EJE EUROPEAN JOURNAL OF ECONOMICS

ISSN: 2669-2384
VOLUME 1, ISSUE 2

Analyzing the Dynamics of the Swaption Market Using Neural Networks

Sándor Kunsági-Máté(✉), Gábor Fáth and István Csabai

Eötvös Loránd University, Budapest, Hungary
kunsagisandor@gmail.com

ABSTRACT

The SABR stochastic volatility model is a widely used option pricing tool in the financial world. It was the first that could successfully capture the static pattern of the volatility smile while predicting its dynamics much better than previously used local volatility models. This model is often used for hedging where the portfolio is usually hedged to the delta and vega risks representing the uncertainty in the underlying asset price and its volatility. The model assumes that the two non-stochastic parameters - ρ and υ - are stable over time. However, traders regularly recalibrate these model parameters to fit the market, making them stochastic as well and introducing additional parameter risks. Then the question arises whether this behaviour is due to the larger complexity of the market or due to the incorrect model choice? In our study we analyzed the dynamics of the volatility smiles of the GBP swaption market using an autoencoder-like neural network, and created an alternative model, a deformation of SABR, in which we can describe the smiles with two stochastic parameters only as it would be required from a real two factor model. This new model can reproduce well the volatility smiles for several months until it reaches a critical point- the onset of the COVID crisis - where the reproduction error increases suddenly. We found that the sharp discrepancy is caused by a sudden change in the representative low dimensional market manifold, and our model is highly sensitive to pick up this change, making it especially useful for regime change detection.

Keywords: Deep Learning; regime change detection; stochastic models; time series; volatility smiles

Cite this article as: Kunsági-Máté, S., Fáth, G., & Csabai, I. (2021). Analyzing the Dynamics of the Swaption Market Using Neural Networks. *European Journal of Economics*, 1(2), 1-13. https://doi.org/10.33422/eje.v1i2.141

1. Introduction

In the early ‘70s, the Black-Scholes model (Black & Scholes, 1973) put the pricing of options on a rigorous conceptual and mathematical basis. Assuming that the underlying asset volatility is constant was a reasonable simplification to make, and indeed, it was a good approximation for equity markets before the 1987 Black Monday crash. Since then the empirical reality is that options at different strike prices (K) imply different volatilities to match their market prices. This behaviour of the market is called the volatility skew or volatility smile. During the ‘90s several local volatility models appeared which were the generalisation of Black-Scholes and tried to capture the observed volatility smile. While these models could well reproduce the static pattern of the smiles they were unable to predict their dynamics. This problem was resolved later by stochastic volatility models. One approach was proposed by Hagan and his colleagues who introduced the so-called SABR stochastic volatility model (Hagan et al., 2002). This model uses two coupled stochastic differential equations for describing the movement of the underlying asset price and its volatility (see Eq. 1).

© The Author(s). 2021 **Open Access**. This article is distributed under the terms of the Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and redistribution in any medium, provided that the original author(s) and source are credited.

$$
\begin{gathered}
dF_t = \sigma_t (F_t)^\beta dW_t^1 \\
d\sigma_t = \upsilon \sigma_t dW_t^2 \\
dW_t^1 dW_t^2 = \rho dt \\
\alpha = \sigma_{t=0}
\end{gathered} \tag{1}
$$

where F_t, σ_t are the forward price and its volatility, υ is the volatility of volatility, $dW^1{}_t$ and $dW^2{}_t$ are two Brownian processes that are correlated through the correlation parameter ρ and β is the parameter which describes the so called backbone of the volatility smiles. There exists a perturbative result (see Eq. 2.) for the normal volatility implied by the option price which we will use later in our analysis. This approximation holds for relatively small α and υ which were in our case $\alpha \sim 0.005$ and $\upsilon \sim 0.5$.

$$
\sigma_N(\alpha,\beta,\rho,\nu,F,K,T) = \upsilon \frac{(F-K)}{x(z)} \left\{ 1 + \left[\frac{\beta(\beta-2)\alpha^2}{24F_{mid}^{2-2\beta}} + \frac{\rho\beta\upsilon\alpha}{4F_{mid}^{1-\beta}} + \frac{2-3\rho^2}{24}\upsilon^2 \right] T \right\} \tag{2}
$$

$$
z = \frac{\upsilon}{\alpha}\left(\frac{F^{1-\beta} - K^{1-\beta}}{1-\beta}\right)
$$

$$
x(z) = ln\left(\frac{\sqrt{1-2\rho z + z^2} + z - \rho}{1-\rho}\right)
$$

$$
F_{mid} = \frac{F+K}{2}
$$

$$
\sigma_{N,ATM}(\alpha,\beta,\rho,\nu,F,T) = \alpha F^\beta \left\{ 1 + \left[\frac{\beta(\beta-2)\alpha^2}{24F_{mid}^{2-2\beta}} + \frac{\rho\beta\upsilon\alpha}{4F_{mid}^{1-\beta}} + \frac{2-3\rho^2}{24}\upsilon^2 \right] T \right\}
$$

It turns out however that the dynamics of the volatility smile is still not properly described by the SABR model either. Traders have to recalibrate relatively often the ρ and υ parameters to fit the model to the observed data.

As in the case of many fields of science, models in the financial area always get more and more complicated having larger and larger degrees of freedom to better describe a specific problem. The usual model development process consists of two main steps: 1) Find the most relevant parameters of the problem and create a mathematical model that describes the relation between them; 2) Compare the model behaviour with the observations, and go back to step 1, if there is a big discrepancy. This clearly shows that during this iterative process it is really hard to find the number of minimally required parameters and the correct relations among them.

In recent years neural networks made a big impact on several areas of sciences and everyday life, and brought a completely different method in solving problems compared to our previously used deterministic approach. Deep Learning provides a data-driven approach where the model is adapted directly to the observed data. We do not have to care about all possible events anymore because the neural network can itself learn the complex relation between the real parameters, hence providing a better representation of the problem. In our work we used neural networks to find an alternative option pricing model which can well describe the smiles and their dynamics with strictly two stochastic parameters.

