV models) [5, 6, 20] ; they are computationally efficient and can be calibrated to market data quickly. However, the accuracy of the calibration depends on the choice of model and the calibration technique. The calibration process involves estimating the model parameters using an optimization algorithm to minimize the loss function, usually taken as the average distance between the model-implied and the observed market prices. Gradient descendent inspired optimizers are widely used for optimization purposes [7, 8, 9], however, in this aticle, we focused on metaheuristic ones such as the Nelder-Mead, Basin-Hopping, and Differential Evolution (hereafter NM, BH and DE respectively), which are particularly useful for complex, non-convex and high-dimensional problems, where gradient descent optimizers are not effective [10, 11, 12]. One common feature of these algorithms is that they do not require the gradient of the loss function to be known. Instead, they explore the search space iteratively using various strategies to find an optimal or near-optimal solution through metaheuristic techniques such as local search, randomization, and exploration-exploitation trade-offs to navigate the search space efficiently.

Along with these three optimizers, we developed two additional calibration methods with the help of ensemble algorithms based on regression trees ; in particular : CatBoost [13, 14] and Random Forest [15, 16, 17] (hereafter CB and RF). These two algorithms, issued from Machine Learning (hereafter ML) applications, are known for their ability to handle high-dimensional data and complex nonlinear relationships. CB Regression is a gradient boosting algorithm that uses decision trees to make predictions. It is particularly effective for handling categorical features and dealing with missing values in the data. The algorithm works by iteratively adding decision trees

to the ensemble, each tree being trained on the residuals of the previous trees. During training, the algorithm uses a variety of techniques to prevent overfitting, such as gradient-based regularization, feature permutations, and learning rate annealing. RF Regression, on the other hand, is also a decision tree-based algorithm that uses an ensemble of trees to make predictions. This algorithm works by creating multiple decision trees on different subsets of the data and averaging their predictions to reduce overfitting. During training, the algorithm randomly selects a subset of the features for each tree and also randomly selects a subset of the data points (with replacement) to build each tree. This introduces randomness into the model and helps to prevent overfitting. The main advantage of these two models, compared to the three optimizers mentioned above, is that once they are trained and optimized, with a suitable database, the predictions are made almost instantaneously, which greatly reduces the calibration time and therefore, can be used for financial market data calibration on the fly.

Explainable ML models, such as CB and RF, are crucial in the financial industry to comply with regulations and manage risks. Without transparency, ML models can have unintended consequences such as perpetuating biases or introducing new sources of risk. From an industrial perspective, it can be challenging to implement ML methods in finance due to the non-explainability of black-box algorithms.

In this article we set out the goal, by doing numerical experiments, of comparing several metaheuristic optimization algorithms and ML models, in terms of their performance in time and precision, to determine the coefficients of the SVI model that best fit the data. To do this, we synthesized a database from market swaption price data with different maturities, to compute the implied volatilities (hereafter IV). This synthetic database allowed us to observe the performance of the aforementioned optimizers and ML models, measured by the optimization and prediction times and the error they commit while estimating the parameters of the SVI model, measured by the root mean squared error (hereafter RMSE). In the case of the ML models, they will be previously trained and optimized with a subsample containing 70% of the total data, and their performance will be assessed with the remaining 30%, in order to discard overfittings. We proceeded then to evaluate the execution time of each prediction, as well as the corresponding RMSE using random subsamples. All computations and numerical experiments were done on a personal computer (PC). The PC had the following specifications : Dell Laptop, Intel Core i7 processor (2.80 GHz), and RAM 16 GB. The computations were performed using Python 3.9, with libraries such as : numpy, pandas, matplotlib, seaborn, scikit-learn, scipy and catboost, running on a Windows 11 operating system.

This article is organized as follows. The Section 2 is dedicated to introducing the concepts that will guide our study, such as the SV concept, the used SVI formula, as well as the loss function to be optimized in such calibration problems. In Section 3, the generation of the synthetic database will be discussed, and the results of the numerical experiments with the optimizers and ML models will be presented. Section 4 is dedicated to the summary and conclusions of this article.

2 Stochastic Volatility Inspired model for option pricing

The SV phenomenon is a term used to describe the tendency of options with the same maturity and underlying asset to have different IV levels depending on their strike price. Typically, when the strike price of an option is closer to the current market price of the underlying asset, the IV tends to be higher than when the strike price is further away. This creates a "smile" shape, shown in Fig. 1, when the IV levels are plotted against the log-moneyness or strikes and fixed time-to-maturity, hence the term "smile volatility".

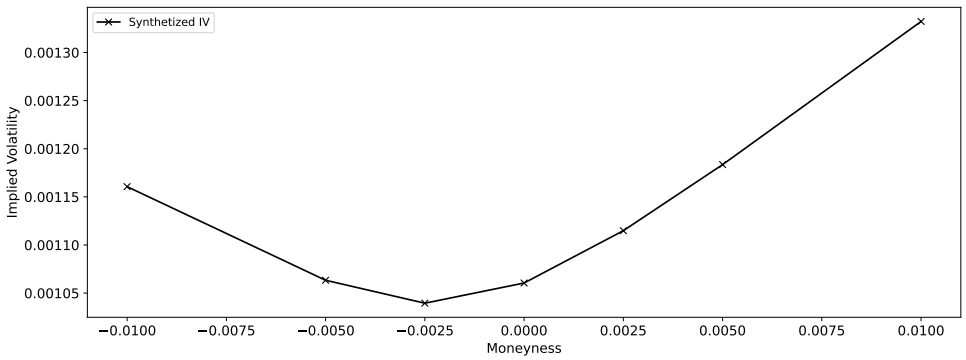

Figure 1 Smile of volatility as a function of the log-moneyness or strikes and fixed time-to-maturity.

This phenomenon is often observed in swaption and equity markets, where investors have different expectations of how the underlying stock price might move over time. The SV phenomenon can have important implications for options pricing, risk management, and trading strategies. Investors and traders need to be aware of this phenomenon and incorporate it into their decision-making processes to effectively manage their options positions. Other common forms existed in the market depending on conditions, like Skew or Smirk [18], which are out of the scope of the present article.

2.1 Stochastic Volatility Inspired model

The SVI is a parametric model for the implied volatility surface of options that was introduced in [3], based on the assumption that the volatility of an option's underlying asset follows a stochastic process, which is modeled as a square-root process. The SVI model then uses a specific functional form to describe the implied volatility as a function of the log-moneyness and time-to-maturity. The parameters of the SVI model are calibrated to observed market prices of options, making it an empirical model. The calibration process involves finding the parameter values that minimize the difference between the model-implied prices of options and their observed market prices, subject to various constraints that ensure the model is arbitrage-free.

