Long-Short Term Memory and Gated Recurrent Units neural networks as a support to investment and risk management strategies

Zacharie Guibert

August 2022

Abstract

Randomness dynamics in financial markets can be captured by the volatility, which directly links to the risk aversion of investors. As a consequence, a successful volatility prediction can be highly profitable from many aspects: For the investment manager, it can be a signal to hedge his investment portfolio in response to high upcoming volatility. For the option trader, it can be a signal for a volatility arbitrage. For the risk manager, it can be a support to forward-looking risk measures which depict the portfolio risk better.

This paper is about a classification prediction exercise in order to support investment and risk management strategies using neural networks. Two variations of Recurrent Neural Networks (RNN), the Long-Short Term Memory (LSTM) and the Gated Recurrent Units (GRU) models are employed to predict whether the volatility spread between the realized and implied volatility (RV-IV spread) ends up positive or negative on the next trading day. In particular, LSTM and GRU models adopt the same structure design - prior tuning of hyperparameters, such that their prediction performances can be compared. SP500 Index and the Eurostoxx 50 Index are used as support for the classification prediction exercise and design of investment and risk management strategies. For both indices, the GRU model outperforms the LSTM model.

Both investment and risk management strategies are built around the interpretation that a positive RV-IV spread prediction is indicative of a bear market to come as a result of higher upcoming volatility. The investment strategy, derived from SP500 GRU predictions, highlights that the GRU model does well in identifying a bear market and avoiding losses, albeit initially struggling to identify a bull market. The risk management strategy incorporates Eurostoxx 50 GRU predictions and machine learning techniques applied to volatilities and correlations to derive forward-looking risk measures in the case of an investment portfolio.

Keywords

Classification prediction, deep learning, volatility, investment strategy, forward-looking risk measures

Techniques

Applications	Model design	Model evaluation
Feature selection	Decision Tree Classifier (DTC) Random Forest Classifier (RFC) Self-Organizing Maps (SOM) Boruta Class imbalance (SMOTE, Near-Miss)	Accuracy score Confusion matrix ROC curve Classification report
Deep neural networks	LSTM / GRU Regularization (Early Stopping, Dropout, ...) Bayesian optimization for hyperparameter tuning	Accuracy score Confusion matrix ROC curve Classification report
Strategies	Hierarchical clustering Decision Trees Monte Carlo simulations	Backtest

1 Introduction

The stock market reflects the interaction of buyers and sellers, influenced by their beliefs and the economic situation. These market players may be inclined to buy or sell with clear trends at times, or act erratically at others as uncertainty rises. Therefore, being able to quantify the level of randomness in the stock market gives insights in the general behavior of the participants and trends. As a consequence, price time series are challenging to analyze because of randomness, as well as non-stationarity (market regimes change over time and across seasons), autocorrelation (the phenomenon of momentum trades), noise (a multitude of market players interact at the same time on the marketplace) as well as imperfect distributed information across market participants, among other things.

From there, the dynamics of the randomness can be captured by the volatility, which we could define as the variability of an underlying stock price over some finite time period. Volatility directly relates to some macro-economic variables as well as the amount of information available on the market: the more information, the higher the volatility [1]. Volatility is more likely to increase when the stock market falls, e.g. when leverage (debt/equity ratio) increases. Naturally, the volatility is directly linked to the risk aversion of investors, and predicting it is important for many reasons. For instance, volatility is key for investment and risk managers to mitigate portfolio risk and meet investment objectives. It is also key for derivatives pricing and option traders, where arbitrages can be made based on difference in volatility expectations.

Volatility can be either forward-looking (implied volatility) or backward-looking (realized volatility). The latter is the volatility that is observed on the financial markets, and is derived from the underlying asset returns. The former is the volatility that is expected from the market, and is derived from option prices. In times of stress in financial markets, the realized volatility becomes higher than the expected (e.g. implied) volatility, as market players integrate in present asset prices the most recent information that they didn't have at the time they built their volatility expectations. As a result, being able to predict whether the realized volatility will be higher than the implied volatility over some finite time horizon in the future is essential to effective investment and risk management strategies, as it indicates a bear market. In this regards, a key indicator to look and predict is the volatility spread between realized and implied volatility.

From the start, mathematics were a natural support to quantify, manage or predict randomness in financial markets, from econometric models to machine learning techniques. As of today, machine learning became a subset of data sciences that uses statistical models to draw insights and make predictions. Nevertheless, the key difference about machine learning solutions compared to traditional econometric models is that machine learning models learn from experience without being explicitly programmed.

This paper is not a regression prediction exercise (predict a value), but a classification prediction exercise (predict a class). The first objective of this paper is to use deep neural networks to classify the future RV-IV spread, or, in other terms, predict whether the realized volatility will be higher than the implied volatility on the next trading day. To achieve this objective, LSTM and GRU neural networks are run in parallel to strengthen the prediction task and compare their respective prediction performance. The second objective is to integrate the predictions of the volatility spread as a support to investment and risk management strategies.

2 Background

The traditional econometric approach to volatility prediction of financial time series is known as a Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model [2]. This model became and remains very popular as it accounts for the heteroskedasticity of prices time series (e.g. the variance of error terms varies as a function of some process, which makes regression modeling very difficult), as well as autocorrelation.