1.1. Related work

The stability of model parameters is still an actively studied area in finance, since many practical applications, e.g. hedging and option pricing rely on the estimated long-term dynamics of the underlying asset. Extensive research has been prepared by (Webber & Becker, 2014) where the authors investigated the effect of fixing some of the model parameters when calibrating the SABR and Heston models on the observed data. They analyzed the South

Africen market related to options on futures. They found that the parameter fixing reduces the variance of the remaining parameters, however the calibration accuracy gets worse. Furthermore they found that some parameters (ρ and ν of SABR) are correlated in time. Another redundancy was found earlier by (Hagan et al., 2002 and West, G, 2005), who demonstrated that the β and ρ parameters has similar effect on the volatility skew. In both works the solution was to empirically detect these redundant parameters and fix one of them during the calibration procedure. These findings clearly show that the SABR model is unable to fit real data when we restrict stable and uncorrelated parameters. This indicates that the model itself should be modified in a way that it has less degrees of freedom which is possible due to the observed correlations. In our study we use neural networks to find this modified model where we can force the relevant parameters to be as stable as possible while trying to keep low the calibration loss as well. Until now, the main applications of neural networks in finance were to replicate the already existing pricing models (Liu et al., 2019; Pagnotti et al., 2019; Gaspar et al., 2020). By doing so we are able to build a closed formula that directly maps the model parameters into option prices. Hence, we do not need to run the time-consuming Monte Carlo simulations using the analytical differential equations and the pricing process will be several orders of magnitudes faster than before. This has a major advantage in the model calibration on the observed market data as well (Horvath et al., 2019; Bayer et al., 2019).

This paper is organised as follows: In Section 2 we describe the main details of the data as well as the results of the classical SABR model fit. In Section 3 we explain our methodology and the implemented network architecture. In Section 4 we show our results derived on the training and test sets. In Sections 5 we discuss the key takehome messages of our study and finally in Section 6 we summarize our work.

2. Materials and Methods

A swaption (also known as a swap option) is an option to enter into an interest rate swap. A swaption has two key properties: the expiry and the tenor. Expiry defines the time interval in which one can enter into a swap, and tenor is the duration of the swap. Expiries can range from a few months to even 30 years while tenors are in the range of 1-30 years. This shows that there is a large variety of swaptions. To keep our analysis easy to interpret we chose a specific swaption with 1 year expiry and 1 year tenor. We used historical data for GBP 1y-1y swaptions from ICAP (www.icap.com) with implied normal volatility values at 8 strike points: ATM (at-the-money strike) + [-50, -25, -12.5, 0.0, 12.5, 25, 50, 100] bps (basis points) shifts. The observed time interval is between 2016 February 15 and 2020 October 21, hence we have skew data from 1221 business days. We fitted the asymptotic SABR formula for normal implied volatility on the data using the Levenberg-Marquardt optimization algorithm, where we fixed the β parameter at zero, and searched for the best values of α, ρ and ν parameters. The results can be seen in Figure 1.

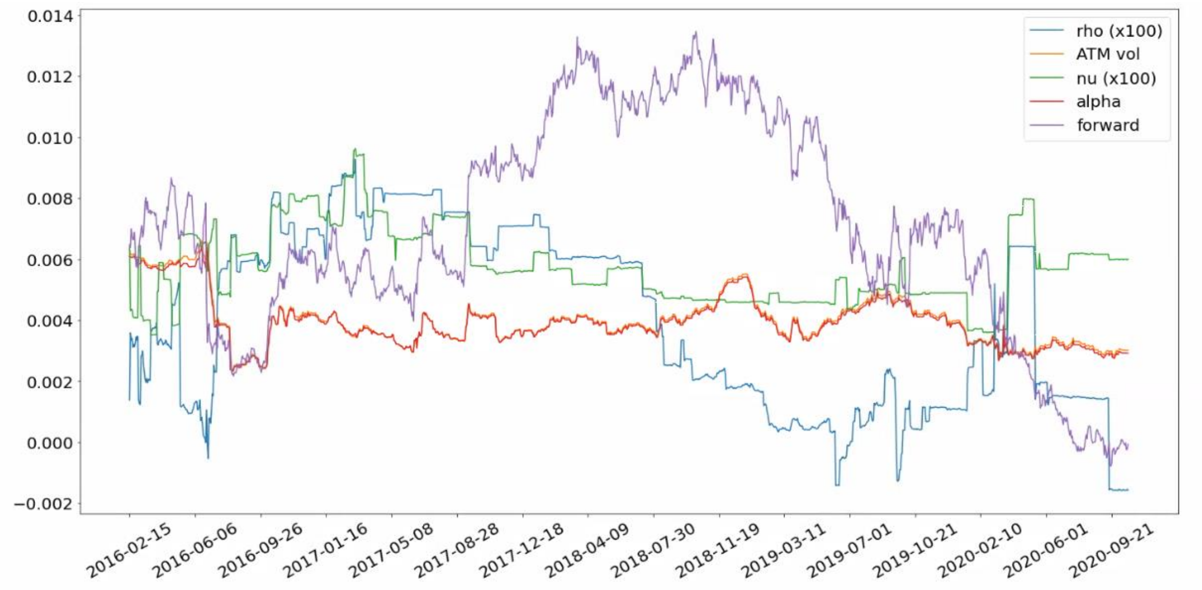

Figure 1: Time evolution of the observable ATM volatility and forward rate as well as the fitted SABR model parameters: α, ρ and ν. For illustration purposes ρ and ν are downscaled by a factor of 100

If the SABR formula would perfectly describe the dynamics of the skews of this specific swaption, we should see that the ρ and ν parameters are almost constant for a long time and only the two stochastic variables - the forward rate and the instantaneous volatility (α) - move over time. However, in reality there are quite a lot of sharp changes in the time evolution of these parameters (as found in (Zhang & Fabozzi, 2016) and (Webber & Becker, 2014)), as seen in Figure 1 (green and blue lines).