Empirical models like the SVI model are commonly used in finance because they provide a flexible way to capture the complex dynamics of financial markets, while also allowing for straightforward calibration to market data. However, it is important to note that empirical models are not based on fundamental economic principles and are therefore subject to certain limitations and assumptions that may not always hold in practice. The commonly used raw SVI model's parameterization, for a given maturity, follows as :

$$\sigma^2_{SVI}(k;\theta) = a + b\{\rho(k-m) + \sqrt{(k-m)^2+\sigma^2}\}, \tag{1}$$

where $\theta = (a, b, \rho, m, \sigma)$ is a real valued vector representing the parameters of the model, namely by order : vertical shift, slope, tilt, horizontal shift, and curvature parameters, respectively ; while $k = \ln(K/F_T)$ represents the log-moneyness, with K and F_T as the corresponding strike and forward prices of the underlying asset. It is important to recall that in this model, usually the log-moneyness is taken into account, however, in the case of interest rate data, negative prices might

exist so one must consider the moneyness instead. On the other hand, the vector parameter θ is not arbitrary and its components are constrained by : $a, m \in \mathbb{R}$, $b \geq 0, |\rho| < 1, \sigma > 0$, and except m, all of them must satisfy the positivity condition : $a + b\sigma\sqrt{1-\rho^2} > 0$, which directly implies that $a > 0$, to ensure that the minimum of the total IV is always positive. Moreover, the AOA (Absence of Arbitrage) constraints in the SVI model are conditions on θ that must be satisfied to ensure the model's validity and avoid arbitrage opportunities in the financial market. The Roger Lee's moment formula [19], imposes that the slope of the right asymptote of the positive branch of the hyperbola Eq.(1) satisfies the condition $b(1+\rho) < 2$, thus, forcing b to lie in the interval $[0, 1)$.

2.2 Calibration to market data

The calibration process of the SVI model involves finding the values of θ in Eq.(1) that fit the best the observed IV at a given maturity. This is typically done by minimizing the distance between the IV predicted by the model and the observed data points, typically measured in terms of the MAE, MSE or similar metrics. To perform the calibration, a dataset of observed IV for a range of strikes and maturities is needed. The dataset is then used to estimate θ using an optimization algorithm, such as the Nelder-Mead, Basin-Hopping or Differential Evolution methods. The resulting θ values are then used to generate the IV curve predicted by the model, which can be compared to the observed data to evaluate the quality of the calibration. It is important to note that the quality of the calibration depends on the quality and representativeness of the dataset used, as well as the appropriateness of the model assumptions. Therefore, it is essential to carefully evaluate the calibration results and conduct sensitivity analyses to assess the robustness of the model to different datasets and parameter values.

In this article the MAE and MSE will be used to calibrate the SVI model with different optimizers and ML algorithms. Defining Θ as the set of all θ that satisfy the constraints described in the previous subsection, and σ_i^2 the IV market data, then the calibration reduces in finding the solutions θ from :

$$MAE : \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} |\sigma_{SVI}^2(k_i, \theta) - \sigma_i^2|, \tag{2}$$

$$MSE : \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} (\sigma_{SVI}^2(k_i, \theta) - \sigma_i^2)^2. \tag{3}$$

The advantages of using MAE are its robustness to outliers as it is less sensitive to them, while the advantages of using MSE are its sensitivity to larger errors. The choice of loss-function depends on the specific problem and the trade-off between these advantages, as we shall see in the next section. In either case, we're dealing with a non-convex optimization problem that suffers from problems of the type : multiple local and/or non-analytic minima, slow convergence, sensitivity to initial conditions, no guaranteed optimal solution, and computationally expensive. For these reasons, in the following section, we will perform some numerical experiments with these two loss-functions, different optimizers, and ML algorithms to compare the calibration performance.

3 Numerical experiments

In this section we will dedicate ourselves to carrying out a series of numerical tests on various optimization and ML algorithms, in order to observe their performance when confronted with databases. To do this, we will synthesize data from real observations of market interest rates, the latter are not abundant or are highly confidential, and therefore, we need an artificial source to measure the speed of the algorithms when optimizing the parameters of the model used, the error that is made in their estimations, as well as details about the algorithms, such as : the number of evaluations that are carried out on the loss-functions, the number of iterations that the algorithms carry out until reaching the optimal solution, as well as the distance from the initial guess point to the optimal solution.

3.1 Data source and synthetization

For our numerical experiments, we took a confidential database consisting of 252 swaptions prices with maturities $T = [5, 10, 15, 20]$ and tenors $t = [1, 2, 5, 10, 15, 20, 30]$ years, respectively. Given the nature of the data involved, which includes banking sensitive information, it is imperative to protect the privacy rights of entities and adhere to ethical guidelines. Access to this confidential database was granted under strict confidentiality agreements and legal obligations. Although specific details about the database cannot be disclosed due to confidentiality restrictions, it is important to emphasize that the study followed established protocols to handle and secure the data appropriately. Internal validation and verification procedures were conducted to ensure the reliability and accuracy of the data used in this research article. These measures were taken to uphold ethical standards and comply with institutional policies and legal requirements regarding data confidentiality. By utilizing this confidential database, this study aims to provide valuable insights while maintaining the utmost respect for privacy and data protection. Furthermore, the amount of data provided is not enough to extract statistical insights about the SVI calibration, so we decided to synthesize data from this original database using some guidelines.

In a first step, we performed a calibration of our initial data using Eq. 1 and Limited-memory Broyden Fletcher Goldfarb Shanno algorithm (hereafter L-BFGS-B) ([21, Chapter 7, Section 7.2]), which is a Quasi-Newton Algorithm (hereafter this method will be denoted by CM). This calibration provided us with an initial estimation of θ ranges (intervals for each parameter in θ). Then, in order to be conservative and avoid future extrapolation, we chose to widen these ranges. This approach allows us to account for potential changes in market regimes, ensuring that our model remains valid across different scenarios. It is important to remark that this widening process needs to be repeated whenever a significant change in regime or range is detected.

Parameter	Observed (Benchmark)	Sampling
a	$[9 \times 10^{-4}, 5 \times 10^{-3}]$	$[-5 \times 10^{-4}, 9 \times 10^{-3}]$
b	$[0.03, 0.13]$	$[1 \times 10^{-3}, 2 \times 10^{-1}]$
ρ	$[-0.89, 1]$	$[-1, 1]$
σ	$[4 \times 10^{-5}, 1 \times 10^{-1}]$	$[2 \times 10^{-5}, 1 \times 10^{-1}]$
m	$[-2 \times 10^{-3}, 2 \times 10^{-3}]$	$[-2 \times 10^{-3}, 2 \times 10^{-3}]$
Prices $p^{(model)}$	$[2 \times 10^{-3}, 6 \times 10^{-3}]$	$[0.0, 2 \times 10^{-2}]$

Table 1 Range of parameters generated and observed.

For each set of simulated parameters within these expanded intervals, we employed the SVI model to generate an IV curves. Consequently, at each data point, we captured a specific combination of the IV along with their corresponding parameter values.

To maintain the quality of the synthetized data, we adopted a straightforward accept-reject method, specifically targeting the ATM (At-The-Money) IV. By setting a predetermined threshold [1] only θ data points with an IV value below this threshold were accepted, while those exceeding the threshold were rejected and omitted from the final synthetized database. This constraint ensures that our synthetic sample predominantly comprises data points that are aligned with our desired level of volatility. Consequently, we achieved a synthetic data sample, of cardinality 10^5, that is reliable and well-suited for subsequent analysis and modeling of IV dynamics. In Fig. 1 is shown one random data point from this synthetic sample.