2.1 Deep learning

The area of machine learning has been on the spotlight in the recent years, and has now settled as a key part of statistics. The key difference between machine learning and classical programming lies in the fact that in classical programming, rules and input are fed to the engine in order to get the output. It is the contrary in machine learning, where inputs and outputs are fed to the engine in order to obtain the rules. Machine learning is all about learning some useful representation of the input data: from the rules, modeling and predictions can be made. Deep learning is a type of machine learning inspired by the structure of the human brain. This computational deep learning structure is called a neural network.

For many years, the GARCH model remained the gold standard for volatility prediction. However, in recent years, deep learning models started to outperform the GARCH model [3], which particularly contributed to their rise in popularity. From this point, several key elements to grasp machine learning and deep learning concepts are crucial for the comprehension of this paper. For readability reasons, these key elements are explain in details in a separate note :*Deep learning toolbox for classification prediction models* [4].

2.1.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are particularly helpful to predict volatility, inasmuch as RNNs are designed to handle temporal data. They have a memory, which happens to be a real support for sequential modelling such as signal prediction, time series forecasting or when it is required to predict the next word of a sentence where previous words matter.

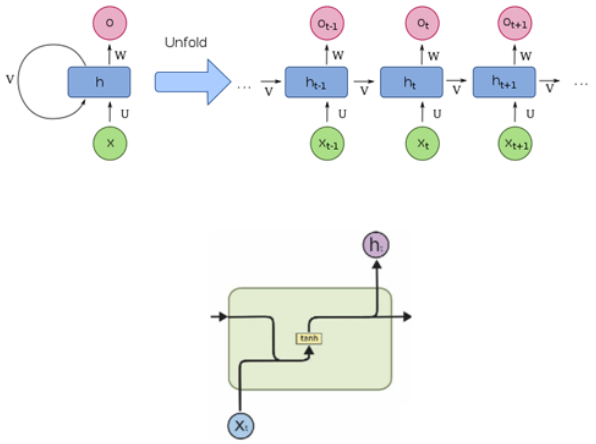

Figure 1: RNN structure (*source: Wikimedia*)

Long-Short Term Memory (LSTM) and Gated Recurrent Units (GRU) models are variants of the RNN class. The popularity of LSTMs and GRUs comes from the fact that they fix the well-known vanishing gradient problem common to RNNs [5, 6]. A LSTM model is a modified RNN that partially solves this issue by introducing feedback loops in the form of memory cells in the neural network [7]. In fact, a LSTM unit contains a memory cell, along with input and output gates to control the flow of information into and out of the LSTM unit, as well as a forget gate to allow the LSTM unit to reset its state so as to not grow continually. By construction, GRUs are considered less complex than LSTMs. The key difference between GRUs and LSTMs sits in the fact that GRUs only have two gates that are reset and update gates, unlike LSTMs, which have three gates that are input, output, and forget. Said differently, GRUs lack an output gate [8].

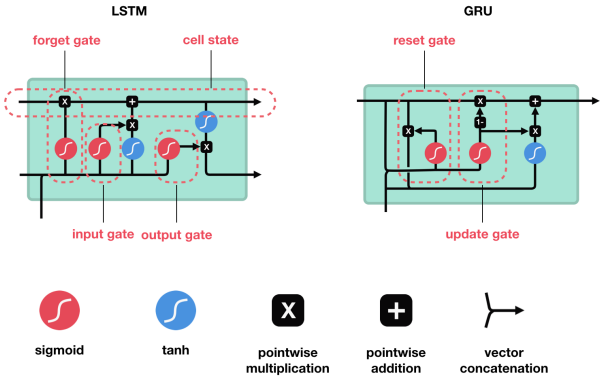

Figure 2: Differences between LSTM and GRU models (*source: Michaël Nguyen*)

As mentioned above, LSTMs and GRUs are designed to keep a memory of older data points, which makes them particularly appropriate to handle time series with varying time periods between significant events. This happens to be key when modelling stock related assets, as extreme market moves are by definition rare, but nonetheless very important to capture. Also, stock prices inherently carry autocorrelation, which LSTMs and GRUs are able to capture through their internal memory.

The natural issue with deep learning models is the large tendency to overfit, as a consequence of their low bias and high variance [9]. For that particular reason, models are trained on 80 % of the input dataset (train set), and validated on the 20% left (test set). In addition, regularization methods minimize the risk of overfitting the data, as they make slight modifications to the learning algorithm such that the model generalizes better [9]. In turn, this also improves the model performance on unseen data as well. Dropout [10] and Early Stopping [11] approaches are adopted in this paper. The dropout technique randomly selects some nodes and removes them along, such that each iteration has a different set of nodes which results in a different set of outputs (ensemble approach). Early Stopping is a cross-validation technique where part of the training set is kept aside as the validation set. Early Stopping is when model training is stopped as the model performance on the validation set gets worse.

Finally, the evaluation of the performance in classification models differs from regression models. In classification, models are evaluated, among others, with Accuracy, F1-score, ROC curve or confusion matrix, while regression models are evaluated with MAE, MSE or RMSE. Classification evaluation metrics must be used with caution, as classification datasets often suffer from imbalance, e.g. when the number of observations differ between classes. Imbalanced classifications pose a challenge for predictive modeling [12], as most of machine learning algorithms used for classification were designed around the assumption of an equal number of examples for each class. This results in models that have poor predictive performance, specifically for the minority class. This is an issue, as typically, the minority class is more important, and therefore the model is more sensitive to classification errors for the minority class than the majority class. Class imbalance can be addressed with oversampling or undersampling.