This instability of these two parameters are critical in hedging applications, where we account for only the movement of the forward rate and its volatility and suppose that ρ and ν are constant over a long time. In our analysis we tried to find a modified SABR model in which the actual forward rate and the unobservable volatility can describe all of the skews with a given ρ and ν. In this way we would find the "correct" SABR model, where there are strictly two stochastic variables only (the observable forward rate and the unobservable α') and two constant parameters (ρ' and ν').

Neural networks, especially autoencoders, are great tools for such nonlinear dimension reduction problems. However, we do not have enough training data to teach the full dynamics of swaption skews/smiles to a network from scratch and the resulting model is often difficult to interpret, an important requirement in the financial world. To reduce these problems we started from two neural networks - the encoder and decoder part of an autoencoder - which were trained on 1 million synthetic volatility smiles generated by the SABR dynamics. The encoder maps the observed volatility smiles to the model parameters, and the decoder part reproduces the original volatility smile from the model and the observable parameters.

2.1. Generating synthetic data

First we determined the range (*r(...)*) of the observed forward price (F) and the fitted model parameters ($\{\theta_{i,real}\}$) of the analysed swaption time series and defined the range of the synthetic data ($\{\theta_{i,syn}\}$) as follows:

$$r(\theta_{i,syn}) = [min(\theta_{i,real}) - 0.05\, r(\theta_{i,real}), max(\theta_{i,real}) + 0.05\, r(\theta_{i,real})\,]$$
$$r(F_{syn}) = [min(F_{real}) - 0.05\, r(F_{real}), max(F_{real}) + 0.05\, r(F_{real})\,]$$

In this way we slightly extended the range of synthetic parameters relative to the real SABR parameters. This extension is crucial since the interpolation feature of a neural network is typically worse at the edges of the corresponding domain. In Tab. 1. we summarized the ranges of the forward and the model parameters which were used for synthetic data generation.

Table 1.
Ranges of the forward and the model parameters used for synthetic data generation

B	0.0
F	[-0.0015, 0.0142]
A	[0.0021, 0.0071]
P	[-0.2166, 1.0]
Y	[0.1424, 1.0020]

The parameters used were uniformly distributed inside the defined domain, and we generated altogether 1 million volatility skews each containing 8 volatilities at the same 8 strikes as in the real dataset. These 8 volatilites and the forward value were given to the input layer of the encoder network which predicted the α, ρ, υ model parameters. The decoder network was designed as the replication of the SABR model, which means that it predicts a volatility value at a single strike using the three model parameters and the forward value. As a consequence the decoder has been trained on the individual points of the synthetic skews. The data set was split into a training and test set using a ratio of 7:3, respectively. We used the real dataset as the validation set. In Figure 2. we have illustrated the distribution of the real and synthetic parameters.

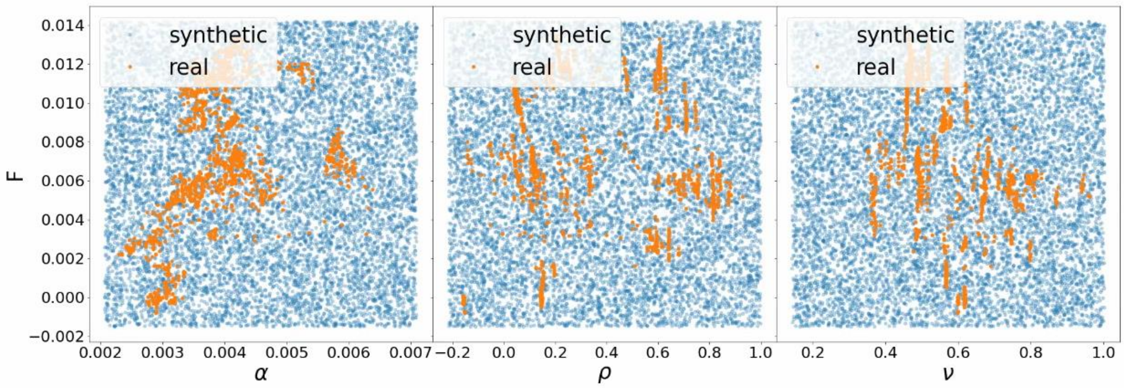

Figure 2: Distribution of the forward rate as a function of the synthetic (blue dots) and real (orange dots) model parameters

2.2. Training the encoder

We used three fully connected hidden layers with 32 units with elu activation function and one readout layer. The batch size was set to 2000 and the model was trained with ADAM optimizer for a few hours. We used the normalized root mean squared error (NRMSE) for the measure of accuracy, where the RMSE was normalized by the size of the domain of the ground truth values. The model was evaluated on the test set which can be seen in Figure 3.

