



Applied Machine Learning for Stochastic Local Volatility Calibration

Jürgen Hakala*

Quantitative Modeling Department, Leonteq Securities AG, Zurich, Switzerland

Stochastic volatility models are a popular choice to price and risk-manage financial derivatives on equity and foreign exchange. For the calibration of stochastic local volatility models a crucial step is the estimation of the expected variance conditional on the realized spot. The spot is given by the model dynamics. Here we suggest to use methods from machine learning to improve the estimation process. We show examples from foreign exchange.

Keywords: radial basis functions, machine learning, local stochastic volatility, derivatives pricing, finance

OPEN ACCESS

Edited by:

Peter Schwendner,
Zurich University of Applied Sciences,
Switzerland

Reviewed by:

Natalie Packham,
Hochschule für Wirtschaft und Recht
Berlin, Germany
Francesco Caravelli,
Los Alamos National Laboratory
(DOE), United States

*Correspondence:

Jürgen Hakala
juergen.hakala@leonteq.com

Specialty section:

This article was submitted to
Artificial Intelligence in Finance,
a section of the journal
Frontiers in Artificial Intelligence

Received: 17 January 2019

Accepted: 25 April 2019

Published: 17 May 2019

Citation:

Hakala J (2019) Applied Machine
Learning for Stochastic Local Volatility
Calibration. *Front. Artif. Intell.* 2:4.
doi: 10.3389/frai.2019.00004

1. INTRODUCTION

For derivatives pricing a major breakthrough was achieved with the risk-neutral valuation principle (Black and Scholes, 1973). Initially the models assumed a deterministic, state-independent volatility of the underlying price process. For many classes of underlying this assumption is too restrictive as it does not allow for an implied volatility that depends on strike as it is observed in the market, at least since the Black Friday in 1987, see for a review and attempted explanation (Benzoni et al., 2011).

Hence the most natural extension of the existing models was to postulate either a state-dependent volatility often duped as local volatility (Derman and Kani, 1994; Dupire, 1994) or to postulate an additional process for the volatility (e.g., Hull and White, 1987; Heston, 1993) which are labeled as stochastic volatility models.

Looking at the properties of these two model classes it was found (Hagan et al., 2002) that local volatility is postulating a dynamics which is not found in real markets. In foreign exchange options markets stochastic volatility models tend to exaggerate the effect of volatility convexity and at the same time these models are unable to match the short-dated volatility smile observed in market-prices. As a practical workaround, models that mix the local volatility and stochastic volatility were developed (Said, 1999; Blacher, 2001). It was observed that the calibration of SLV models is a hard problem which requires either a specific parametrization to derive fast pricing of vanilla options or quite time-consuming numerical optimization procedures (Guyon and Henry-Labordere, 2011). See as well Homescu (2014) for a great summary and best practice of local stochastic volatility models.

A shortcut to derive manageable calibration times was developed by Guyon and Henry-Labordere (2011) and Van der Stoep et al. (2014) using a Monte Carlo procedure to derive the required estimation of the conditional variance. In this paper we suggest to use methods from machine learning, in particular radial basis functions and variations thereof to derive fast and efficient estimators.

2. LOCAL STOCHASTIC VOLATILITY CALIBRATION

The LSV model in general is of the form:

$$\begin{aligned} dS_t &= \mu(t)S_t dt + \sigma(S_t, t)f(V_t)S_t dW_t \\ dV_t &= \mu_V(V_t)dt + \xi\chi(V_t)dX_t \\ \langle dW_t, dX_t \rangle &= \rho dt \end{aligned}$$

with spot S_t , variance V_t , drift μ , (state-dependent) drift for the variance μ_V , vol of variance $\xi\chi(V_t)$, and correlation ρ . The LSV calibration is the process to determine the leverage function σ given the local volatility function σ_{Dupire}^2 and all the other parameters of the model. There is a fundamental relationship of the leverage function and the local volatility function (Dupire, 1996) where the expectation $E^{P(S_t, V_t, \sigma)}$ of the conditional variance V_t is taken with respect to the risk-neutral measure induced by the model. The notation indicates that $P(S_t, V_t, \sigma)$ is the joint probability of spot process S_t , variance process V_t and the solution for σ which depends on the probability distribution of (S_t, V_t) .

$$\sigma_{Dupire}^2(S_t, t) = E^{P(S_t, V_t, \sigma)}(V_t | S_t = S_t) \sigma^2(S_t, t) \quad (1)$$

Plugging the solution into the model equation makes this a McKean SDE where the expectation depends on the probability of the process itself.

To solve this equation Monte Carlo simulation can be used. The equations are discretized and the forward propagation of the spot S_t and variance V_t is interleaved with the estimation of conditional expectation using the realized paths of S_t and V_t . Contrary to standard Monte Carlo where all paths develop independently we need to bring all simulated paths to the estimation procedure. Using Euler discretization

$$\begin{aligned} \Delta \ln(S_t) &= \mu(t)\Delta t - \frac{1}{2}\sigma^2(S_t, t)f^2(V_t)\Delta t \\ &\quad + \sigma(S_t, t)f(V_t) \left(\sqrt{1 - \rho^2} \Delta W + \rho \Delta X \right) \\ \Delta V_t &= \mu_V(V_t)\Delta t + \xi\chi(V_t)\Delta X \end{aligned}$$

with $\Delta W, \Delta X$ independent increments. The estimation of the conditional expectation can be seen as finding the function $R(S) = E(V_t | S_t = S)$ based on the samples as observed pairs

$$(S_t^1, V_t^1), \dots, (S_t^n, V_t^n)$$

where the spot S_t^i and variance V_t^i are the time t realizations of spot and variance on path i . Originally it was proposed to estimate the function R using kernel regression (Guyon and Henry-Labordere, 2011):

$$R((S^1, V^1), \dots, (S^N, V^N))(S) = \frac{\sum_{i=1}^N V_i K_h(S - S_i)}{\sum_{i=1}^N K_h(S - S_i)} \quad (2)$$

with Kernel functions K_h , where we dropped the t index as it is clear from the context. Alternatively (Van der Stoep et al., 2014) proposed to use binning techniques or sets of polynomials.

Subsequently we will evaluate alternative regression techniques to estimate the conditional expectation based on the realized paths. This can be rephrased as a supervised learning problem where each path is a (noisy) example.

3. REGRESSION AS A SUPERVISED LEARNING PROBLEM

The task to find a relationship between some input variables and an output from examples is one of the problems tackled by machine learning and is well studied as supervised learning. There are many classes of supervised learning algorithms and setups and we would like to demonstrate guidelines to which specific choices are suitable for the problem at hand. The basic problem is, given a set of examples x_i, y_i to find a function $f(x)$ such that an error functional is minimized. The task is to find a function such that there is low error on unseen examples, this is called generalization. There is a balance to strike between the error on the examples used for training and the error on the validation set of examples which are examples not used during training, for a fundamental analysis of the learning theory and the relation between capacity and the generalization (see e.g., Vapnik, 2013), in particular chapter 4.

3.1. Kernel Regression

The approach taken in Guyon and Henry-Labordere (2011) as stated above is Nadarajan-Watson kernel regression which is one of the so called non-parametric methods. The method is identical to Equation (2). The estimator is given as:

$$R((x_1, y_1), \dots, (x_N, y_N))(X) = \frac{\sum_{i=1}^N y_i K_h(x - x_i)}{\sum_{i=1}^N K_h(x - x_i)}$$

In this approach a Kernel function $K(x)$ is used, which satisfies:

$$\begin{aligned} K(x) &\geq 0 \\ K(x) &= K(-x) \\ \int_{-\infty}^{\infty} K(x) dx &= 1 \\ K_h(x) &= \frac{1}{h} K\left(\frac{x}{h}\right) \end{aligned}$$

There is a variety of Kernel functions well studied in the literature (Härdle, 1990):

- Gaussian $\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$
- Quartic $\frac{15}{16}(1 - x^2)^2$
- Epanechnikov $\frac{3}{4}(1 - x^2)$
- Sigmoid $\frac{2}{\pi} \frac{1}{e^x + e^{-x}}$

Often the Kernel function used is Gaussian hence the support of the function is infinite or it will be the Epanechnikov Kernel which has bounded support.

The crucial choice is the bandwidth of the Kernel functions. There is a rule-of-thumb derived from normal distribution

assumptions, (Silverman, 1986):

$$h = \left(\frac{4\sigma^5}{3n} \right)^{\frac{1}{5}}$$

for the standard deviation σ of the data and n data points.