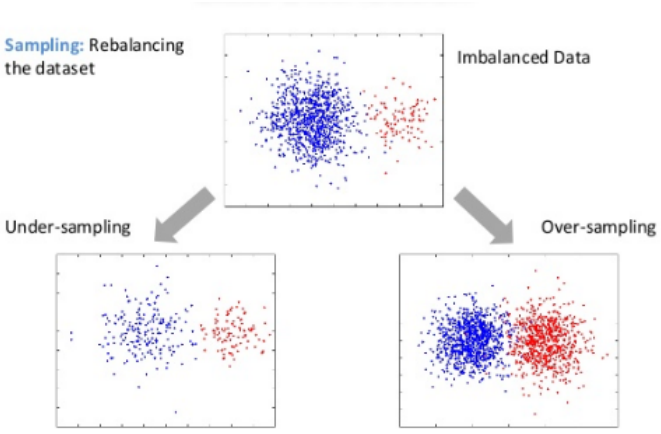

Figure 3: Addressing class imbalance (*source: Towards Data Science*)

2.1.2 Feature selection

In machine learning, feature selection is equally important as the model design, as considerable quantities of data are fed to the model so that it can learn better. However, it is

unlikely that all the features available in the dataset are useful in terms of explanatory power. More importantly, adding redundant variables reduces the generalization capability of the model and may also reduce the overall accuracy of a classifier [13]. Furthermore, adding more and more variables to a model increases its overall complexity. The objective of feature selection is to find the best set of significant features for the model.

The techniques for feature selection in machine learning can be broadly classified into supervised and unsupervised categories. Supervised techniques use labeled data, with the objective to identify the most relevant features in the data. Unsupervised techniques do not require labeled data, as the objective is to identify similar patterns across the different features. In this paper, three feature selection techniques are combined [13] to find the best feature candidates: Boruta (supervised), Decision Tree Classifier (supervised) and Self-Organizing Maps (unsupervised).

- Decision Tree Classifier (DTC) [14] creates a binary tree in order to find the best numerical or categorical feature to split the dataset, using in the case of classification an impurity criterion such as Gini impurity.
- Self-Organizing Maps (SOM) [15] is an unsupervised machine learning technique used to produce a low dimensional (typically two-dimensional) representation of a higher dimensional data set while preserving the topological structure of the data.

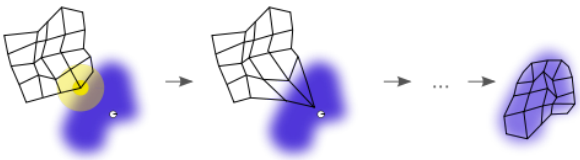

Figure 4: SOM training (*source: Wikipedia*)

- Boruta [16] is a wrapper built around the random forest classification algorithm, which returns a numerical estimate of feature importance. The importance measure of an attribute is obtained as the loss of accuracy in classification caused by the random permutation of attribute values between objects. Boruta is an all relevant feature selection method.

2.2 Volatility

As a general definition, volatility can be defined as the assessment of the degree of uncertainty. This degree of uncertainty is directly related to the quantity of information available in the market: for instance, volatility on Mondays is higher than the other week days as the first trading day of the week incorporates the news (e.g. new available information) from the weekend. Some factors have a direct effect on this degree of uncertainty. To name a few: rising interest rates, tighter policies from major central banks, increasing inflation, war, slowing growth in countries producing consumer goods such as China as well as supply chain disruptions.

This degree of uncertainty on financial markets can be measured in a forward-looking way, which refers to implied volatility. It can also be measured in a backward-looking way, which refers to realized volatility. In both ways, the volatility directly links to the amount of information that is available in the market. Predicting it attempts to get information ahead of the market.

2.2.1 Implied volatility

Implied volatility (IV) is the volatility value obtained by solving for σ in the Black Scholes option pricing formula [1], e.g. for the option price $V(S,t)$ at time t with an underlying asset price S and annualized interest rate r:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

In practise, IV is derived directly from option prices of an underlying asset, and reflects the volatility that the market expects in the future for this asset. It is popular as it is a forward-looking metric; nevertheless it assumes that the market is correctly predicting the future. Indeed, option markets are considered competitive and prices must incorporate market expectations about future volatility. It is therefore reasonable to conjecture that implied volatilities are the best source of information when forecasting volatility [1].

2.2.2 Realized volatility

Realized volatility, also called historical volatility, is a measure of price variation over a period of time [1]. Volatility is traditionally calculated based on daily log-returns, e.g. for asset prices S, log returns are expressed as $r_{t+1} = log(\frac{S_{t+1}}{S_t})$. For n the number of trading periods, N the number of trading periods in one year, and returns $r_{t-n}, ..., r_{t-1}$ whose average is $\bar{r}$, the realized volatility provides a simple estimate of the standard deviation of returns for the trading period t.

$$RV = \sqrt{\frac{N}{n-1}\sum_{i=1}^{n}(r_{t-i} - \bar{r})^2}$$

It is popular as it is easy to compute; nevertheless it is a backward-looking measure.