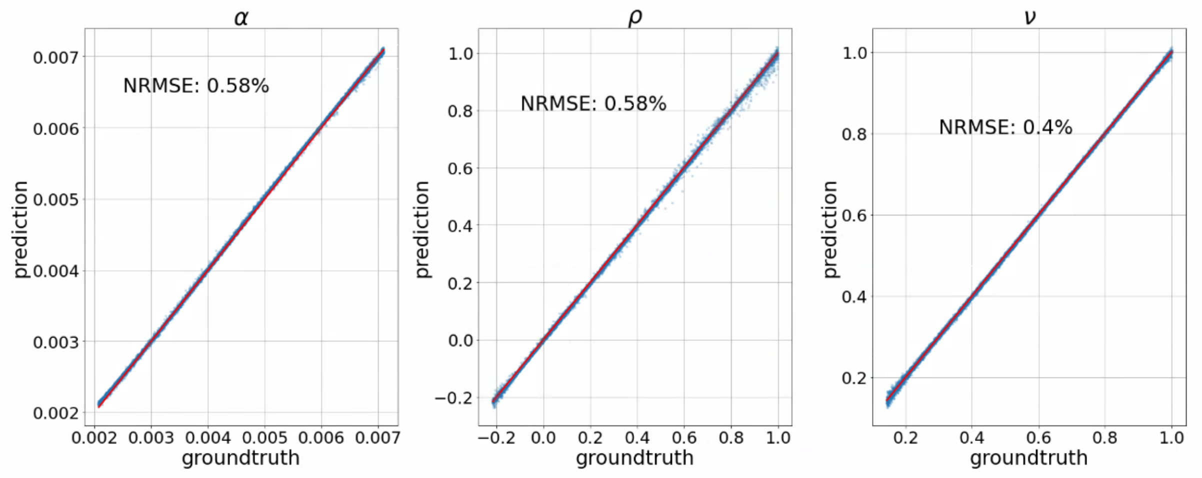

Figure 3: Prediction efficiency of the encoder network regarding to the α, ρ, υ model parameters

The model produced a high prediction efficiency: in all parameters the NRMSE value is below or equal to 0.61 %.

2.3. Training the decoder

The decoder contains four fully connected layers with elu activation function and one readout layer. The network predicts a single volatility value given the relative strike (*ΔK*), the forward value and the three SABR model parameters. The batch size was set to 2000 and we used again the ADAM optimizer. After a few hours of training, we got an accuracy of NRMSE=0.19% (see Figure 4.).

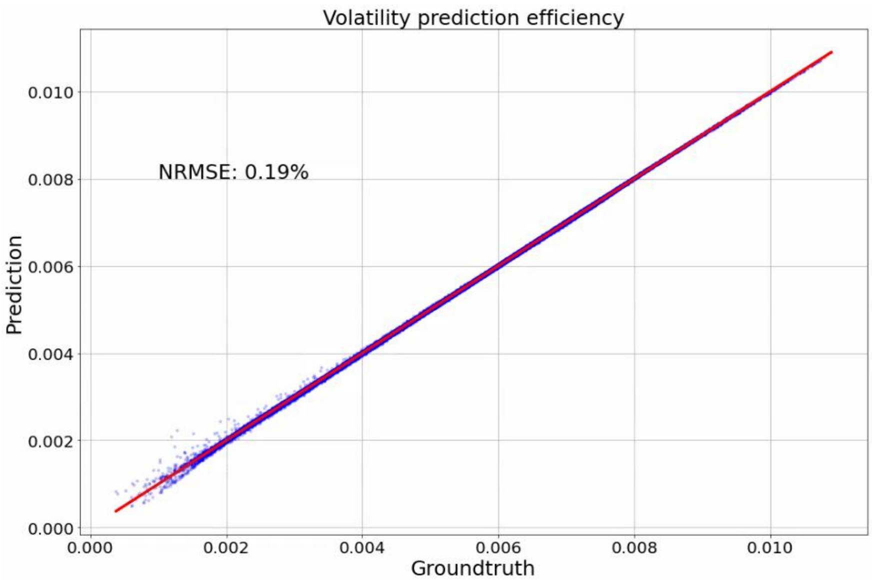

Figure 4: Implied normal volatility prediction efficiency of the decoder network

2.4. Combining the encoder and decoder networks

After we trained the encoder and decoder parts individually, we linked the two parts together into an autoencoder-like structure (see Figure 5.). We use this term for this model since unlike in the case of the classical autoencoders, here the decoder part not only takes the latent parameters as input but also the observable forward rate and *ΔK*.

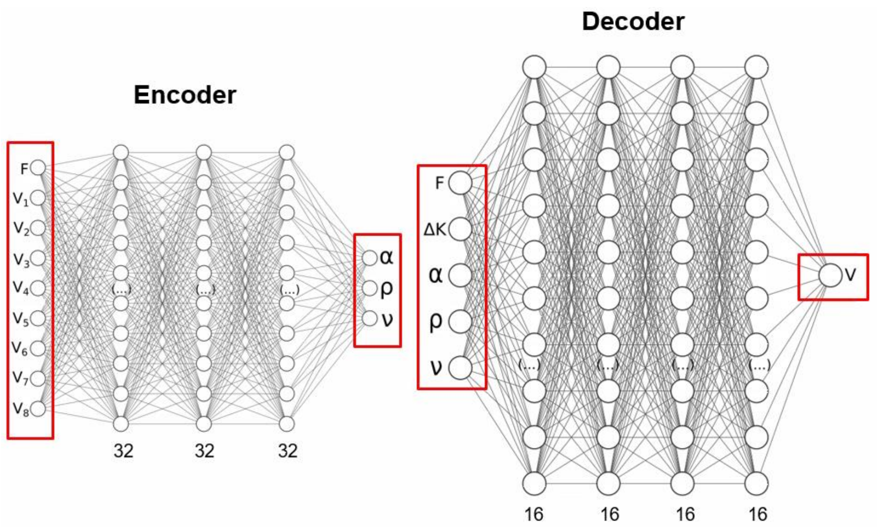

Figure 5: Architecture of the combined model. **Left**: encoder network; **right**: decoder network. The input and output parameters are highlighted with red brackets

We defined the evaluation of this model as follows: 1) The encoder part first maps the 8 volatility values and the forward rate into the 3 dimensional latent space spanned by the α, ρ, υ model parameters; 2) The decoder part takes these latent variables and the forward rate and predicts the volatility values at the 8 strikes, hence reproducing the original volatility skew. This model was further trained on the real data set where we required the network to reproduce well the input volatility smiles while keeping the ρ' and υ' parameters as constant as possible, see the loss function in Eq. 3. The ρ_0 and υ_0 constant parameters were chosen as the mean value of the real ρ and υ parameters regarding to the train set.

$$L = \sum_{i=0}^{n} \left(V_{gt,i} - V_{pred,i}\right)^2 + (\rho_0 - \rho')^2 + (\upsilon_0 - \upsilon')^2 \tag{3}$$

The training set was split chronologically at 2019 September 1 to a training and test set to avoid leakage of future information. We also randomly selected the 20% of the train set for the validation set. Hence, the train, validation and test set contained data of 738, 187 and 297 business days, respectively.