We generate 7 moneyness values, since we're going to use later ML models, and we don't want any compromise with bias and variance.

Then we have also tried to model a time-based approach.

Throughout the analysis of the behavior of the corresponding IV curves over extended periods, derived from the original database, we observed that the curves maintained a certain shape throughout their evolution, and in many cases, only a specific portion of the smile was affected

1. A bound on the distribution law so that it does not spread over all real numbers.

by these changes ; in addition, the return distributions appeared to deviate from the usual ones. Based on these observations, we devised our dataset generation approach.

Let denote by $R(k)$ the difference between the new and the old curve for each strikes.

For each curve, we introduced a variable, denoted as L_R, which determined where the curve will be modified (Left or Right with respect to the ATM). Additionally, we incorporated a variable depending on the simulated return ATM, denoted as D, determining the direction of the curve ($D = +1/-1$ indicating an "upward/downward" movement).

$$\begin{aligned} D &= 1, if R(0) \geq 0 \\ &= -1, else. \end{aligned} \tag{4}$$

Using these variables, we modified the curve by simulating the changes according to a combination of probability distributions, guided by the given direction and sense.

All the futures return will be given by the following relation,

$$R(k) = D \cdot |r_k|, \tag{5}$$

Where r_k is a simulation from the return distribution. This formula ensures us a correct evolution of the curve.

By adopting this methodology, we created a dataset that captured the dynamic nature of the IV curves, incorporating both directional changes and variations in shape. This approach allowed us to generate a comprehensive dataset for further analysis and modeling of implied volatility dynamics.

It is important to note that we incorporated two distinct conditions : normal and stressed. The normal condition represents the standard evolution of the implied volatility curves, while the stressed condition accounts for exceptional market conditions or events. In the stressed condition, instead of introducing a completely different distribution, we reshaped the existing distribution by adjusting the maximum variation from 5% to 25%. This adjustment allowed us to capture the unique characteristics and dynamics of the implied volatility curves during stressed periods, while maintaining consistency with the underlying distribution observed in normal conditions. By adopting this methodology, we created a 10^5 large sample with values of θ that captured the dynamic nature of the provided IV curves, incorporating both directional changes and variations in shape. This approach allowed us to generate a comprehensive database for further analysis and modeling of IV dynamics.

In the end, we had 2 databases, one fixed and one temporal.

3.2 SVI model calibration

The calibration process, as described in the previous section, consists on finding the optimal parameters θ that fit the SVI model Eq.(1) the best, with the available data. There are five parameters in θ to estimate out of seven available ordered pairs of the form $(k_i, iv(k_i))_{i=1,\dots,7}$ from the data, being k_i the moneyness and $iv(k_i)$ the corresponding IV value at k_i ; therefore, the risks of the model not capturing the properties of the data are high. To circumvent this obstacle, before performing the optimization with NM, BH and DE, we interpolate the data using a cubic spline [2]. Once this is done, with the help of the interpolation function, we generate 20 points, including the original ones, in order to help the optimizers to capture the interaction between the data, and to find the optimal θ. During the calibration using these optimizers, we established a search space within intervals 20% wider than the maximum and minimum of each of the synthetized θ components, to allow room for a more exhaustive search, while the initial guess points were set at the center of these intervals.

In the beginning, we had decided, to find the optimal θ using the three optimizers mentioned above, solving the problem MSE in Eq.(3), as suggested in the related literature ; however, during the numerical experiments, we observed that NM optimizer was oftenly stucked at unstable local

2. We tried other interpolation techniques, and this turned out to be the most accurate.

minima, and failed to calibrate well the Eq.(1) on randomly selected IV curves ; for this reason, we decided, for this particular optimizer, to solve the problem MAE in Eq.(2) instead, which notably improved the results. For the BH and DF optimizers, the problem MSE was solved. Regarding the ML models, the approach is different, as it is a subsample method, unlike a point-wise one as in the previous cases. Indeed, the synthetic sample is made of the moneyness, the inferred IV, and the θ values. The pairs $(k_i, iv(k_i))_{i=1,\dots,7}$ are used as predictor features, while the synthetized θ is the target one ; it is then a supervised ML problem of nonlinear multioutput regression.

To produce the models, we randomly splitted the synthetized sample into two subsamples : one to train and one to test them. The test subsample, which represents 30% of the total sample, is randomly chosen and it has not been previously seen by the algorithms, and therefore, they are only used to evaluate their performance and predictive power. Moreover, the ordered pairs $(k_i, iv(k_i))_{i=1,\dots,7}$ are casted into vectors of fourteen coordinates that will be injected into both models, to compute the five SVI parameters in θ. With the training subsample, we perform a cross-validation grid search procedure to optimize the hyper-parameters of each algorithm, which took 3.94 hours to optimize and to produce the CB model, while for the RF model took 1.9 hours. This step is essential to avoid overfittings and to generalize as much as possible the produced models to unseen data. While gridsearching, we used the MSE and the MAE as scoring metrics, but for in both cases, the last one overperformed. Once optimized both algorithms, two models were produced, and then we proceeded to evaluate their performance using instead the MSE, in order to be able to compare to the other optimizers. The MSE for CB and RF models on the train subsample are of 0.059 and 0.128 respectively ; while on the test subsample, the obtained MSE in both cases is of 0.067, indicating that the produced RF model is slightly overfitted. Nevertheless, we kept it to perform numerical experiments. In addition, for both models, we evaluated the significance of the variables in the prediction process, measured in both cases by the mean impurity decrease [3], and shown in Fig. 2, where only the most prominent contributions are shown.