Alternatively cross validation, particular “leave-one-out cross validation” can be used to determine an optimal Kernel width. Cross validation is quite costly computationally and hence can only be used to cross check *ad-hoc* choices.

Local Linear Kernel Regression is a variation of Kernel regression which employs local linear terms and which is given by the solution of

$$R((x_1, y_1), \dots, (x_N, y_N))(x) = \min_{\alpha, \beta} \sum_{i=1}^N (y_i - \alpha - (x - x_i)\beta)^2 K_h(x - x_i)$$

The minimum is found by solving a 2×2 linear system .

In general the Kernel approaches suffers from some systematic shortcomings, mainly the fact that all examples are used, no compression happens, secondly a bias is introduced close to the boundary and the difficult choice of suitable bandwidth, where practically sound theoretical methods as cross validation cannot be used for computation time reasons.

3.2. Radial Basis Functions

Radial Basis Functions (RBF) and Partition of Unity Radial Basis Functions (PURBF) respectively take the form

$$RBF(x) = \sum_{i=1}^C w_i K_{h_i}(x - c_i)$$

$$PURBF(x) = \frac{\sum_{i=1}^C w_i K_{h_i}(x - c_i)}{\sum_{i=1}^C K_{h_i}(x - c_i)}$$

PURBF are quite similar in functional form to Kernel regression. The main difference is that the number of basis functions is much smaller than the number of examples. It was proven that RBF and PURBF are universal function approximators (Hakala et al., 1994) which makes them suitable to approximate our estimation problem. If the L_2 norm is used the weights are optimized by solving the normal equation