2.2.3 Volatility dilemna

Conditional to the same amount of available information, implied and realized volatilities with a same time horizon (for instance one month) should be equal, as both are measures of the same volatility. However, realized volatility cannot be observed for the same period as implied volatility, since implied volatility is in the future in which the realized volatility has not been observed yet. As a consequence, implied volatility (e.g. the market view of future volatility) often differ from the later observed realized volatility as both carry a different amount of information.

In some cases, the market accurately forecasts volatility as it knows something that cannot be derived from past observations. In others, the market is wrong in its volatility forecasts, which opens the door to volatility arbitrages. Nevertheless, the realized volatility is deterministic, and therefore can be used as the most accurate volatility measure. A natural estimate of the future volatility is therefore the spread between the realized volatility and implied volatility.

As mentioned above, the first objective of this paper is to predict whether the realized volatility will be above the implied volatility (e.g. a positive RV-IV spread) on the next trading day. To achieve this objective, the target variable is the volatility spread between the realized volatility and implied volatility. LSTM and GRU neural networks are used to predict the class of the RV-IV spread, e.g. if the spread will be positive or negative on the next trading day. In particular, a positive volatility spread indicates a bear market as a result of higher upcoming volatility.

2.3 Correlations

In financial theory, notions of volatility and correlation are central. If volatility no longer needs an introduction at this stage, the Capital Asset Pricing Model (CAPM) and the Arbitrage Pricing Theory (APT) use linear correlation ρ as a measure of dependence between different financial instruments. Both CAPM and APT employ an elegant theory, essentially founded on an assumption of multivariate normally distributed returns, in order to arrive at an optimal portfolio selection. In fact, this assumption of multivariate normality (or, more generally, ellipticity of the joint distribution) is necessary for linear correlation to be a meaningful measure of dependence [17]. Linear correlation is the most widely used measure of dependence between two random variables X and Y with finite variances. It is defined as

$$\rho(X,Y) = \frac{Cov[X,Y]}{\sqrt{\sigma^2[X]\sigma^2[Y]}}$$

where $Cov[X,Y]$ is the covariance between X and Y, $Cov[X,Y] \equiv E[X,Y] - E[X]E[Y]$ and $\sigma^2[X] \equiv Cov[X,X]$, $\sigma^2[Y]$ denote the variances of X and Y.

The risk of a portfolio consists of the individual risks, generally measured by the volatility, and their relationships (dependencies), measured by correlations. The correlation matrix is an aggregate of all joint dependencies between assets of a portfolio. A common approach to building a correlation matrix is to directly estimate the pairwise correlation across the securities that form an investment universe by using historical data. Factor-based correlation matrices are derived from historical returns of risk factors rather than asset returns. Nevertheless, empirical correlation matrices are unstable and backward-looking, and factor-based correlation matrices do not fully address these concerns, while failing to grasp the complex interactions among securities – both are purely observation driven, and do not impose a structural view of the investment universe, supported by economic theory [18].

2.4 Risk measurement

Finally, volatilities have a direct effect on correlations as, in times of market stress – and therefore higher volatility regime, markets tend to become more correlated, resulting in an increase in the overall risk of the portfolio [19]. Two popular measures of portfolio risk are the Value-at-Risk (VaR) and the Expected-shortfall (ES), which are derived from the covariance matrix – product between volatilities and correlations. In computing portfolio market risk measures such as Value-at-Risk, portfolio managers typically rely on correlation and volatility estimates, based on the assumption that they do not change over time. Hence, reliable estimates of correlations and volatilities are absolutely necessary to measure portfolio risk. Also, a direct consequence of a positive RV-IV spread is a change in the correlation and volatility structure of the portfolio. Adequate risk management must incorporate such signals when computing risk figures.

Portfolio VaR analysis requires the ability to generate the distribution of returns of portfolio underlying assets with a given covariance matrix of returns. In order to simulate n normal correlated variables, a popular technique is the Cholesky decomposition [20], which factors a positive definite correlation matrix ρ into a unique product of lower triangular matrix and its transpose $\rho = L \cdot L^T$. Said differently, this means that for any $j > i$,

$$\rho_{ij} = \sum_{k=1}^{i} l_{ik} l_{jk}$$

For example, for $n = 2$, L becomes

$$L = \begin{pmatrix} 1 & 0 \\ \rho_{12} & \sqrt{1-\rho_{12}^2} \end{pmatrix}$$

For n uncorrelated normal variables $X = x_1, ..., x_n$ (for example returns of financial assets), the transformed variables $Y = L \cdot X^T$ now have a joint multivariate normal distribution which incorporates the correlation structure ρ. For invertibility reasons, both correlation and covariance matrices must be positive semi-definite (e.g. have strictly positive eigenvalues).

3 Dataset, features and label

The illustration of the ability of LSTM and GRU neural networks to predict the volatility spread is made on two underlying assets: the SP500 Index and the Eurostoxx 50 Index. The SP500 model acts as a proof of concept, while the Eurostoxx 50 incorporates a more complex structure of features.

3.1 Model I: SP500 Index

3.1.1 Data

Model I consists in predicting whether the spread between SP500 realized volatility and SP500 implied volatility (RV-IV spread) will end up positive or negative on the next trading day. The dataset is made of SP500 daily prices to

calculate SP500 daily returns and realized volatility, the VIX Index to get the SP500 implied volatility, as well as the US 10-year Government yield as potential additional prediction feature. Data is collected from the the 1^{st} of January 2011 to the 30^{th} of June 2022, for a total 2,891 daily prices. All the data is obtained from Yahoo Finance.