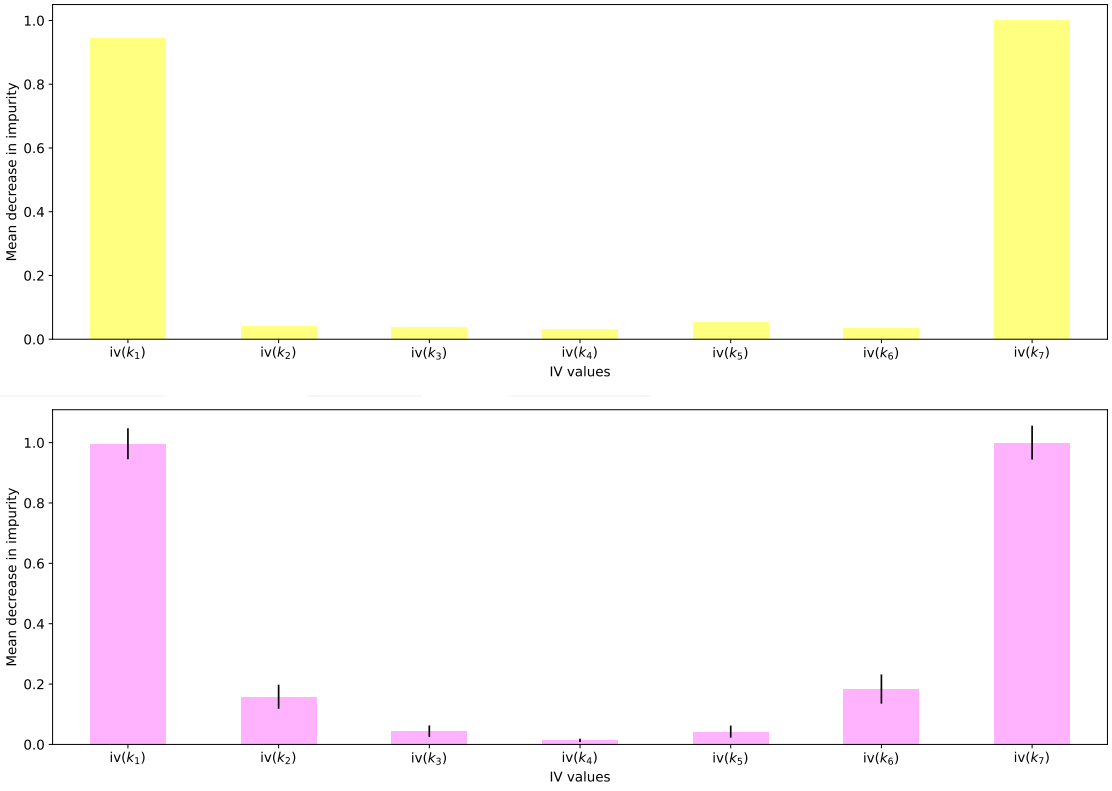

Figure 2 (color online) Feature importance, measured by the mean impurity decrease for the CB (upper panel) and RF (lower panel) models. The standard deviations on each feature are shown for the RF model as vertical black lines.

3. The mean impurity decrease is a metric that measures the importance of each feature in the model's decision-making process.

Recall that in a RF regression model, the mean impurity decrease is calculated by averaging the impurity decreases across all decision trees in the forest. It represents the average reduction in impurity (e.g., Gini impurity or entropy) achieved by splitting on a particular feature across all decision trees. Similarly, in a CB regression model, the mean impurity decrease measures the average reduction in the loss function achieved by splitting on a specific feature across all the boosting iterations. CB, being a gradient boosting algorithm, builds an ensemble of weak regression trees iteratively. The mean impurity decrease reflects the average contribution of a feature to the improvement of the loss function throughout the boosting process. A high mean impurity decrease indicates that the feature plays a more significant role in the overall predictive power of the model, as it is shown in both cases. The models, therefore, are more sensitive to b and ρ, which control the scaling of the linear component of the IV curve.

In Fig. 3 are shown, in black symbols, a randomly chosen data point from the synthetic database ; the solid black line represents the SVI model with the parameters generated synthetically. The blue, green, and red lines correspond to the calibrations performed with the NM, BH, and DE optimizers, while in yellow and magenta the CB and RF ones.

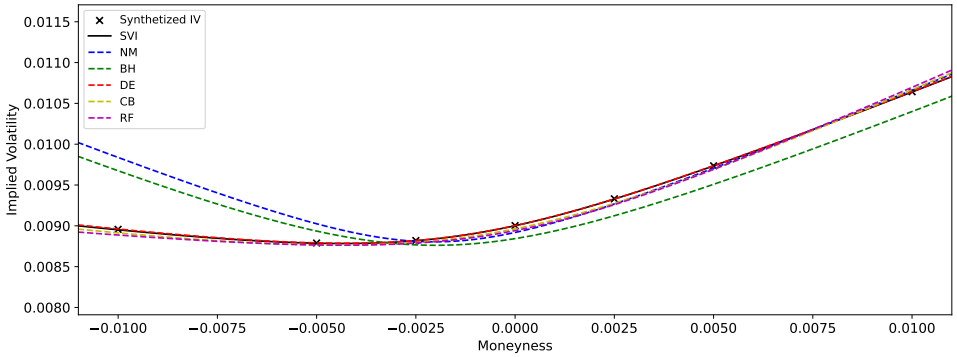

Figure 3 (color online) Different calibrations of the SVI model on a random synthetic data point (black symbols), using the NM (blue line), BH (green line) and DE (red line) optimizers, and the calibration by ML models CB (yellow line) and RF (magenta line).

The five calibrations carried out capture till some extents the behavior of the IV as a function of the moneyness. The best calibration was achieved by the DE optimizer, followed by CB and RF. The calibrations with the less performance were obtained with the BH and the NM optimizers ; however, given the stochastic nature of all these five algorithms, results may vary, which is why we will perform some sensitivity analysis to compare their performance with randomly selected data from our synthetic database.

3.2.1 Calibration performance

In this subsection, we compare the performance of the NM, BH and DE optimizers, as well as the CB and RF models, regarding the error made during the calibration and predictions, measured by the RMSE. In the case of the NM, BH and DE optimizers, the RMSE was computed using the synthetized IV values and the ones predictec by Eq.(1) once evaluated at the optimal θ, at the provided synthetic values of the moneyness k_i. For the CB and RF models, the RMSE was computed using the synthetic values of θ, and the ones predicted by the models. The RMSE is computed by randomly selecting, with replacement, 10^4 data points, and the results are shown in Fig. 4 as violin plots.

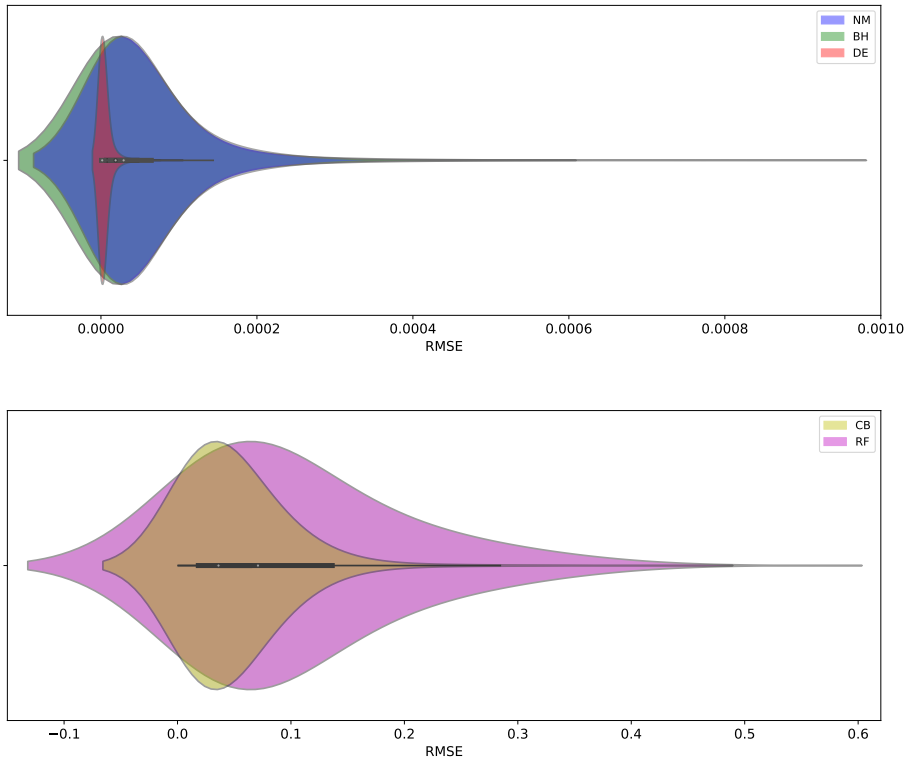

Figure 4 (color online) Violin plots of the RMSE for NM (blue), BH (green) and DE (red) (upper panel), and RMSE for CB (yellow) and RF (magenta) (lower panel).