$$LS = \frac{1}{2N} \sum_{i=1}^N (y_i - RBF(x_i))^2.$$

The solution s given by the weights w_i , which satisfy

$$w_i = (A^T A)_{ik}^{-1} A_{kj} y_j$$

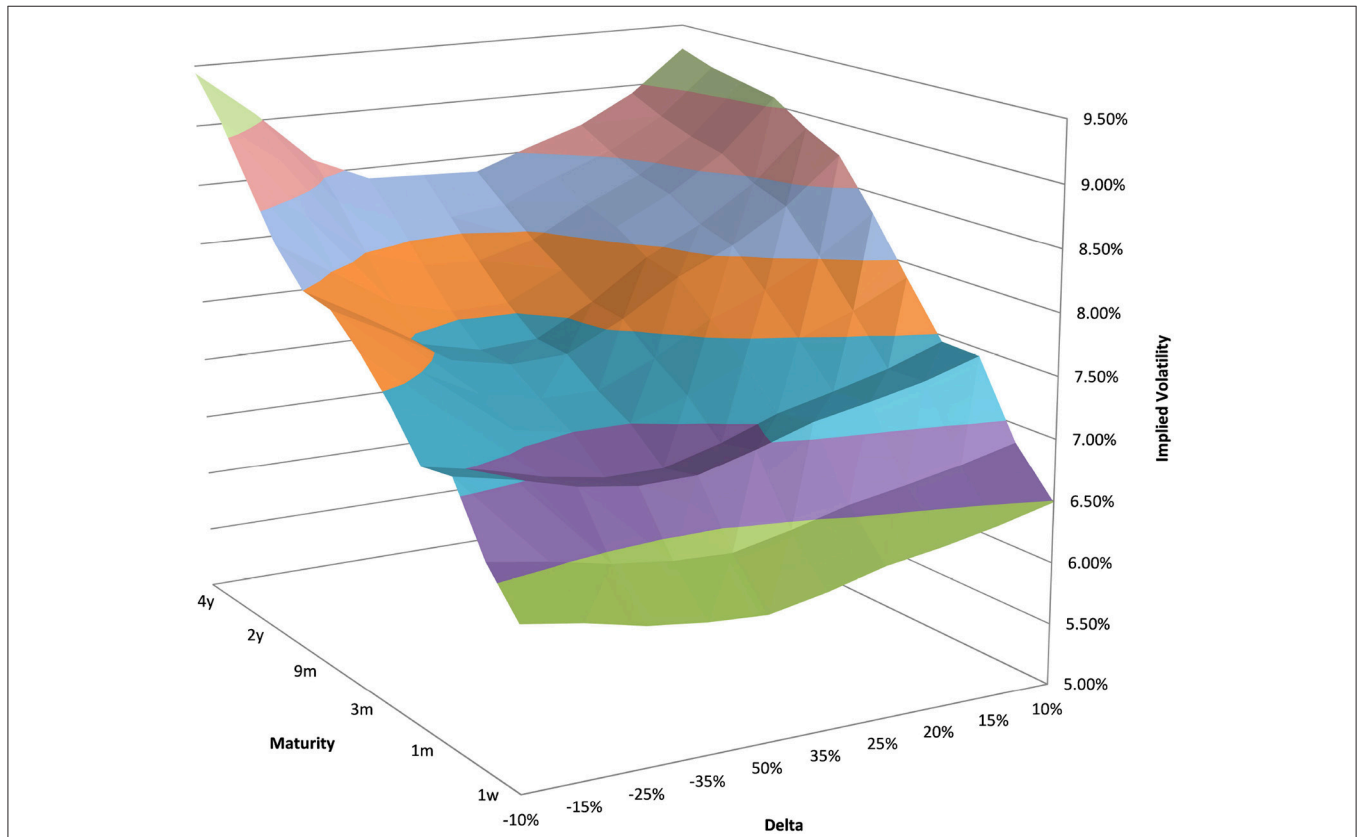
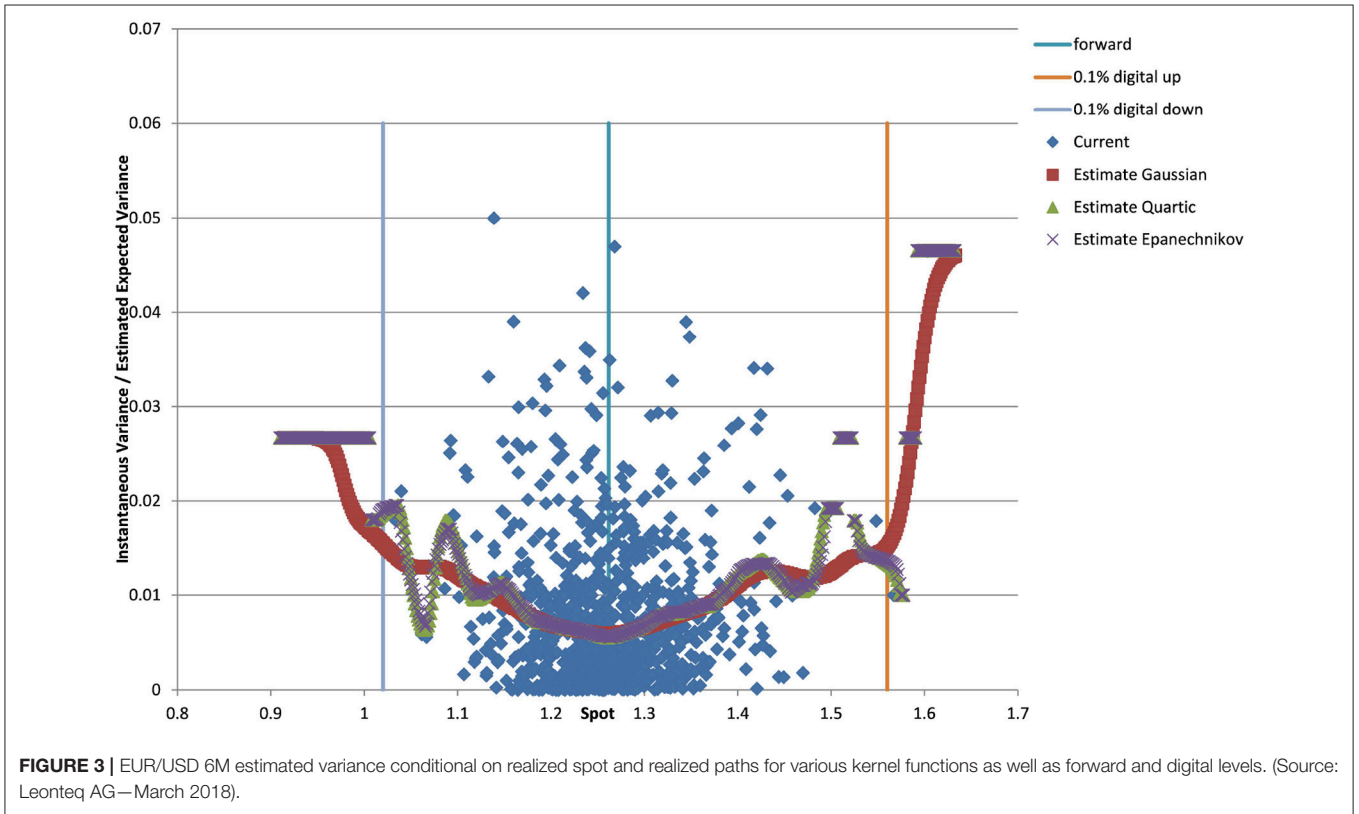
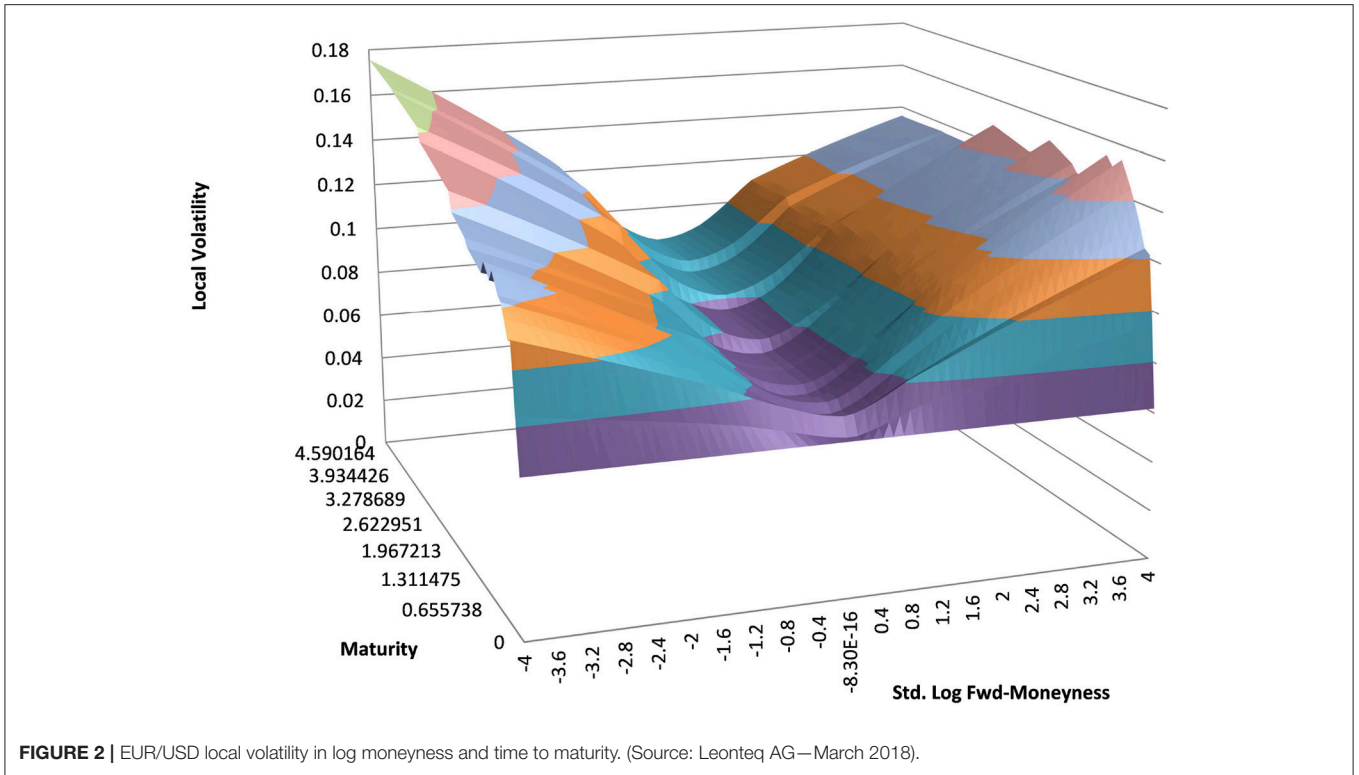
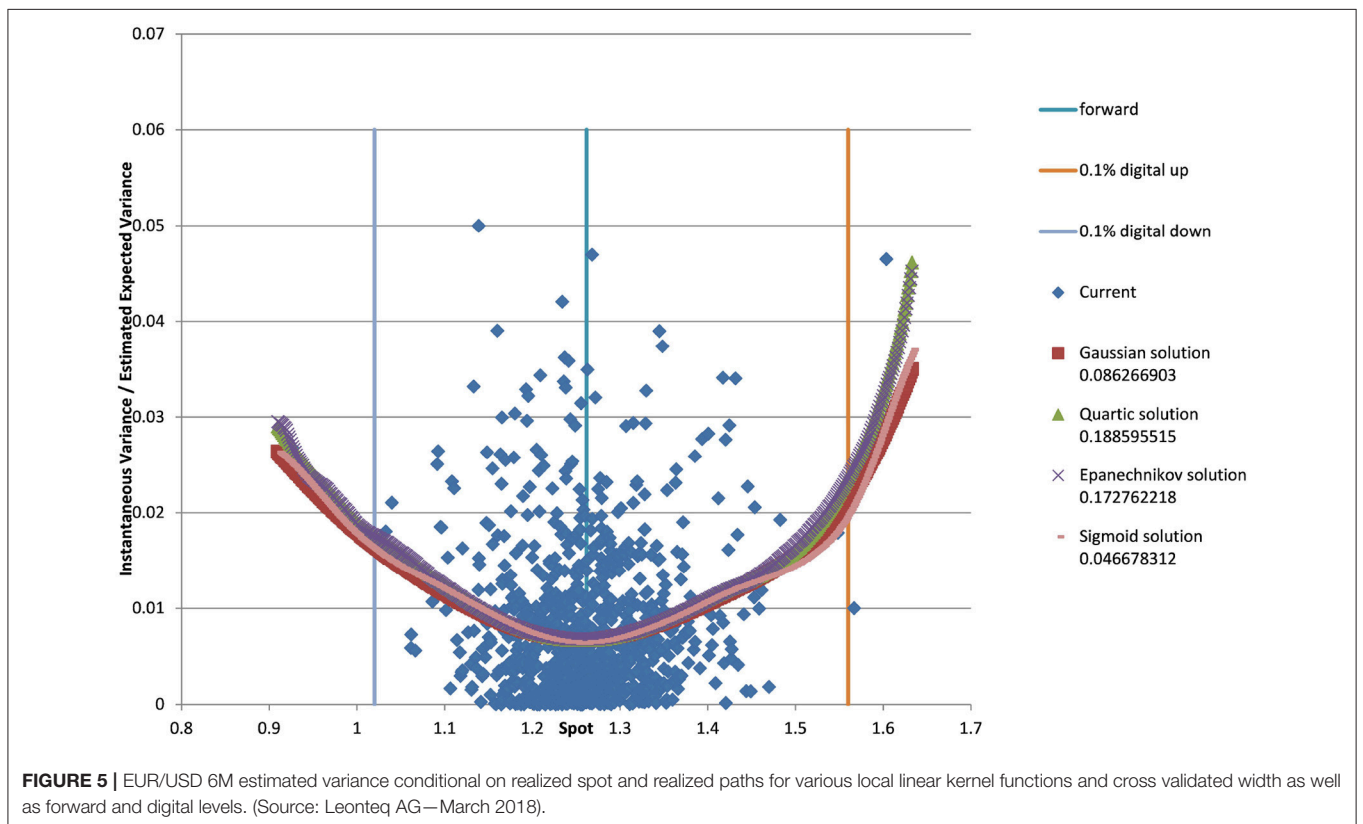
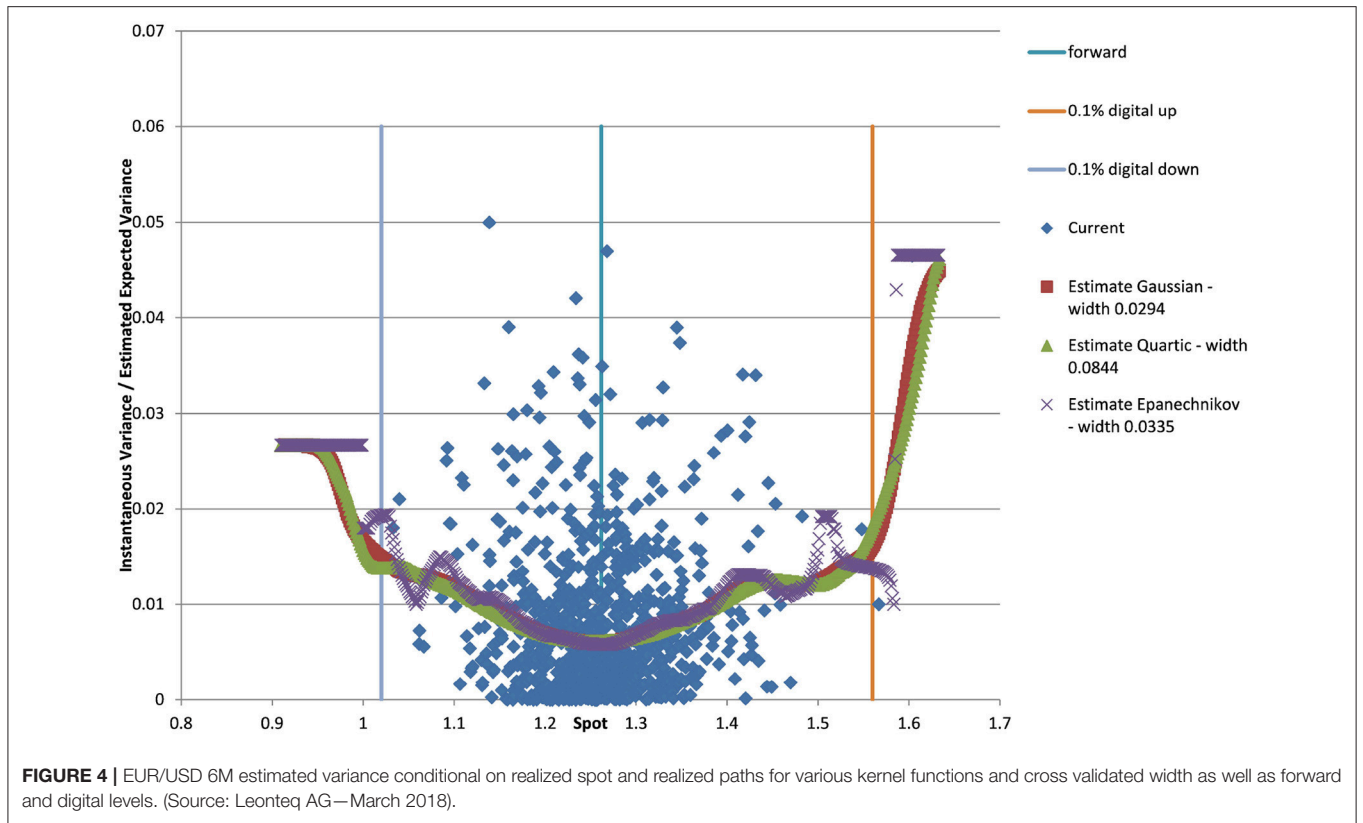