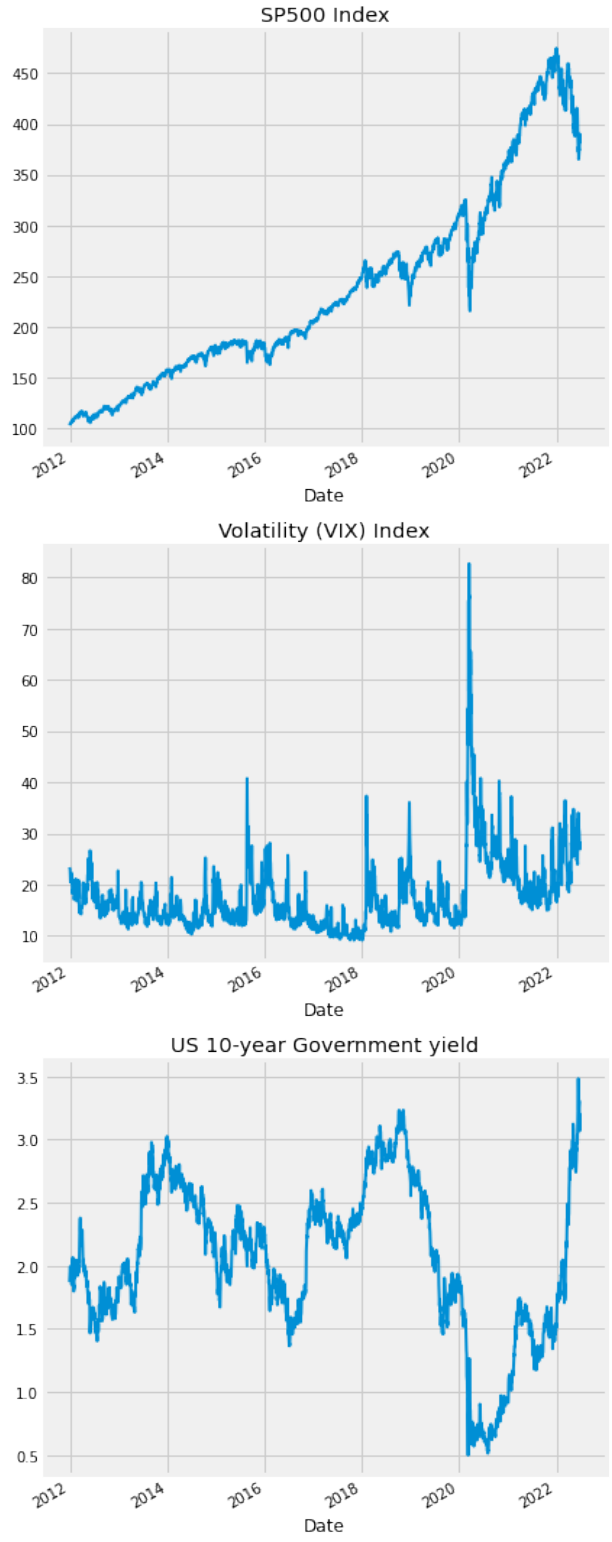

Figure 5: SP500, VIX and US 10-year Government yield from 2011 to 2022 (*source: Yahoo finance*)

The VIX Index represents the market expected SP500 volatility in a month time. From this point, SP500 realized volatility refers to the standard deviation of SP500 returns over the past 21 days of observations (one month), such that implied and realized volatility share the same time horizon.

3.1.2 Features

Among others, returns, momentum, Exponential Moving Average (EMA), historical volatility, Average True Range (ATR), Bollinger bands and others are calculated when possible for different lagged period, from 1 day to 252 days. In total, 156 features are available to predict the SP500 RV-IV spread over 2,638 days, from the 30^{th} of December 2011 until the 29^{th} of June 2022. All features are scaled with the Min-Max approach in order to account equally for their respective influence in the prediction model [13].

3.1.3 Label

The label (target variable) is the SP500 RV-IV spread, which gets the value 1 if the realized volatility is higher than the implied volatility on the next trading day.

$$y_t = \begin{cases} 1 & \text{if } realized_{\,t+1} > implied_{\,t+1} \equiv \text{RV-IV}_{t+1} > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $realized_{\,t+1}$ is the 1-day forward SP500 realized volatility, $implied_{\,t+1}$ is the 1-day forward SP500 implied volatility, and RV-IV_{t+1} the difference between the two. Over the 2,638 days of available volatility figures, the realized volatility was above the implied volatility on a total of 411 days, so around 16% of the time. At this stage, the dataset is imbalanced.