The minimum, maximum and median RMSE of NM, BH, DE, CB and RF are shown in Tab.2. Of the five calibration methods used, the best is the one provided by the DE optimizer, which demonstrates its superior efficiency in finding optimal solutions for θ with the given data.

	NM	BH	DE	CB	RF
Min	5.684×10^{-8}	1.030×10^{-7}	1.728×10^{-8}	6.700×10^{-4}	7.540×10^{-4}
Max	5.230×10^{-4}	8.760×10^{-4}	6.580×10^{-5}	0.423	0.471
Median	2.909×10^{-5}	1.880×10^{-5}	1.636×10^{-6}	0.036	0.071

Table 2 Minimum, maximum and median RMSE.

3.2.2 Optimization and prediction times

In this subsection, we compare the optimization time of NM, BH, and DE. For the CB and RF models, we compare instead, the time they take to make a prediction. The optimization/prediction time was measured, as mentioned in the previous subsection, by randomly selecting, with replacement, 10^4 data points, and computing the time the optimizers/models took to optimize/predict the outcome θ. The distributions of the optimization and prediction times are shown in Fig. 5 in violin plots.

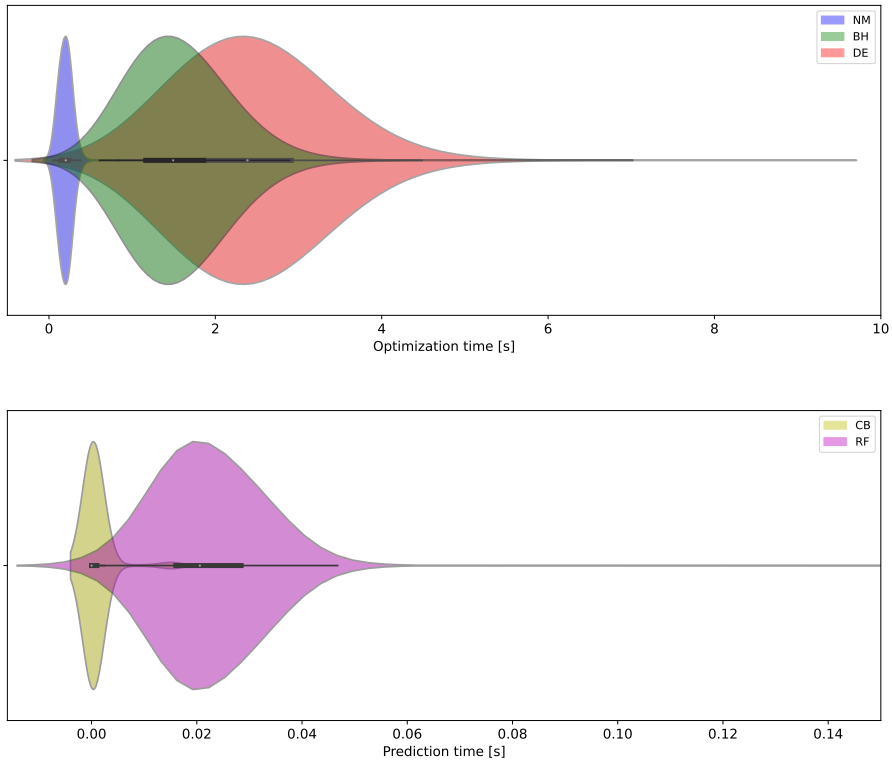

Figure 5 (color online) Violin plots of the optimization times (upper panel) for NM (blue), BH (green) and DE (red), and prediction times (lower panel) for CB (yellow) and RF (magenta).

The minimum, maximum and median optimization times (in seconds) taken by NM, BH and DE, as well as the prediction times (in seconds) of CB and RF are shown in Tab.3. This finding suggests that the CB model demonstrates superior efficiency in finding optimal solutions within a given time frame. The comparison of optimization and prediction times provides valuable insights into the performance of these algorithms and informs the selection of the most appropriate optimization approach for complex problem domains.

	NM	BH	DE	CB	RF
Min	0.047	0.608	0.818	$< 10^{-6}$	$< 10^{-6}$
Max	0.744	6.212	8.484	0.017	0.272
Median	0.202	1.493	2.386	$< 10^{-6}$	0.021

Table 3 Minimum, maximum and median optimization and prediction times (in seconds).

Given the high speed of predictions of both ML models, we wanted to check if providing the initial guess point for NM, BH and DE, using CB and RF, would decrease their optimization times. The results are shown in Fig. 6.

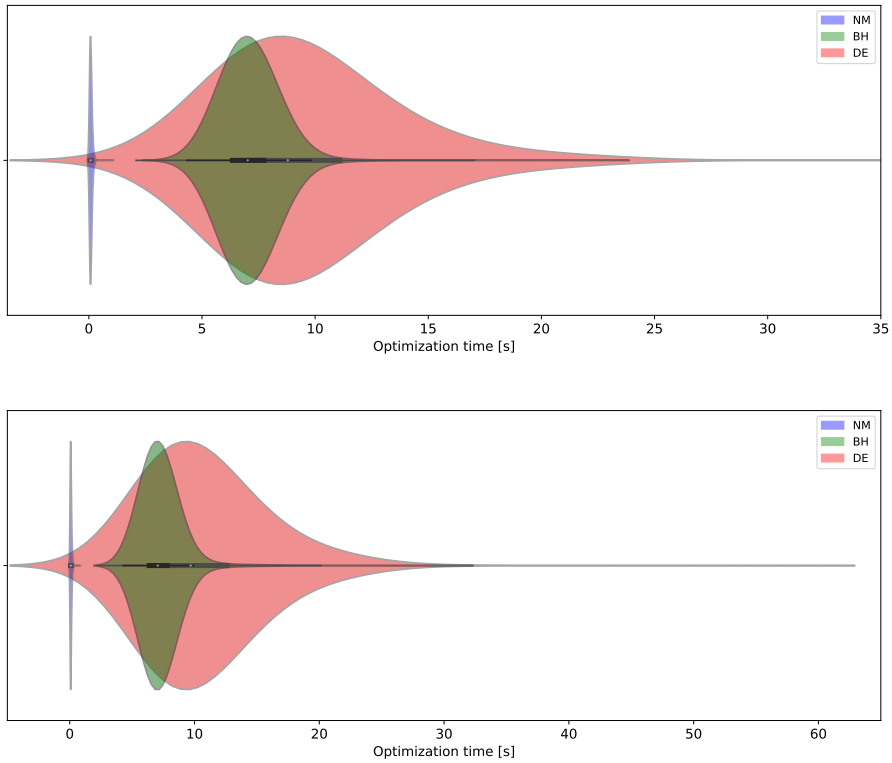

Figure 6 (color online) Violin plots of the optimization times for NM (blue), BH (green) and DE (red), provided the inital guess point by CB (upper panel) and RF (lower panel).

In Tab.4 are shown these results, point out that for BH and DF, this strategy is not effective, which might be explained by the very nature of both algorithms. However, for the NM optimizer, there was a slightly improvement in the optimization time but it is not yet significant.