FIGURE 1 | EUR/USD implied volatility in log moneyness and time to maturity. (Source: Leonteq AG—March 2018).





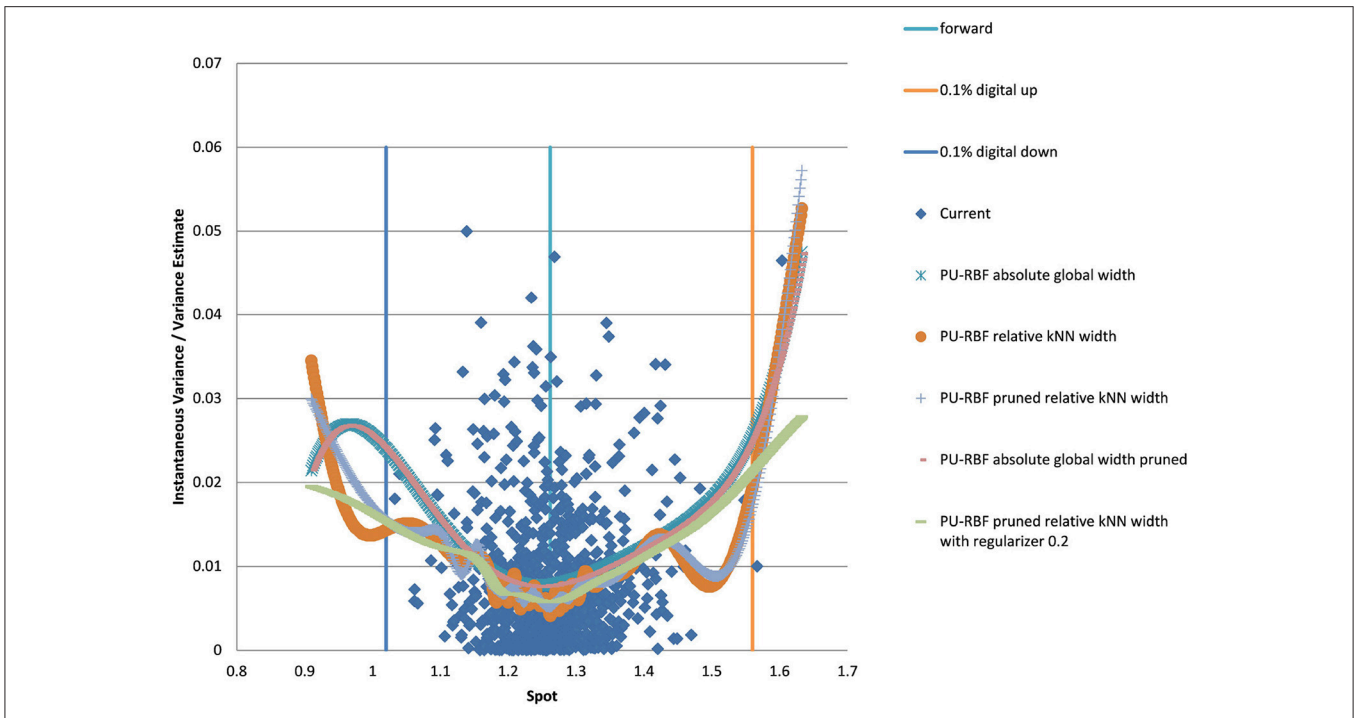


FIGURE 6 | EUR/USD 6M estimated variance conditional on realized spot and realized paths using PURBF as well as forward and digital levels. (Source: Leonteq AG—March 2018).

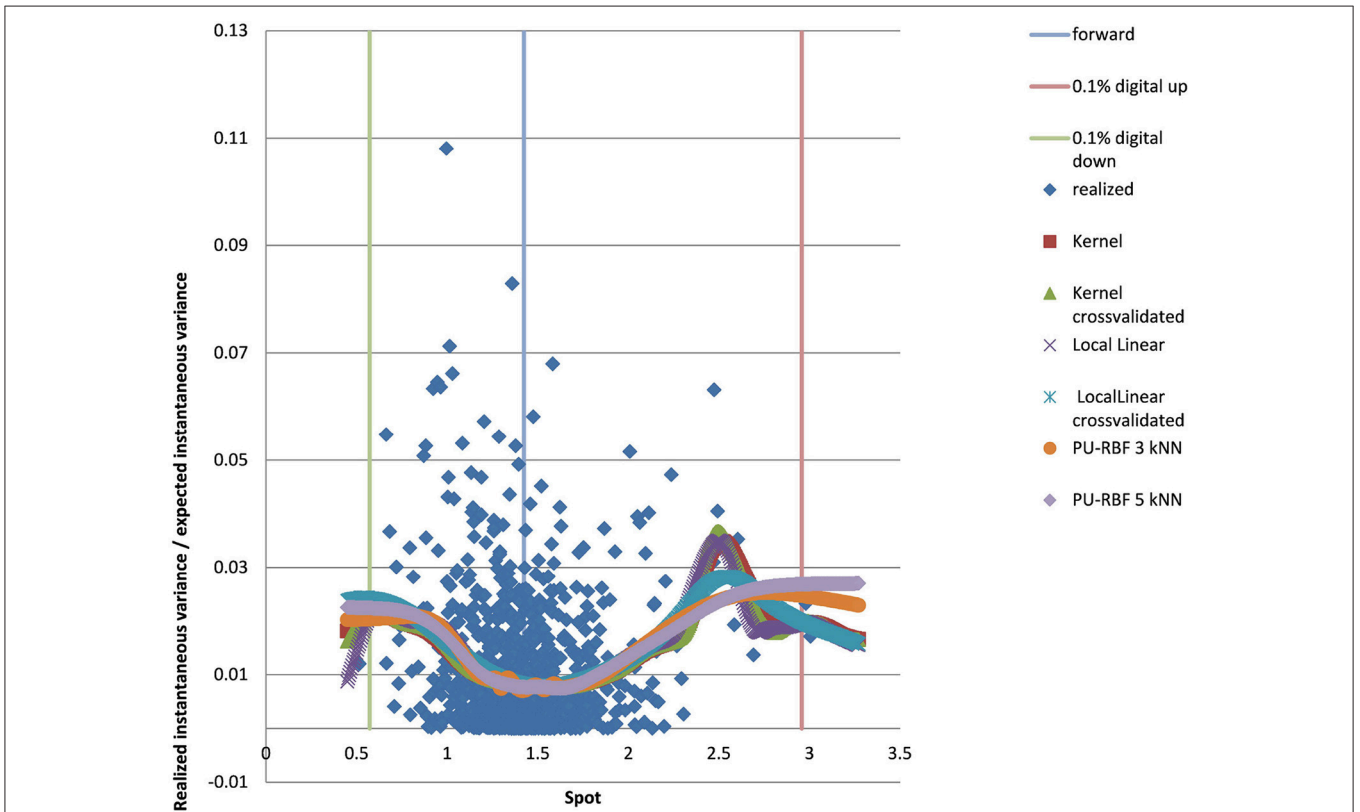


FIGURE 7 | EUR/USD 5Y estimated variance conditional on realized spot and realized paths using kernel regression, local linear kernel regression, and PURBF as well as forward and digital levels. (Source: Leonteq AG—March 2018).