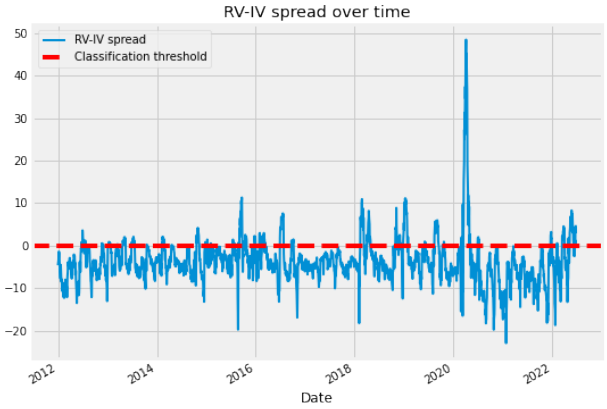

Figure 6: SP500 RV-IV spread (*source: Python notebook M1, section 1.3*)

Finally, the ADF statistical test confirms that the SP500 RV-IV spread time series is stationary above 99% confidence level.

3.1.4 Feature selection

Prior feature selection, all time series were split into a train set of 80% for model training and test set of 20% for model validation purposes. Train and test intervals do not overlap to prevent look-ahead bias. As mentioned

in section 2, feature selection is equally important as the prediction model [13].

138 features out of 156 are classified as relevant for the RV-IV classification prediction as a result of Boruta algorithm. From there, two additional feature sets are proposed.

- Choice A: Boruta 138 features are successively passed to a Decision Tree Classifier (DTC) and a Self Organizing Map (SOM). Choice A features only consists of DTC and SOM 20 most important features with a correlation below 75%.
- Choice B: Boruta 138 features are simply passed to a correlation condition. Choice B features simply lists Boruta features with a correlation below 75%.

In total, Choice A represents 14 features and Choice B represents 26 features. The ROC curve is a particularly appropriate tool to choose between Boruta, Choice A and Choice B feature sets, as it indicates how well each model distinguishes between the two classes it aims to predict. ROC curve results indicate the Boruta model brings the most explanatory power to the prediction exercise, with a total AUC score of 81%, but with 138 features among which only 26 have a correlation below 75%. Therefore, in order to avoid overfitting and as a trade off between number of features and AUC score, Choice B features set is the best candidate for our classification problem, as it has only 26 features for an AUC score of 74%. Choice A feature set is the less significant, with an AUC score of 64%.

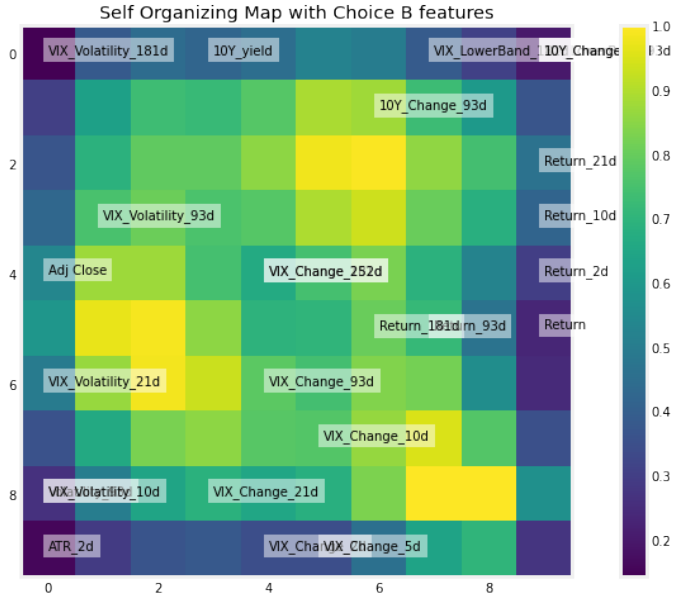

Figure 7: Self-Organizing Map of Choice B feature set (*source: Python notebook M1, section 2.4.2*)

3.1.5 Class imbalance

Up to this point, and as highlighted in section 3.1.3 above, the dataset is imbalanced, as only 16% of observations have a realized volatility above the implied volatility. Oversampling with the SMOTE algorithm results in a train accuracy of 88% and a test accuracy of 73%. On the other hand, under-sampling with the Near-Miss algorithm results in a train accuracy of 78% and a test accuracy of 66%. The natural choice is to use the oversampled dataset as input for the prediction exercise as a result of greater accuracy and aligned tendency to overfit in comparison with the undersampled dataset.

3.2 Model II: Eurostoxx 50 Index

3.2.1 Data

Model II consists in predicting whether the spread between Eurostoxx 50 realized volatility and Eurostoxx 50 implied volatility (RV-IV spread) will end up positive or negative on the next trading day. Model II incorporates additional features as inputs to the prediction model to estimate whether more features bring additional performance in the prediction task. The dataset is made of Eurostoxx 50 daily prices to calculate Eurostoxx 50 daily returns and realized volatility, and the V2X Index to get the Eurostoxx 50 implied volatility. In addition, model II incorporates variables which are assumed to have an impact on volatility: Euro 2, 5 and 10-year Government yields, 6M EURIBOR, as well as daily prices for oil (*CO1 Comdty*) and wheat (*W1 COMB Comdty*). The dataset represents in total 3,259 daily prices, from the the 1^{st} of January 2010 to the 30^{th} of June 2022. All the data is obtained from Bloomberg.