3. Results

3.1. Evaluation on the training set

The combined model was again trained for approx. 3 hours using a batch size of 32. In the following we will refer to the trained model as the „deformation of SABR“ or „deformed SABR“ and the original model with no further training on real data will be called as the „SABR replication“. We can see the evaluation results of the model on the training set in Figure 6.

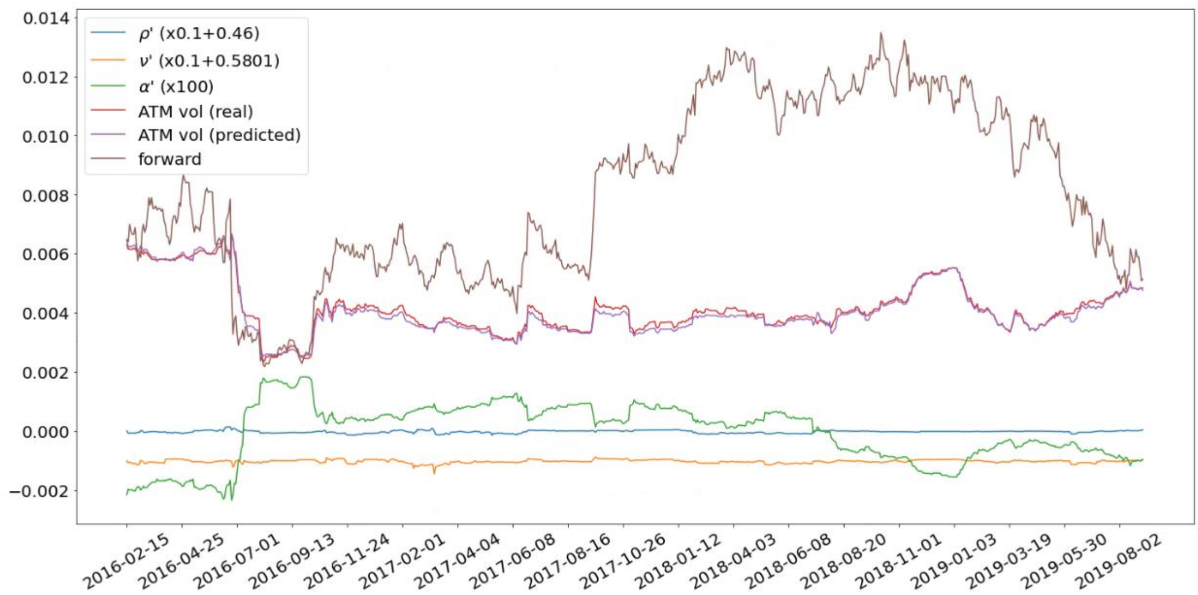

Figure 6: Time evolution of the observable forward rate and ATM volatility as well as the reproduced ATM volatility and the new α', ρ', υ' latent variables regarding to the training set. For illustration purposes the α', ρ', υ' parameters are rescaled which is detailed in the legend

We can see that the ρ' and υ' parameters are extremely stable, their relative standard deviation is 0.0009% and 0.0011%, respectively. The α' parameter has also changed significantly from the original α volatility parameter, where its domain has been extended to the negative regime as well. We have also plotted the time evolution of the real and reconstructed ATM volatilities using this strictly two-factor model, where the reproduction error has been increased to NRMSE=2.5% from the original 0.5% but it is still accaptable. The reproduction error of the volatility skews at all strikes (ΔK) can be seen on Figure 7. The results are compared to the SABR replication.

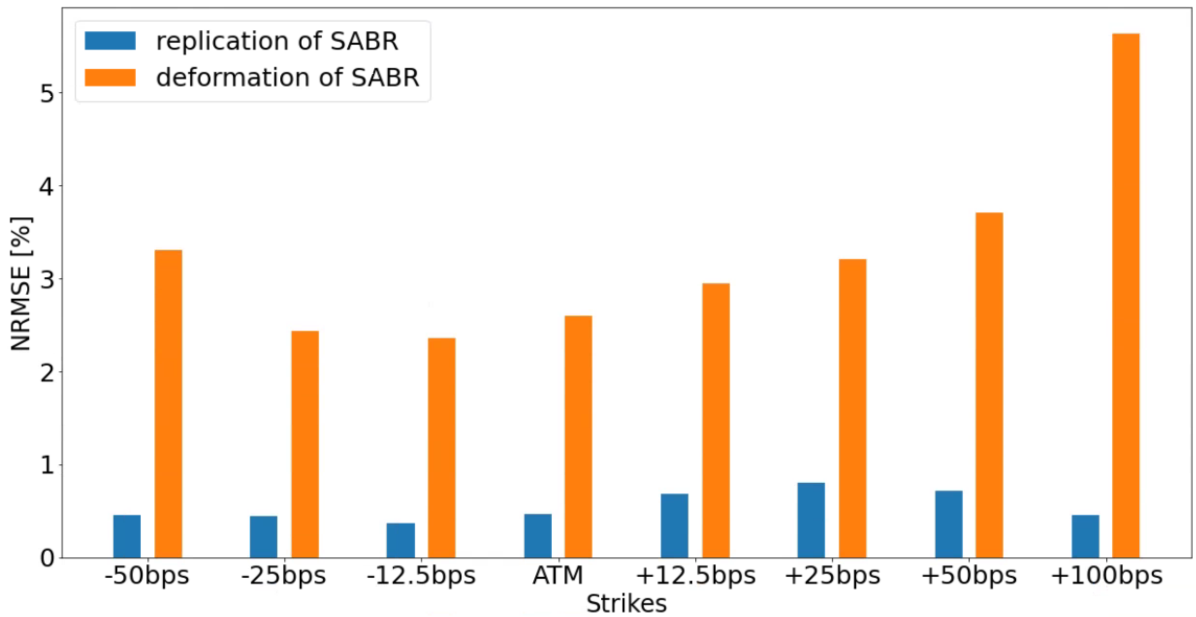

Figure 7: Reproduction error of the normal implied volatility at the different strikes. Blue and orange bars refer to the neural network replication of the SABR model, and its further trained version on real data (deformation of SABR), respectively