with the matrix A_{ij} given as

$$A_{ij} = K_{h_j}(x_i - c_j). \tag{3}$$

The remaining parameters are determined heuristically:

- centers c_i - in 1D we chose $\min(x_i), \max(x_i)$, and a random subset of the remaining x_i . Centers which are too close (in their local width) are pruned—the criterion for unit j is $\min_i \left(\frac{|c_i - c_j|}{h_j} \right) \leq \Theta$ with a suitable global pruning constant Θ . There are resource allocating approaches as well (e.g., Fritzsche, 1994) which seem less suitable here.
- width h_i - are chosen individually as a multiple of the k nearest neighbors or *ad-hoc*.

3.2.1. Regularization

Often the solution of the normal equation will be ill-conditioned. To counteract the bad conditioning of the problem and to get a better generalization we will use a regularizer on the L_2 norm of the weights (e.g., Goodfellow et al., 2016, Chapter 7.1).

$$LSR = \frac{1}{2N} \sum_{i=1}^N (y_i - RBF(x_i))^2 + \lambda \sum_{j=1}^C w_j^2$$

The corresponding solution is given as

$$w_i = (A^T A - \lambda \text{id})_{ij}^{-1} A_{jk} y_k$$

with identity matrix id . The same solution applies for the PURBF function instead of the RBF one.

3.3. Computational Efficiency

For standard Kernel Regression computational effort is mainly due to sorting $O(n \log(n))$ of the spot observations to enable an efficient lookup of relevant spot observations during the retrieval phase. Optimal determination of width (cross-validation) requires the evolution of all kernels at all points several times which is very costly compared to the lookup. Local linear Kernel Regression requires an additional inversion of a 2×2 matrix which is negligible. For RBF and PURBF the solution to a small linear system is required. In particular the size is much smaller than the number of samples. Sorted examples can be used to optimize the training as the required matrix is determined by sums over the samples. Width and pruning computations require local computation of the order of the number of kernel functions. Overall the computational effort for RBF/PURBF is comparable and might be smaller in the retrieval phase than for Kernel Regression itself.

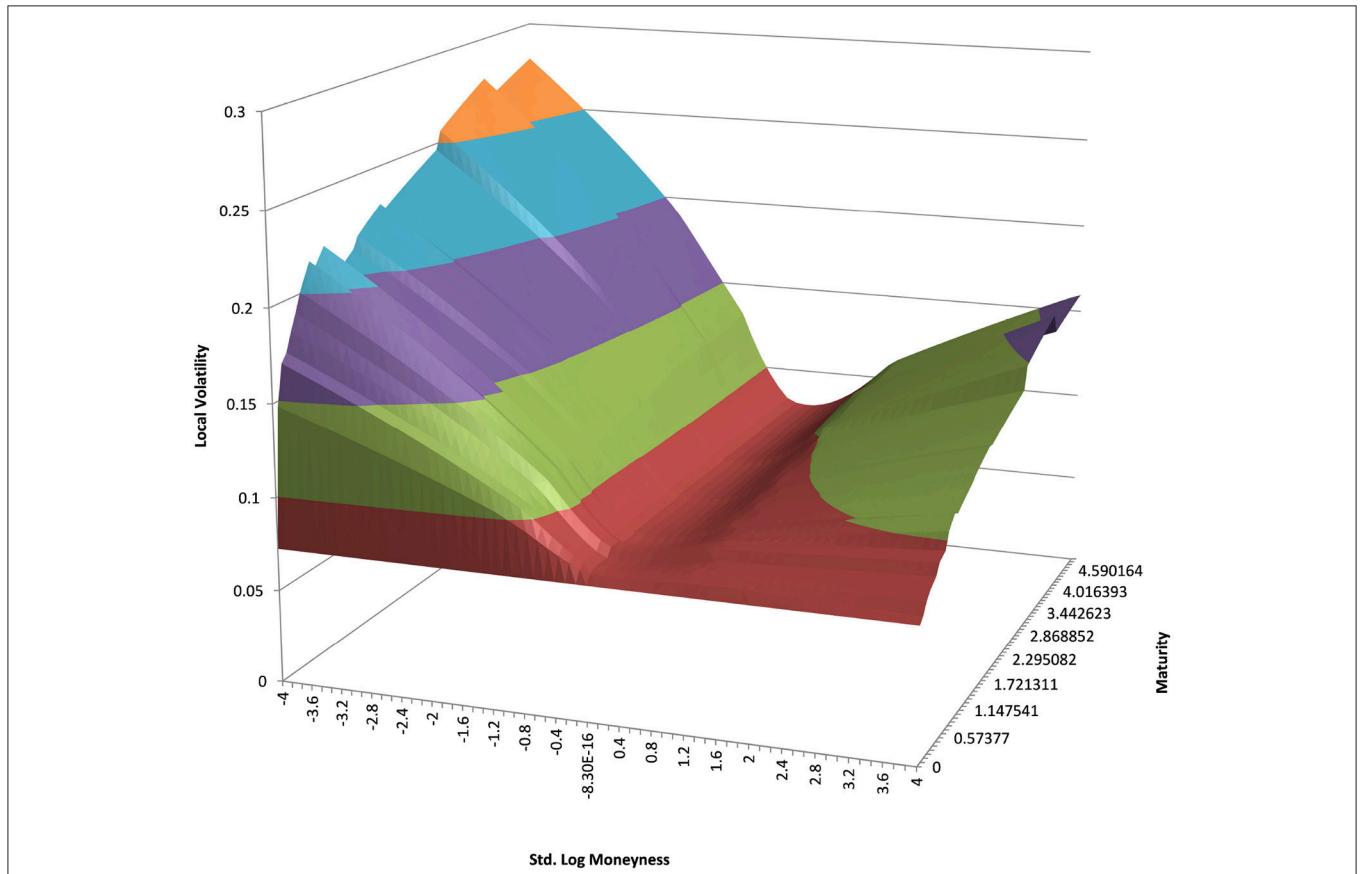


FIGURE 8 | USD/JPY local volatility in log moneyess and time to maturity. (Source: Leonteq AG—March 2018).

3.4. Alternative Architectures

In the last couple of years popularity of multilayer perceptron (MLP) and deep versions thereof grew enormously. For our application we rule out these architectures as the training is much more involved in the MLP case with a many remaining questions about a suitable number of hidden units, number of layers, type of activation functions. We could envision to use a pretrained MLP to get the solution without training. We postpone this approach for potential future use.

4. APPLICATION TO LSV MODEL IN FOREIGN EXCHANGE

The model we will study is of Heston type

$$\begin{aligned}
 dS_t &= \mu(t)S_t dt + \sigma(S_t, t)\sqrt{V_t}S_t dW_t \\
 dV_t &= \kappa(\bar{V} - V_t)dt + \xi\sqrt{V_t}dX_t \\
 \langle dW_t, dV_t \rangle &= \rho dt
 \end{aligned}$$

with mean reversion speed κ , mean reversion level \bar{V} .

The advantage is that we have a semi-closed form solution for vanilla call- and put options in the Heston model without the leverage function hence the first step is to calibrate the Heston model and then apply a scaling to the vol of vol parameter to reduce the SV impact and to let the local volatility compensate to match the vanilla option market. In this study we will use a volatility mixing of 66% which means that we scale the vol of variance by this factor before calibrating of the leverage function.