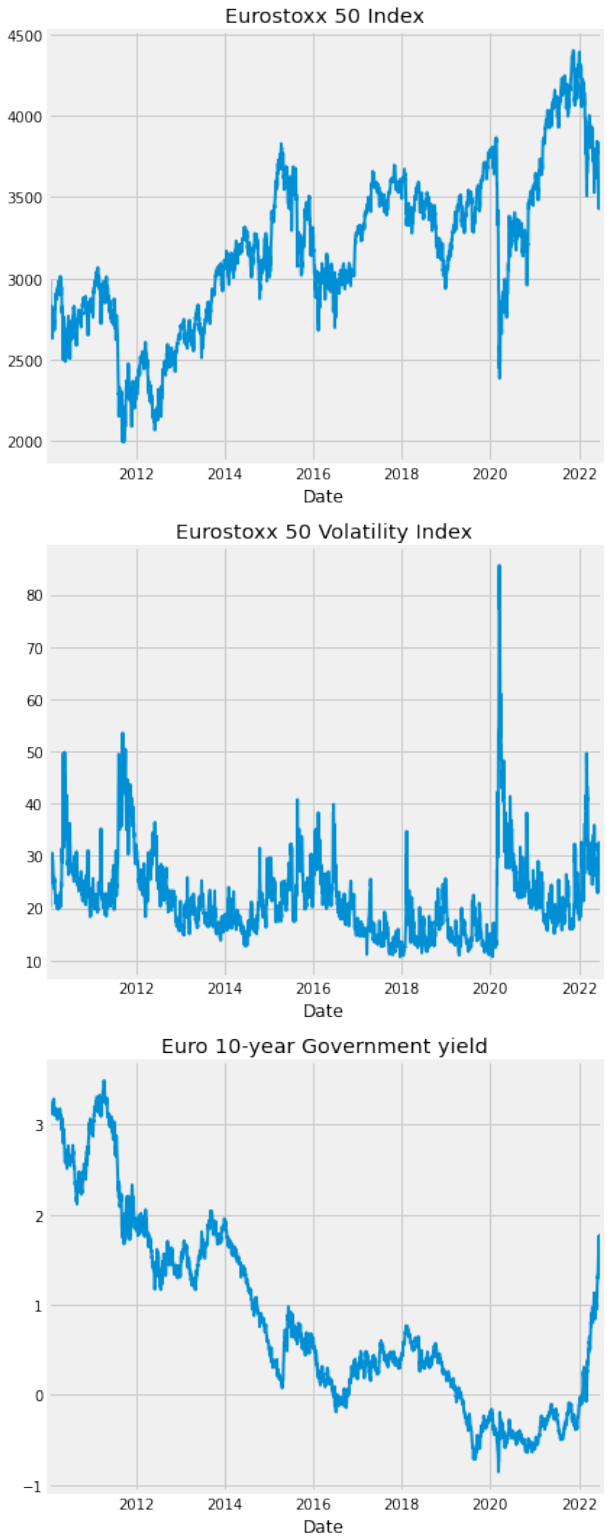

Figure 8: Eurostoxx 50, V2X and Euro 10-year Government yield from 2010 to 2022 (*source: Bloomberg*)

As for model I, the V2X Index represents the market expected Eurostoxx 50 volatility in a month time. From this point, Eurostoxx 50 realized volatility refers to the standard deviation of Eurostoxx 50 returns over the past 21 days of observations (one month), such that implied and realized volatility share the same time horizon.

3.2.2 Features

Among others, returns, momentum, Exponential Moving Average (EMA), historical volatility, Bollinger bands and others are calculated when possible for different lagged period, from 1 day to 252 days. In addition, the dataset is complemented with forward-looking GARCH volatility predictions 1 day, 5 days and 10 days ahead. In total, 334 features are available to predict the Eurostoxx 50 RV-IV spread over 3,006 days, from the 22^{nd} of December 2011 until the 29^{th} of June 2022. All features are scaled with the Min-Max technique in order to account equally for their respective influence in the prediction model [13].

3.2.3 Label

The label (target variable) is the Eurostoxx 50 RV-IV spread, which gets the value 1 if the realized volatility is higher than the implied volatility on the next trading day.

$$y_t = \begin{cases} 1 & \text{if } realized_{\,t+1} > implied_{\,t+1} \equiv \text{RV-IV}_{t+1} > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $realized_{\,t+1}$ is the 1-day forward Eurostoxx 50 realized volatility, $implied_{\,t+1}$ is the 1-day forward Eurostoxx 50 implied volatility, and RV-IV_{t+1} the difference between the two. Over the 3,006 days of available volatility figures, the realized volatility was above the implied volatility on a total of 567 days, so around 19% of the time. At this stage, the dataset is imbalanced.

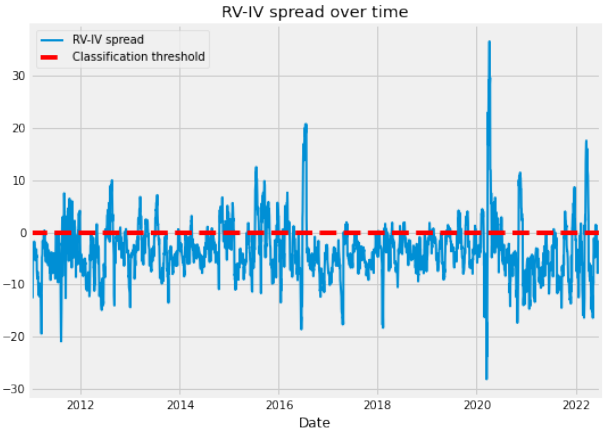

Figure 9: Eurostoxx 50 RV-IV spread (*source: Python notebook M2, section 1.3*)

Finally, the ADF statistical test confirms that the Eurostoxx 50 RV-IV spread time series is stationary above 99% confidence level.