Model		NM	BH	DE
CB	Min	$< 10^{-6}$	4.143	2.091
	Max	1.021	22.134	29.648
	Median	0.079	7.023	8.793
RF	Min	0.016	4.268	2.205
	Max	0.781	30.000	55.937
	Median	0.079	7.057	9.693

Table 4 Minimum, maximum and median optimization times (in seconds) for NM, BH and DE, providing the initial guess points by CB and RF.

3.2.3 Time Series Calibration with Prior Knowledge

In the preceding sections, we discussed our calibration method, which relied solely on current information without considering the past. In real-life scenarios, we usually have a previously established parameter. As such, we can adjust the calibration by searching for local minima near the previous value of θ.

This approach is highly advantageous because the market's shape does not tend to change rapidly under normal conditions. By using the previous calibration as a starting point and adjusting it based on the local minima found, we can create a precise calibration that takes into account both past and present information.

Overall, by considering both past and present information, our calibration method can help us better understand and navigate the dynamic and ever-changing world of financial markets.

To thoroughly evaluate the robustness and effectiveness of our models, we subjected them to stress tests simulating extreme market conditions. This approach allowed us to assess the models' performance under challenging circumstances and provided us with valuable insights into their limitations and potential weaknesses.

Two-Step Ensemble Approach

To improve the precision of our model and minimize potential errors, we opted for a two-step approach in ensemble methods. Although this method requires additional time, the time complexity of ML techniques is low enough that the overall time required for our approach can still be comparable to that of traditional methods, but with the added benefits of higher precision and accuracy.

For the minimization algorithm, we decided to use the L-BFGS-B.

Furthermore, we chose this algorithm to avoid the black-box effect often associated with ML models, by using a well-established and validated optimization technique in the final step.

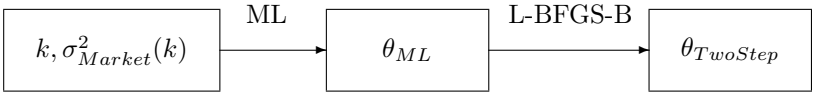

Figure 7 Two-Step Ensemble methodology.

In order to have a better vision of the results, we can synthesis like this.

Normal Situation			
Optimisation method	Mean Time (s)	MSE	Variance
L-BFGS-B CB	$2e^{-3}$	$3e^{-8}$	$3e^{-8}$
L-BFGS-B RF	$1e^{-2}$	$1e^{-7}$	$1e^{-7}$
L-BFGS-B	$5e^{-3}$	$3e^{-7}$	$4e^{-7}$
Stressed Situation			
Optimization Method	Average Time (s)	Average Error	Std Error
L-BFGS-B CB	$4e^{-3}$	$3e^{-8}$	$5e^{-8}$
L-BFGS-B RF	$3e^{-2}$	$5e^{-8}$	$8e^{-8}$
L-BFGS-B	$7e^{-3}$	$8e^{-7}$	$1e^{-6}$

Table 5 Synthesis of both normal and stressed situations with prior knowledge for CB, RF and CM.

The two-step ensemble approach could be a valuable tool that can provide a more accurate and comprehensive view of model performance and potential errors. This approach is especially relevant in financial markets, where accurate predictions are crucial and errors can have significant consequences.

We evaluated the performance of the models under both normal market conditions, with a 5% variation in implied volatility, and stressed conditions, with a 25% variation in implied volatility.

The results from the stressed situations indicate that the CM with L-BFGS-B optimization is not able to adapt to extreme market conditions, and thus, its performance is not satisfactory. On the other hand, the CB algorithm has shown its strength by exhibiting robustness and adaptability to such scenarios.

Furthermore, the lower average time and mean square error achieved by the CB algorithm compared to the CM and RF algorithm demonstrates the superiority of the former in terms of

computational efficiency and accuracy. These findings emphasize the potential of machine learning techniques in the financial domain, particularly in providing reliable and precise predictions in a short amount of time. This highlights the importance of using machine learning techniques that can handle non-linear and complex relationships in financial data.

4 Conclusion

In this article, we have explored the potential applications of heuristic optimization algorithms, such as NM, BH and DE, and ensemble ML models, such as CB and RF, to fast-calibrate the SVI model to financial data. Overall, our study highlights the potential of these optimizers and ML models to enhance financial decision-making, while also emphasizing the need for careful validation of these models. Based on the results obtained from the numerical experiments, we could compare the performance of the optimization algorithms NM, BH and DE, and ML models CB and RF in terms of optimization/prediction times and RMSE values. In terms of optimization/prediction times, CB and RF consistently outperform NM, BH, and DE, exhibiting significantly lower median, mean, and standard deviation values. This indicates that CB and RF are more efficient in terms of computational time compared to the other algorithms. When considering the RMSE values, DE consistently performs the best, followed by BH and NM. CB and RF show slightly higher RMSE values compared to the other algorithms, indicating that they might not achieve the same level of accuracy in prediction tasks as DE, BH, and NM. Considering both the optimization/prediction times and the RMSE values, it can be concluded that DE is a strong performer, delivering relatively low RMSE values while maintaining reasonable optimization times. NM and BH also show competitive results in terms of RMSE, although their optimization times are higher compared to DE.

Considering both, optimization time and RMSE, we can conclude that DE stands out as the best performing optimizer among the five evaluated, and the most suitable algorithm to optimize an SVI model with real live data.

References

[1] Heston, S. ; *A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options*, Rev. Fin. Studies, 6 (**2**), 1993.

[2] Gatheral J. ; *A parsimonious arbitrage-free implied volatility parameterization with application to the valuation of volatility derivatives* ; Presentation at Global Derivatives & Risk Management, Madrid, 2004.

[3] Gatheral, J., Jacquier, A. ; *Arbitrage-Free SVI Volatility Surfaces*, Quant. Fin., 14 (**1**), 2014.

[4] Gatheral, J., Taleb, N. N. ; *The Volatility Surface : A Practitioner's Guide*, (**357**), John Wiley & Sons, 2006.

[5] Gatheral, J., Jaisson T., Rosenbaum M. ; *Volatility Is Rough*, Quant. Fin., 18 (**6**), 2018.

[6] Sangjoon, K., Shephard N., Siddhartha C. ; *Markov Chain Monte Carlo Simulation Methods in Econometrics*, Rev. Ec. Studies, (**65**), 1998.

[7] Lemaréchal C. ; *Cauchy and the gradient method*, Doc. Math. Extra, 2012.

[8] Hadamard J. ; *Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées*, Mémoires présentés par divers savants éstrangers à l'Académie des Sciences de l'Institut de France, France, 1908.

[9] Courant R. ; *Variational methods for the solution of problems of equilibrium and vibrations*, Bull. Am. Math. Soc., 49 (**1**), 1943.

[10] Nelder J.A., Mead R. ; *A Simplex Method for Function Minimization*, The Comp. Jour., 7 (**4**), 1965.

[11] Wales D., Doye J. P. K. ; *Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms*, Jour. Phys. Chem. A, 101 (**28**), 1997.

[12] Storn R., Price K. ; *Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces*, Jour. Glob. Opt., 11 (**4**), 1997.