To compare the performance of the various regression algorithms on this model we will show for a specific slice the realized spot/variance and the corresponding results of the regression functions.

4.1. Example EUR/USD 6M

You can see the volatility surface in **Figure 1** and the corresponding local volatility surface in **Figure 2**. The snapshot of data, including spot, volatility, and interest rates was taken in March 2018. We show the results of different kernel estimators, using Silverman’s rule of thumb for the width, including the samples indicated as *Current*, as well as the forward and the level of a 0.1% digital on the upside and downside **Figure 3**.

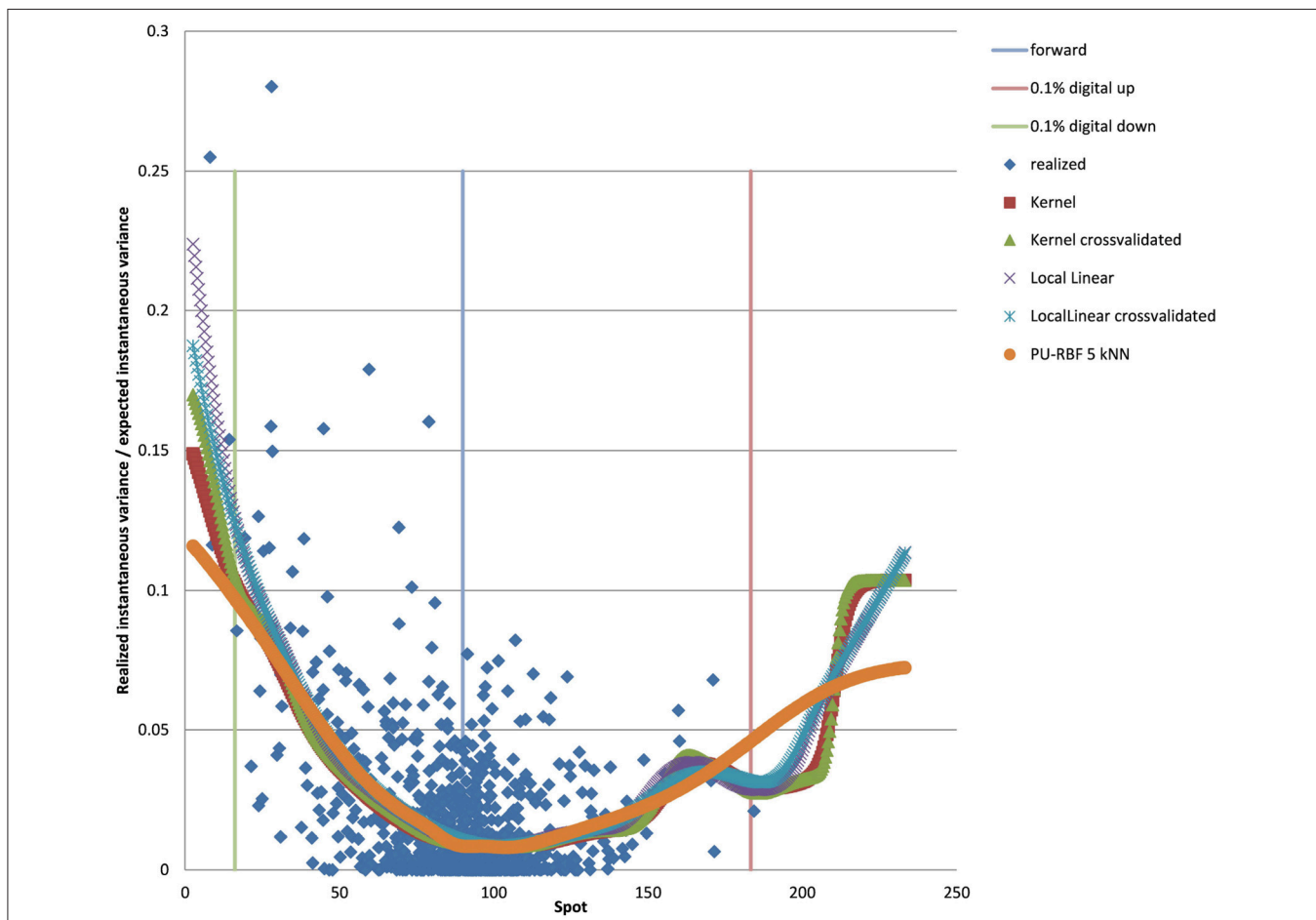


FIGURE 9 | USD/JPY 5Y estimated variance conditional on realized spot and realized paths using kernel regression, local linear kernel regression, and PURBF as well as forward and digital levels. (Source: Leonteq AG—March 2018).

In **Figure 4** you see the results of Kernel Regression for Silverman’s rule and additionally the cross-validated width for the same kernel functions. Notice that the optimal width varies between different kernels. We show results for local linear kernel regression in **Figure 5**. It can be seen that the bias at the boundaries is reduced in comparison to the kernel regression. Again we show results for Silverman’s rule and additionally the cross-validated width.

For the PURBF we show results in **Figure 6** using a global width, relative knn width, pruned and relative knn width, pruned with global width, and pruned, knn width and regularizer ($\lambda = 0.2$). We use 40 units in all cases as this number seems sufficiently versatile for the number of particles we want to use (2,048).

The last version with regularizer, pruning, and local width is the preferred version as it shows a smooth behavior without a bias in the boundaries and matches the part with many data-points in the middle without oscillations.

4.2. Example EUR/USD 5Y

We show the results for 5Y maturity and the same volatility surface in **Figure 7**. Among the tested approaches the PURBFwith 5 nearest neighbors performs best.

4.3. Example USD/JPY 5Y

We show the results for USD/JPY, see the local volatility surface in **Figure 8**. The estimation across the spot range is

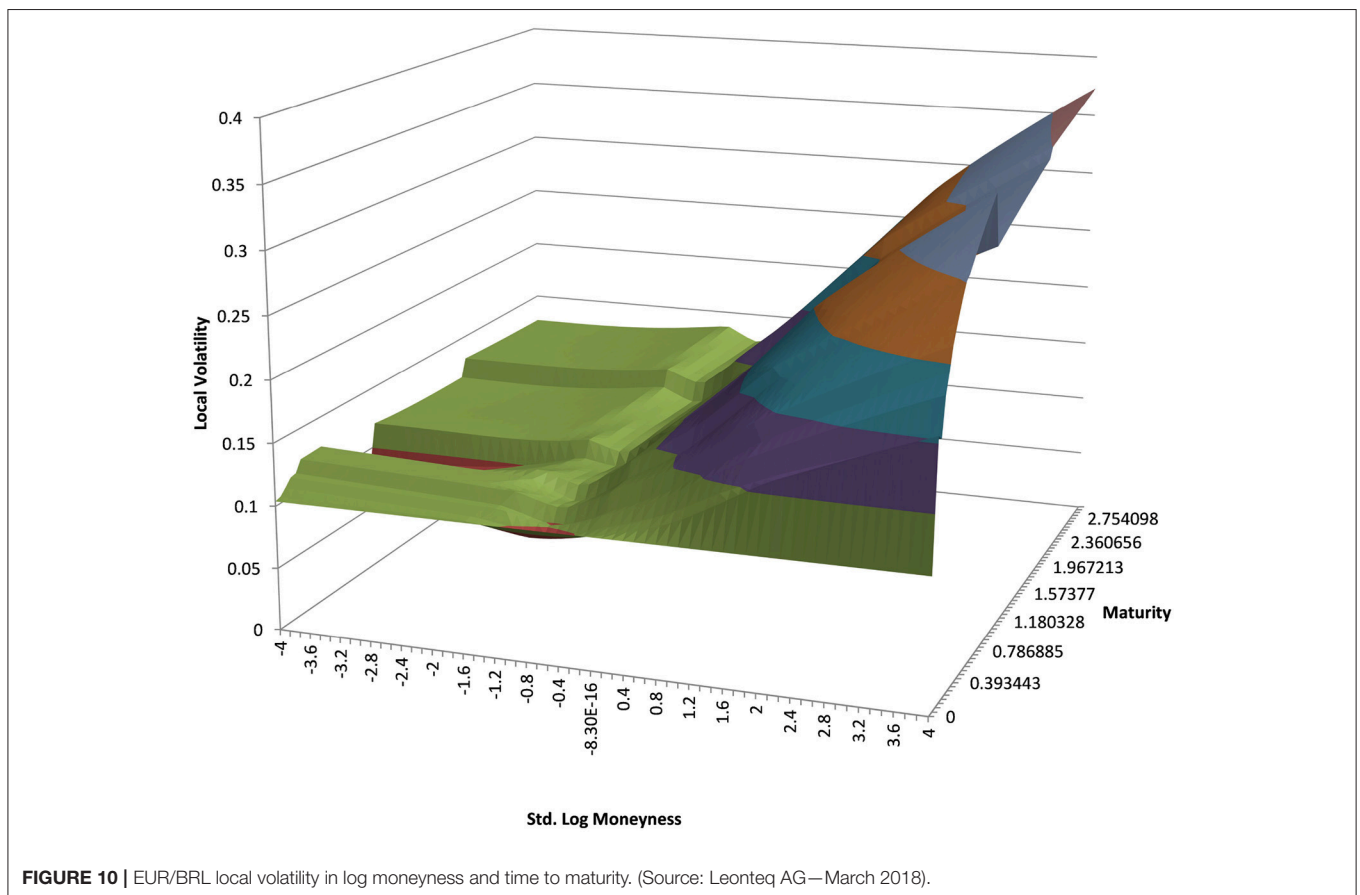
shown in **Figure 9**. Again the PURBFwith 5 nearest neighbors performs best.