We can see that requiring from the network to behave as a two-factor model causes an increase in the reproduction error. The mean NRMSE values are around 3% and 0.5% in case of the deformed SABR and the SABR replication, respectively. We can also notice that the reproduction efficiency is the best near to ATM and gets worse towards the edges in case of the deformed SABR. Let us check the effect of the forward rate (F) and α' on the volatility skews. In Figure 8 we have plotted the skews on a ΔK -F and ΔK - α' grid where we fixed the α' and F parameters at their mean value, respectively.

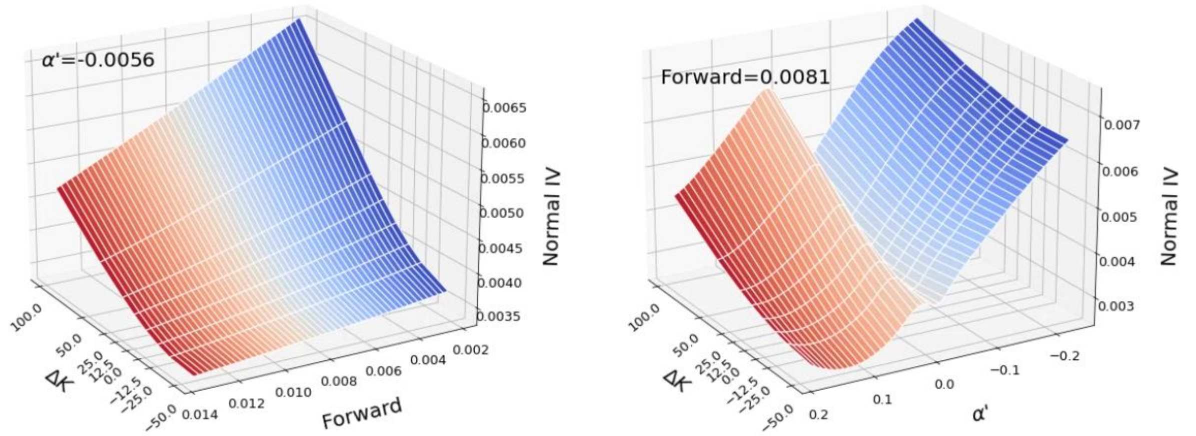

Figure 8: Effect of the forward rate and α' on the volatility skews in the deformation of SABR. **Left**: normal implied volatilities calculated on a ΔK -F grid while fixing α'. **Right**: normal implied volatilities calculated on a ΔK - α' grid while fixing F

We can observe that the forward rate has a simple increasing effect on the skews by decreasing F. Contrarily, α' has a much more complex effect. Moving along the α' axis the skews move vertically and change their slope and convexity at the same time. This behaviour is what we could expect: all the impacts of the original SABR model parameters (α,ρ,υ) on the volatility skews must be compressed into the single latent parameter α'. We can also notice that the sign of α' encodes mainly the skewness: for α'<0 there are shallower volatility skews while for α'>0 we can find much steeper skews.

3.2. Evaluation on the test set

After the training we evaluated the model on the unseen test data (from 2019 September 1 to 2020 October 21)). The time evolution of the latent variables and the observed parameters can be seen in Figure 9.

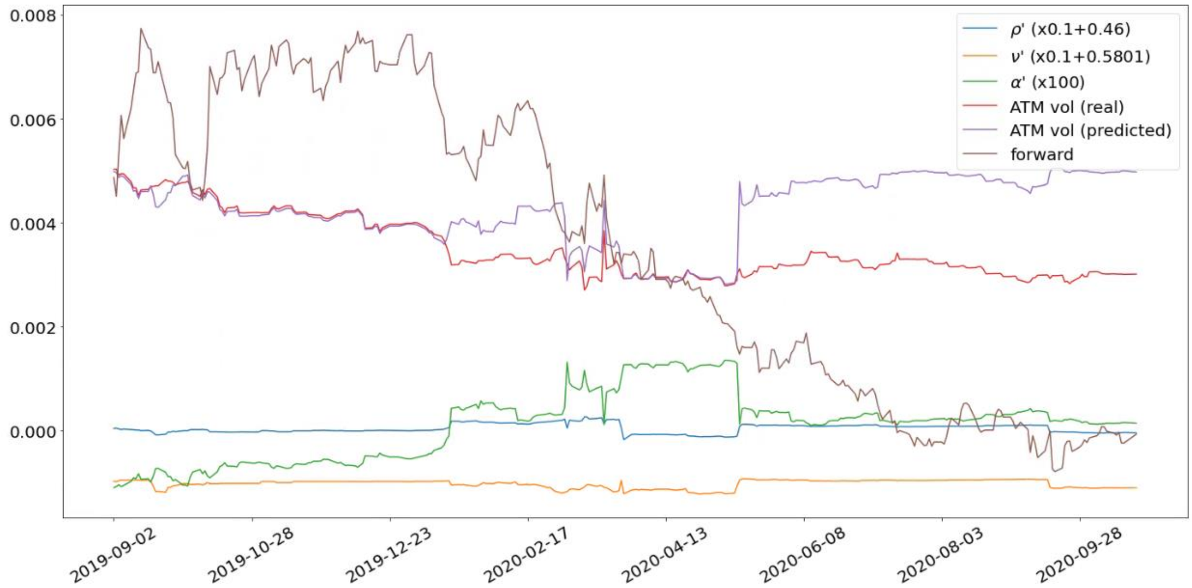

Figure 9: Time evolution of the observable forward rate and ATM volatility as well as the reproduced ATM volatility and the new α', ρ', υ' latent variables regarding to the **test set**. For illustration purposes the α', ρ', υ' parameters are rescaled which is detailed in the legend