[13] Prokhorenkova L., Gusev G., Vorobev A., Dorogush A. V., Gulin A. ; *CatBoost : unbiased boosting with categorical features*, NeurIPS, 2018.

[14] Dorogush A. V., Ershov V., Gulin A. ; *CatBoost : gradient boosting with categorical features support*, Workshop on ML Systems at NIPS, 2017.

[15] Ho T. K. ; *Random decision forests*, Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal QC, 1995.

[16] Ho T. K. ; *The random subspace method for constructing decision forests*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20 (**8**), 1998.

[17] Breiman L. ; *Random Forests*, Machine Learning, 45 (**1**), 2001.

[18] Zhang Jin E., Xiang Yi ; *The implied volatility smirk*, Quant. Fin. 8 (**3**), 2008.

[19] Lee. R. W. ; *The moment formula for implied volatility at extreme strikes*, Wiley Blackwell, 14 (**3**), 2004.

[20] Bandi, F. M., Russell, J. R., Yang, C. ; *Journal of Economics*, **147** (1), 2008.

[21] Nocedal J. Wright S. J. ; *Numerical Optimization*, 2006.

A Appendix

The Nedler-Mead algorithm, also known as the downhill simplex method, is a metaheuristic search algorithm that works by iteratively contracting, reflecting, expanding, and shrinking a simplex [4] It has time and space complexities $O(N^2)$ and $O(N)$ respectively, being N number of iterations. The algorithm is robust, but it may get stuck at local and non-analytical minima. When the initial guess point is closer to the optimal value, it means that the simplex, initially constructed around the initial guess point, is already relatively close to the minimum. In such cases, the algorithm has a better starting point and requires fewer iterations to converge to the optimal solution. As the distance between the initial guess point and the optimal value decreases, the simplex is more likely to contain the optimal point within its vertices. This effectively reduces the exploration space and guides the algorithm towards the minimum. Consequently, the algorithm converges faster, and the number of iterations decreases, as it is shown in Tab.6, Tab.7, Fig.8 and Fig.9.

Model		NM	BH	DE
	Min	129	100	34
	Max	649	100	225
	Median	461	100	107
CB	Min	13	100	100
	Max	640	100	1000
	Median	180	100	390
RF	Min	47	100	102
	Max	652	100	1000
	Median	173	100	424

Table 6 Minimum, maximum and median number of iterations for NM, BH and DE, and providing the initial guess points by CB and RF.

The Basin-Hopping algorithm, widely used to find the minimum energy structure for large molecules and proteins, is a stochastic global optimizer that combines a local minimization algorithm with a Monte Carlo sampling method. It starts from a random initial point and searches for the global minimum by iteratively perturbing the current solution, performing a local minimization around it, and accepting or rejecting new solutions based on the values of the target function. Since the algorithm tries to find solutions using this strategy, the number of iterations is fixed. It has time and space complexities $O(N \times M)$ and $O(D)$ respectively , where N number of iterations, M average time complexity of evaluating the target function and performing the local search, D number of variables to be optimized. When the initial guess point is far from the optimal value, the algorithm has a larger search space to explore. It needs to perform more iterations to traverse the landscape, evaluate different regions, and potentially find lower minima. As the distance between the initial guess point and the optimal value decreases, the search space becomes smaller. The algorithm needs to perform more precise and detailed exploration to identify the optimal solution accurately. This often involves finer adjustments and smaller perturbations in the vicinity of the current solution. Consequently, more iterations are required to navigate this reduced search space effectively. This is shown in Tab.6, Tab.7, Fig.8 and Fig.9.

The Differential Evolution algorithm is a metaheuristic population-based optimizer that operates on a set of candidate solutions (a population) and evolves the population over successive generations by randomly selecting and recombining individuals to create new candidate solutions. The algorithm utilizes mutation to create new trial solutions by perturbing existing solutions, crossover to combine information from the mutant and target vectors, selection to choose improved solutions for the next generation, enabling iterative exploration of the solution space towards optimal solutions, and can effectively search for the global minimum of non-linear and non-convex target functions.

4. A geometric figure consisting of n+1 vertices in n-dimensional space) in the parameter space.

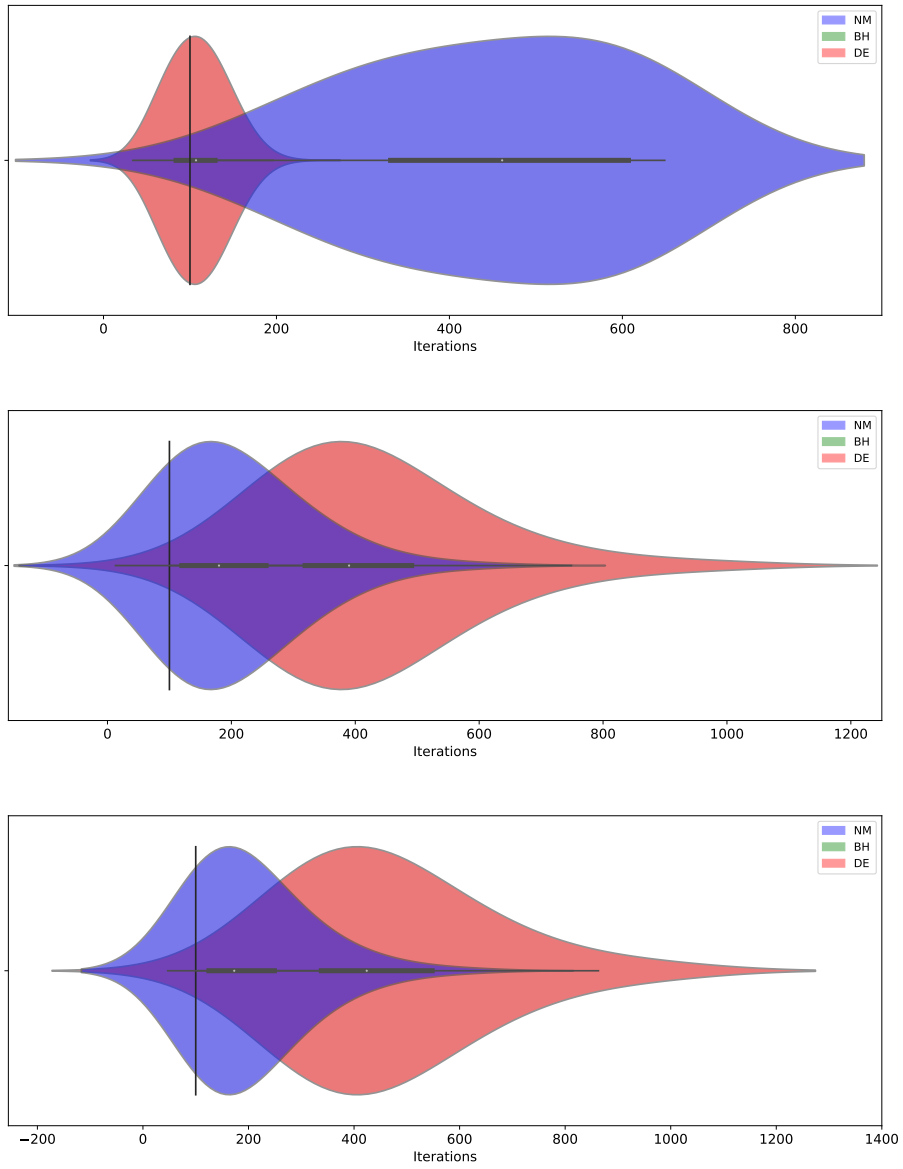

Figure 8 (color online) Violin plots of the number of iterations for NM (blue), BH (green) and DE (red), provided the inital guess point by CB (upper panel) and RF (lower panel).