4.4. Example EUR/BRL 3Y

We show the results for EUR/BRL, which is a highly skewed and highly drifting underlying. See the local volatility surface in **Figure 10**. The estimation across the spot range is shown in **Figure 11**. Note that in this case the range of spot realizations is quite skewed as is expected from the skewed volatility surface. Nevertheless the PURBFwith 5 nearest neighbors puts a relatively smooth estimator through the samples and performs better than other methods.

4.5. Pricing Examples

To see the impact on exotics pricing we look at one-touch options. A one-touch option pays one unit of the counter currency at the maturity date if the spot trades at or beyond the touch-level at any time during the life of the option. We show the impact as a function of the Black-Scholes price (TV), similar to (Clark, 2011). The TV of a one-touch can be between 0% and the discount factor to maturity, which is in the range of 100%. For fixed market parameters like spot, volatility and the risk-neutral drift TV is a function of the touch-level only, hence makes a unique scale to show the model impact. The deviation of the LSV model price from the TV is the desired effect of an alternative model, which incorporates volatility risk management



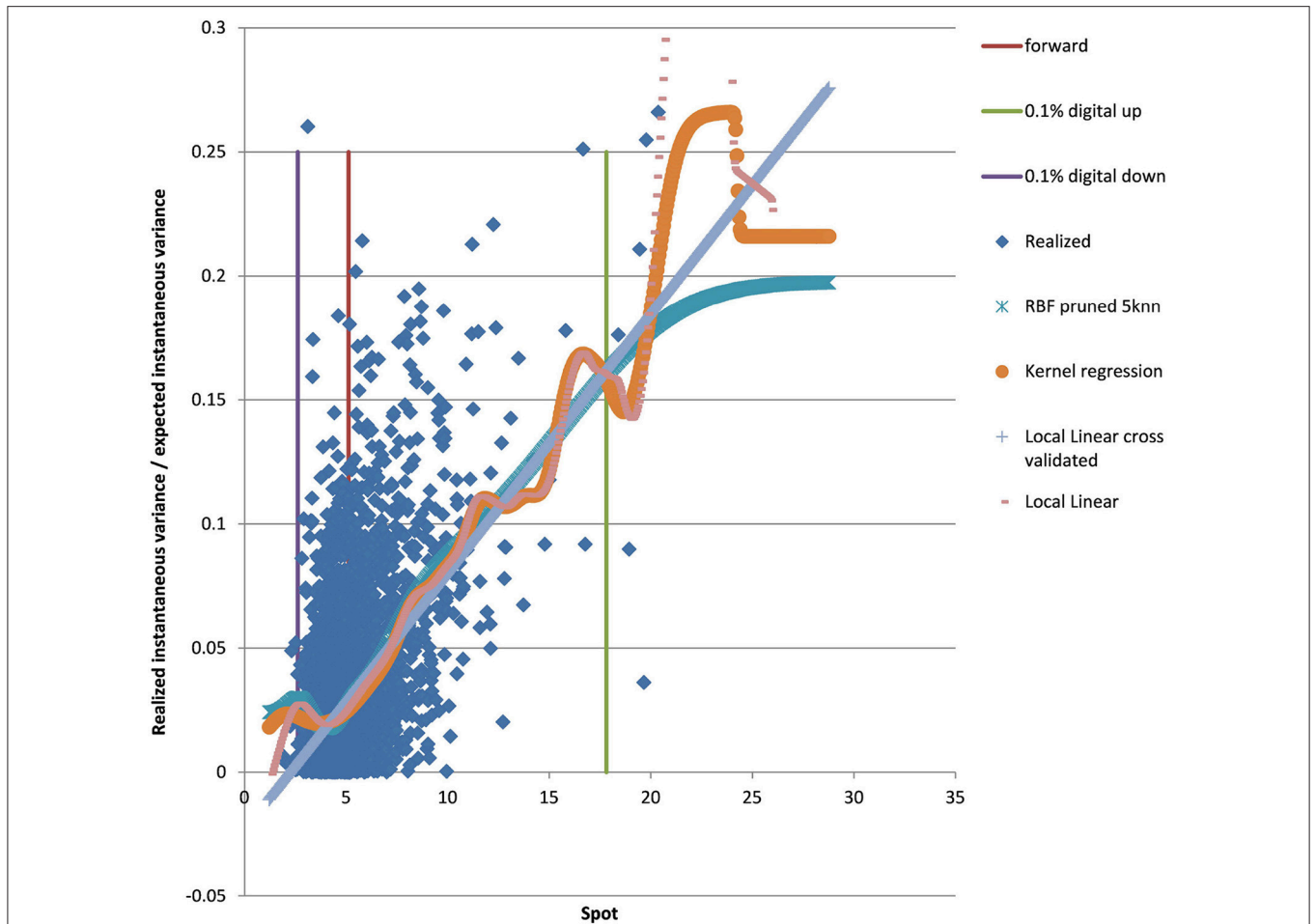


FIGURE 11 | EUR/BRL 3Y 5Y estimated variance conditional on realized spot and realized paths using kernel regression, local linear kernel regression, and PURBF as well as forward and digital levels. (Source: Leonteq AG—March 2018).

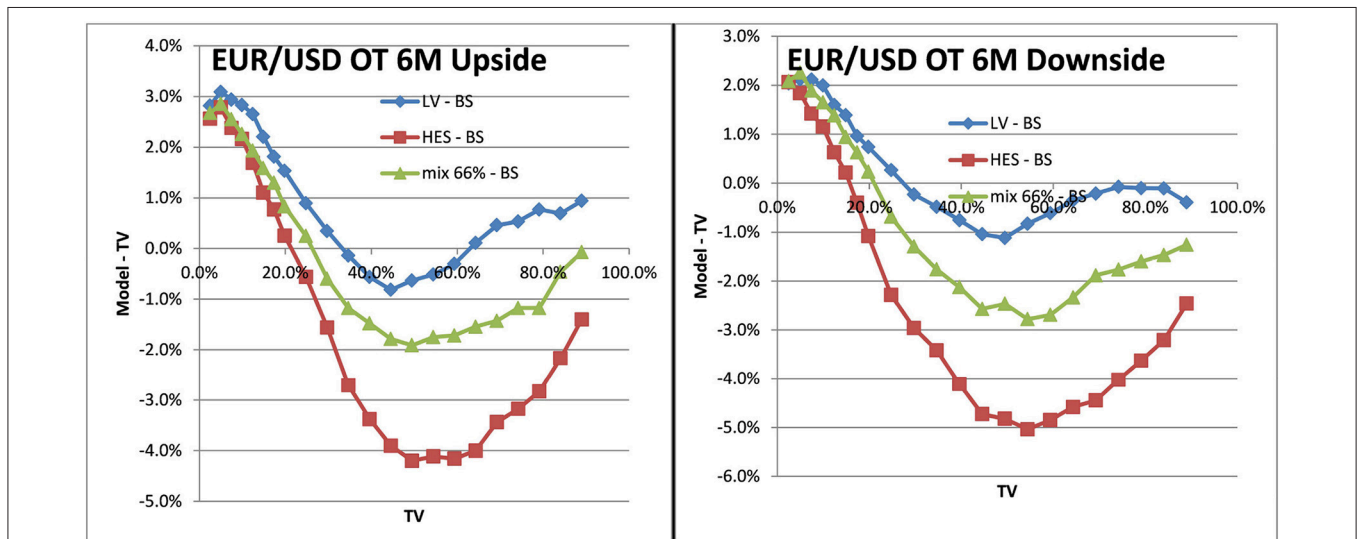


FIGURE 12 | One-touch prices EUR/USD 6M LV vs. TV, SV vs. TV, LSV vs. TV. Upside one-touches on the left, downside on the right. (Source: Leonteq AG—March 2018).