The first thing that we can observe is that on the unseen data the ρ' and ν' latent variables still remained almost constant. The other important note is that the ATM volatility reconstructions are still good (NRMSE=6.9%) until a well-defined date. From 2020 January 15 - which is close to the beginning of the pandemic crisis - the reproductions start to deviate significantly from the groundtruth values. In Figure 10 we plotted the absolute reproduction error – namely the root mean squared error (RMSE) - averaged at all strikes as well as calculated for each strike individually.

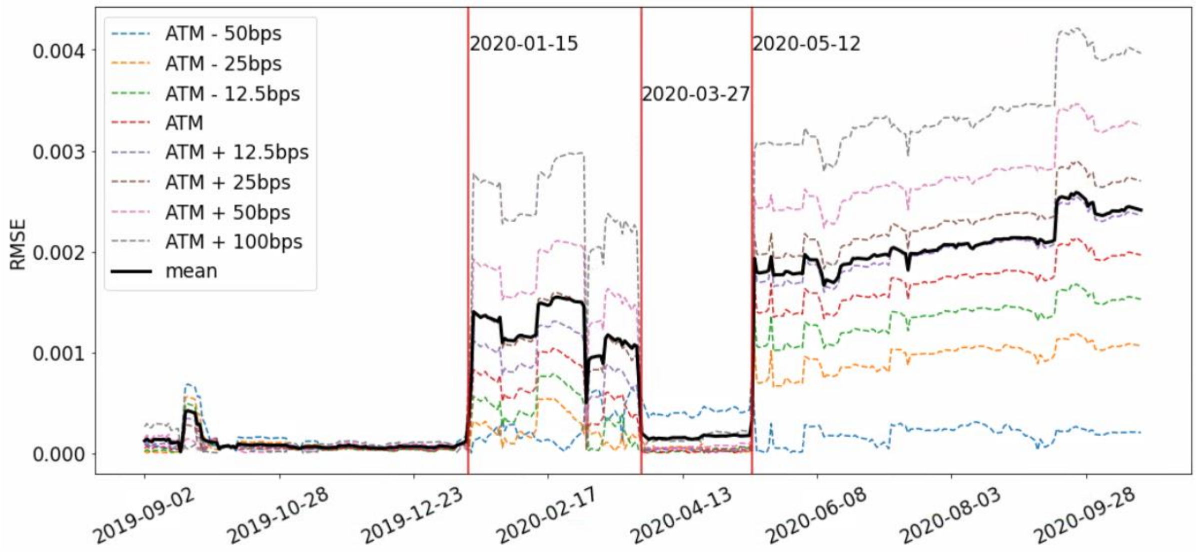

Figure 10: Time evolution of the root mean squared error of the reproduction. Colored, dotted lines refer to the difference measured at the different strikes individually, whereas black continues line represents the mean deviation

We can see that at all strikes the reconstructed volatility values diverge from the observations between 2020 January 15 and 2020 March 27 as well as from 2020 May 12 to the end of the test set. This means that the model can well describe the dynamics of the swaption skews by using only two stochastic variables if the market is "normal", and produces significant differences when the market goes into an unusual environment. A neural network has typically a good interpolation ability and it is bad at extrapolating. If it gets an unseen sample which is „far outside" of the manifold of the train set, its prediction efficiency will decrease. In the following we have calculated the Euclidean distance for each test sample to all other train samples in the 9-dimensional space spanned by the $V_{1,...,8}$ and F observed parameters and we extracted the minimum distance for each test sample (see Figure 11.). In this way we measured how unusual a specific skew for the network is.

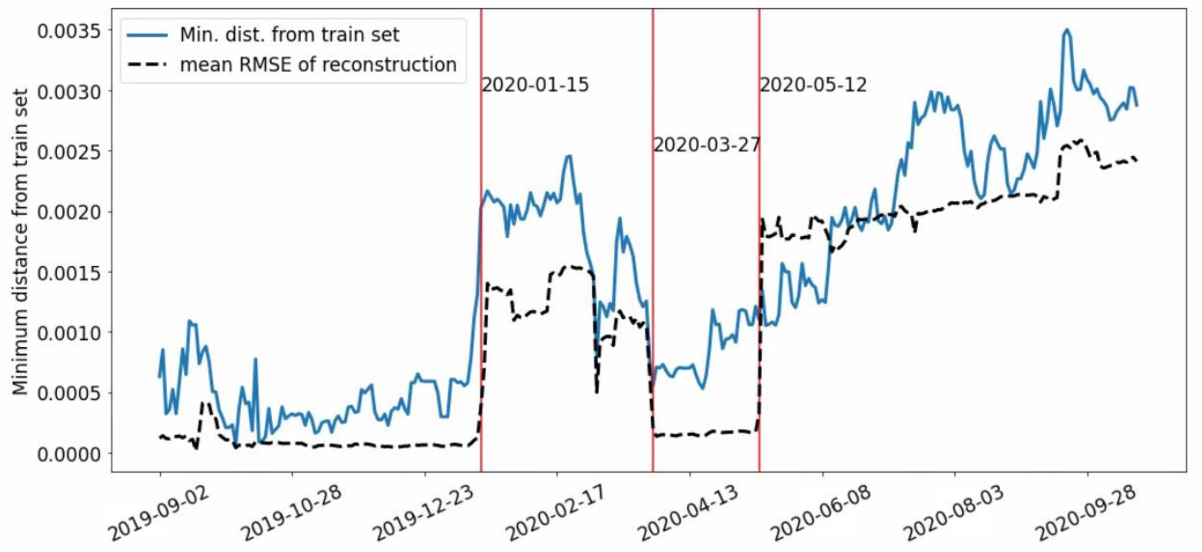

Figure 11: Relation between the distance of test samples from the train set and the reproduction error. Blue continuous line represents the minimum of all distances from that specific test sample to all train samples, while the black dotted line refers to the RMSE of the reproduction