It time and space complexities $O(G \times N \times D)$ and $O(N \times D)$, respectively, where G is the number of generations, N is the population size, and D is the number of variables to optimize. As the distance between the initial guess point and the optimal value decreases, the algorithm enters a region of the search space that contains more promising solutions. The population starts to converge towards the optimal solution. However, as the algorithm gets closer to the optimal value, the convergence slows down. This is because the search space becomes more narrow and the algorithm needs to make smaller adjustments to improve the solutions.

To ensure the convergence, the algorithm may require additional iterations to make these small adjustments and fine-tune the solutions. This phenomenon is often referred to as "slowing down" or "stagnation" in optimization algorithms. The algorithm spends more time exploring the region near the optimal value to make incremental improvements, resulting in an increase in the number of iterations. This is shown in Tab.6, Tab.7, Fig.8 and Fig.9.

Model		NM	BH	DE
	Min	0.002	0.001	0.006
	Max	1.005	1.005	1.005
	Median	0.090	0.201	0.494
CB	Min	1.610×10^{-4}	3.930×10^{-4}	6.000×10^{-6}
	Max	3.282	3.353	3.219
	Median	0.062	0.081	0.086
RF	Min	0.001	7.560×10^{-4}	2.450×10^{-4}
	Max	2.943	3.551	3.249
	Median	0.103	0.144	0.210

Table 7 Minimum, maximum and median distance from initial guess point to the optimal solution for NM, BH and DE, and providing the initial guess points by CB and RF.

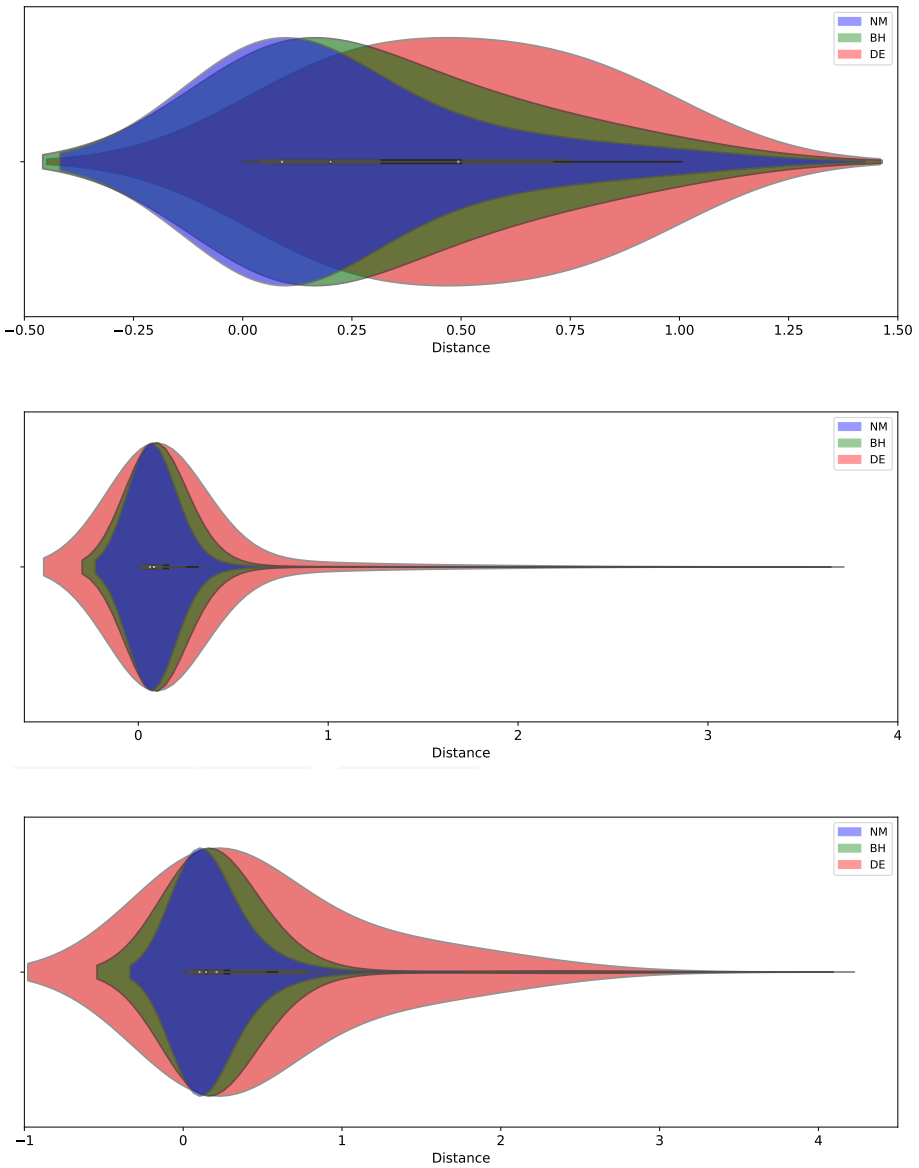

Figure 9 (color online) Violin plots of the euclidean distance for NM (blue), BH (green) and DE (red), provided the inital guess point by CB (upper panel) and RF (lower panel).

Nexialog Consulting est un cabinet de conseil spécialisé en Banque et en Assurance. Organisés autour de 3 domaines d'activité - Risques Bancaires, Financiers & Assurantiels - nous intervenons au sein des équipes métiers afin de les accompagner depuis le cadrage jusqu'à la mise en œuvre de leurs projets. Associant innovation et expertise, le savoir-faire de notre cabinet a permis de consolider notre positionnement sur ce segment et de bénéficier d'une croissance forte et régulière.

Les besoins de nos clients étant en constante évolution, nous nous adaptons continuellement pour proposer le meilleur accompagnement. Le département R&D de Nexialog Consulting se donne pour objectif de proposer des solutions innovantes à des problématiques métier ou d'actualité. Pour cela, nous nous appuyons sur des librairies internes et sur le travail de nos consultants. Le pôle R&D Nexialog a également pour mission de former les collaborateurs sur l'évolution des techniques et la réglementation en lien avec leur activité.

Site web du cabinet : https://www.nexialog.com
Publications : https://www.nexialog.com/publications/

Contacts

Ali BEHBAHANI
Associé, Fondateur
Tél : + 33 (0) 1 44 73 86 78
Email : abehbahani@nexialog.com

Christelle BONDOUX
Associée, Directrice commerciale
Tél : + 33 (0) 1 44 73 75 67
Email : cbondoux@nexialog.com

Adrien MISKO
Senior Manager R&D
Email : amisko@nexialog.com

Areski COUSIN
Directeur scientifique R&D
Email : acousin@nexialog.com