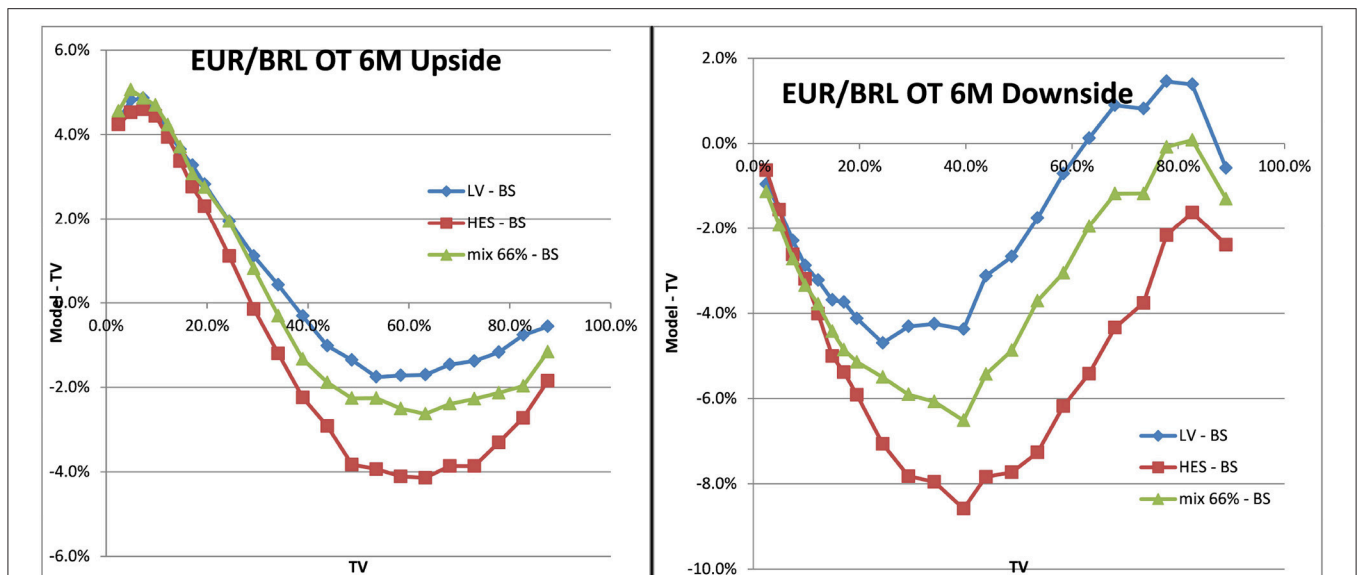


FIGURE 13 | One-touch prices EUR/BRL 6M LV vs. TV, SV vs. TV, LSV vs. TV. Upside one-touches on the left, downside on the right. (Source: Leonteq AG—March 2018).

and hedging, compared to the Black-Scholes model. The form of the deviation is not obvious and would require a rather complicated hedging argument of volatility risk and cross spot-volatility risk.

With the mixed local stochastic volatility model and mixing rate of 66% we expected the LSV price to be within the bounds of stochastic and local volatility price. We use Monte Carlo pricing with a fixed number of 32,000 paths (antithetic) and Quasi Random Numbers, a time discretization of 5 days and fixed 2048 particles. We denote the Black-Scholes prices as BS or TV (theoretical value) in the graphs and use LV as abbreviation for prices in local volatility and HES for the Heston model without local volatility component. The prices can be seen in **Figure 12** for EUR/USD 6M and in **Figure 13** for EUR/BRL 6M. We observe the expected behavior in all cases, the mixed local stochastic volatility prices are within the range of local and stochastic volatility prices and the mixing parameter can be used to adjust the behavior to observed exotics prices (e.g., one touches) in the market. Usually this mixing parameter is quite stable across longer periods, often weeks or even months.

5. CONCLUSION

We apply machine learning principles to improve the calibration process of the local stochastic volatility models. The suggested meta parameters and heuristics seem to apply to a wide variety of

underlyings in FX, liquid pairs like EUR/USD as well as emerging markets as EUR/BRL. The computational efficiency is at about the same level as for the formerly suggested Kernel Regression based approach. The results given by the PURBF function with pruning, regularization, and local width determined by 5 nearest neighbor performed significantly better than the Kernel based approaches, hence we would suggest to consider this approach in the calibration process.

Further work will be dedicated to improve the computational speed and to establish better measures of the quality. In particular in situations where vol surfaces are almost arbitrageable we will need the method to continue to provide numerically stable results.

AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and has approved it for publication.

ACKNOWLEDGMENTS

I would like to thank the Quant team of Leonteq to provide such a stimulating atmosphere, providing an environment to engage on a wide variety of practical and research oriented aspects of Quantitative Finance. In particular thanks to Nadzeya Bedziuk and Dmitry Davydov to critically crosscheck the document, still all errors are due to me.

REFERENCES

Benzoni, L., Pierre, C.-D., and Goldstein, R. S. (2011). Explaining asset pricing puzzles associated with the 1987 market crash. *J. Financ. Econ.* 101, 552–573. doi: 10.1016/j.jfineco.2011.01.008

Blacher, G. (2001). “A new approach for designing and calibrating stochastic volatility models for optimal delta-vega hedging of exotics,” in *ICBI Global Derivatives, Conference Presentation* (Juan-Les-Pins).
 Black, F., and Scholes, M. (1973). The pricing of options and corporate liabilities. *J. Polit. Econ.* 81, 637–654.

- Clark, I. J. (2011). *Foreign Exchange Option Pricing: A Practitioner's Guide*. John Wiley & Sons.
- Derman, E., and Kani, I. (1994). Riding on a smile. *Risk* 7, 32–39.
- Dupire, B. (1994). Pricing with a smile. *Risk* 7, 18–20.
- Dupire, B. (1996). “A unified theory of volatility,” in *Derivatives Pricing: The Classic Collection*, ed P. Carr (Risk Books), 185–196.
- Fritzke, B. (1994). Fast learning with incremental RBF networks. *Neural Process. Lett.* 1, 2–5. doi: 10.1007/BF02312392
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. Available online at: <https://www.deeplearningbook.org/>
- Guyon, J., and Henry-Labordere, P. (2011). The smile calibration problem solved. *SSRN Electron. J.* doi: 10.2139/ssrn.1885032
- Hagan, P. S., Kumar, D., Lesniewski, A. S., and Woodward, D. E. (2002). “Managing smile risk,” in *The Best of Wilmott*, ed P. Wilmott (Hoboken, NJ: Wiley), 249–296.
- Hakala, J., Koslowski, C., and Eckmiller, R. (1994). “Partition of unity rbf networks are universal function approximators,” in *ICANN 94* (Berlin; Heidelberg: Springer), 459–462.
- Härdle, W. (1990). *Applied Nonparametric Regression*. Cambridge University Press.
- Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Financ. Stud.* 6, 327–343. doi: 10.1093/rfs/6.2.327
- Homescu, C. (2014). Local stochastic volatility models: calibration and pricing. *SSRN Electron. J.* doi: 10.2139/ssrn.2448098
- Hull, J., and White, A. (1987). The pricing of options on assets with stochastic volatilities. *J. Finan.* 42, 281–300. doi: 10.1111/j.1540-6261.1987.tb02568.x
- Said, K. (1999). Pricing exotics under the smile. *Risk* 12, 72–75.
- Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. New York, NY: Chapman Hall.
- Van der Stoep, A. W., Grzelak, L. A., and Oosterlee, C. W. (2014). The heston stochastic-local volatility model: efficient Monte Carlo simulation. *Int. J. Theor. Appl. Finan.* 17:1450045. doi: 10.2139/ssrn.2278122
- Vapnik, V. (2013). *The Nature of Statistical Learning Theory*. Springer Science & Business Media.
- Legal Disclaimer:** This publication serves only for information purposes and is not research; it constitutes neither a recommendation for the purchase or sale of any financial products nor an offer or invitation for an offer. No representation or warranty, either express or implied is provided in relation to the accuracy, completeness, or reliability of the information herein. Before investing in financial products, investors are highly recommended to contact their financial advisor for advice specifically focused on the investor's individual situation; the information contained in this document does not substitute such advice.
- Conflict of Interest Statement:** The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
- Copyright © 2019 Hakala. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.