We can see that there is a strong correlation between the minimum distance of the test samples to the train set and the reconstruction efficiency of the network. This means that by decreasing the number of degree of freedom of the model causes a high sensitivity to unusual volatility skew – forward rate pairs. This feature could be extremely useful in regime change detection since the model can detect immediately if the market moves from the usual space.

4. Discussion

According to these results we are now closer to answer the original question: does the incorrect model choice or the complexity of the market cause the large fluctuation of the SABR parameters over time? It is now clear that for this specific swaption time series we are able to find a two-factor model which can follow the dynamics of swaption skews not only in the training set but even in the test set for several months. The price of keeping the original ρ and υ constant is not surprisingly the lower accuracy. However, the reconstruction performance is still acceptable: NRMSE~3-4% for training set and NRMSE~6-7% for the stable part of the test set. This shows that the investigation of the true number of degrees of freedom is very important before creating too complex pricing models. Using neural networks we can fix the number of latent parameters and find the model that fits the best to the observed data. On the other hand - as in the case of all deep learning studies - the model efficiency will strongly depend on the choice of training set, as we saw in our study by the model evaluation on the test set. However, if we intentionally use „normal" market data as the training set then the model will be sensitive to the unusual market movements. The development of such market regime change detectors can help in finding optimal trading strategies and is an active research area in computational finance, see (Bucci & Ciciretti, 2021).

5. Conclusions

In our study we have investigated a well-known problem that the SABR supposed to be non-stochastic model parameters (ρ and ν) are in practice stochastic. Our goal was to find an alternative option pricing model which is a strictly of two factors having the forward rate and a new α' variable as stochastic parameters. For this study we used neural networks which have high flexibility to adjust their weights to real data while using an arbitrarily customizable loss function. Due to the relatively small number of real market data we first trained a model calibrator (encoder) and a model replicator (decoder) network on synthetic data, where we achieved the accuracy of previous studies and demonstrated again the excellent interpolation capabilities of neural networks. We have combined the two networks together into an autoencoder-like structure which is a novel approach and is much more interpretable than starting with a usual (variational) autoencoder, see (Bergeron et al., 2021). In our method we know the exact meaning of the original latent space, and we let the neural network initially replicating SABR to adapt its weights and the latent space to the real observations.

We have shown that one can find such an option pricing model but with the price of somewhat lower accuracy. We have also demonstrated that this two-factor model is able to follow the dynamics of the previously unseen swaption market over several months. However, in extreme environments the model performance suddenly decays, which is a sensitive indicator of unusual market conditions.

Acknowledgment

Prepared with the professional support of the Doctoral Student Scholarship Program of the Co-operative Doctoral Program of the Ministry of Innovation and Technology financed from the National Research, Development and Innovation Fund.

References

Bayer, C., Horvath, B., Muguruza, A., Stemper, B., and Tomas, M. (2019). On deep calibration of (rough) stochastic volatility models. arXiv e-prints, page arXiv:1908.08806.

Bergeron, M., Fung, N., Hull, J., Poulos, Z. (2021). "Variational Autoencoders: A Hands-Off Approach to Volatility," Available: https://arxiv.org/pdf/2102.03945.pdf

Black, F. and Scholes, M. (1973). "The Pricing of Options and Corporate Liabilities," *Journal of Political Economy*, vol. 81, pp. 637-654. https://doi.org/10.1086/260062

Bucci, A. and Ciciretti V. (2021). "Market Regime Detection via Realized Covariances: A Comparison between Unsupervised Learning and Nonlinear Models, " Available: https://arxiv.org/abs/2104.03667

Gaspar, R. M., Lopes, S. D., and Sequeira, B. (2020). Neural network pricing of american put options. Risks, 8(3). https://doi.org/10.3390/risks8030073

Hagan, P., Kumar, D., Lesniewski, A., and Woodward, D. (2002). "Managing Smile Risk," *Wilmott Magazine*, vol. 1, pp. 84-108.

Horvath, B., Muguruza, A., and Tomas, M. (2019). Deep Learning Volatility. arXiv e-prints, page arXiv:1901.09647. https://doi.org/10.2139/ssrn.3322085

Liu, S., Oosterlee, C. W., and Bohte, S. M. (2019). Pricing options and computing implied volatilities using neural networks. Risks, 7(1). https://doi.org/10.3390/risks7010016

Pagnottoni, P. (2019). Neural network models for bitcoin option pricing. Frontiers in Artificial Intelligence, 2. https://doi.org/10.3389/frai.2019.00005

Webber, C., Becker R. (2014). “A survey and implementation of some calibration algorithms for the SABR and Heston models,“ Available: https://open.uct.ac.za/bitstream/handle/11427/8518/thesis_com_2014_com_webber_c.pdf?sequence=1

West, G (2005). Calibration of the SABR model in illiquid markets. Applied Mathematical Finance, 12(4): 371-385. https://doi.org/10.1080/13504860500148672

Zhang, M. and Fabozzi, F. J. (2016). “On the Estimation of the SABR Model's Beta Parameter: The Role of Hedging in Determining the Beta Parameter,” *Journal of Derivatives*, vol. 24, pp. 48-57. https://doi.org/10.3905/jod.2016.24.1.048