3.2.4 Feature selection

Prior feature selection, all time series were split into a train set of 80% for model training and test set of 20% for model validation purposes. Train and test intervals do not overlap to prevent look-ahead bias. As for model I, Boruta selection algorithm is first used. In total, 294 features out of 334 are classified as relevant for the RV-IV spread classification prediction as a result of Boruta

algorithm. From there, two additional feature sets are proposed.

- Choice A: Boruta 294 features are successively passed to a Decision Tree Classifier (DTC) algorithm and a Self Organizing Map (SOM). Choice A features only consists of DTC and SOM 20 most important features with a correlation below 75%.
- Choice B: Boruta 135 features are simply passed to a correlation condition. Choice B features simply lists Boruta features with a correlation below 75%.

In total, Choice A represents 19 features and Choice B represents 40 features. ROC curve results indicates that Choice B brings the most explanatory power to the prediction exercise, with a total AUC score of 77%. Boruta also shows an AUC score of 77% (but with 8 times more features) and Choice A 74%.

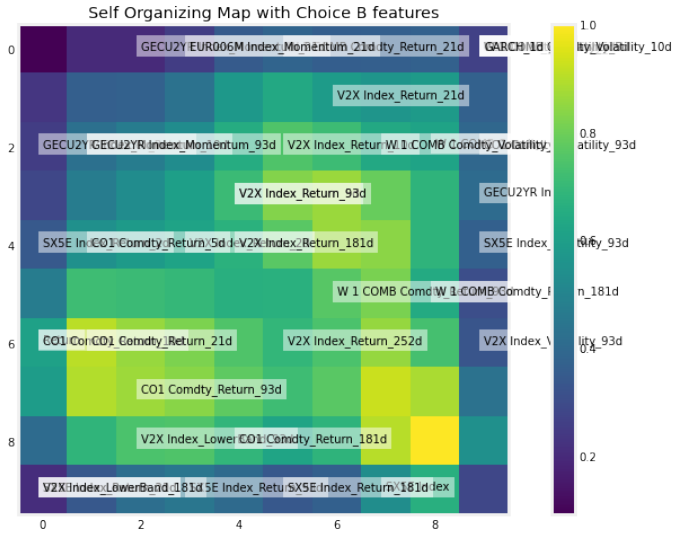

Figure 10: Self-Organizing Map of Choice B feature set (*source: Python notebook M2, section 2.4.2*)

3.2.5 Class imbalance

Up to this point, and as highlighted in section 3.2.3 above, the dataset is imbalanced as they are only 19% of dataset observations where the realized volatility is above the implied volatility. Over-sampling with the SMOTE algorithm results in a train accuracy of 76% and a test accuracy of 60%. On the other hand, under-sampling with the Near-Miss algorithm results in a train accuracy of 65%, and a test accuracy of 26%. The natural choice is to use the oversampled feature set for the prediction exercise, as the under-sampling method largely suffers from overfitting.

4 Methods

4.1 Baseline model

The main function of a baseline model is to contextualize the results of trained models. An appropriate baseline model should be simple (no training nor intelligence), fast (computationally trivial to make predictions) and repeatable (deterministic). As a result, the expected accuracy of the baseline model is around 50% on a balanced dataset. Here, a dummy classifier serves as a baseline model and acts as benchmark to monitor the performance of LSTM and GRU classifiers. More specifically, the dummy classifier that is used in this paper generates predictions uniformly at random. In other words, it makes predictions that ignores the input features, and outputs a uniform distribution at random of 0 and 1.

4.2 LSTM and GRU models

Beyond predicting the class, the objective of this paper is also to compare the performance of LSTM and GRU neural networks on a similar task [8]. Hence, LSTM and GRU models were designed with the same architecture for comparability reasons: First, both models have a stacked structure of five layers: one input layer, three hidden layers as well as one output layer (Dense layer). Second, there are two separate dropout layers after the first two hidden layers to prevent overfitting [9]. Third, the Dense output layer is only made of of unit (neuron), and has a sigmoïd activation function. Fourth, the Adam optimizer is used in both models, and the loss function is the binary cross entropy inasmuch as it is a classification prediction problem. As a complement to the two dropout layers, both models also incorporate Early Stopping for a better out-of-sample generalization [9].

4.3 Hyperparameter tuning

Neural networks such as LSTM and GRU models have many different parameters that can be optimized to better fit the available data. Hyperparameters are parameters whose values are set before the learning process begins - e.g. parameters that are not directly learnt within estimators. By contrast, the value of simple parameters are derived through training. Hyperparameter tuning is the process of choosing the optimal set of parameters in order to get the best prediction model, and is consequently essential in model design.

Hyperparameter tuning is performed through a Bayesian optimization. Bayesian methods attempt to build a probability distribution over some possible function, called the surrogate function, that estimates how good the model might be for a given choice of hyperparameters. Bayesian optimization runs models many times with different sets of hyperparameter values, and evaluates the past model information to select hyperparameter values to build the newer model based on the surrogate function [21].

The tuning process with the Bayesian optimization results in six hyperparameters: the number of units in each of the three layers, the dropout rate, the learning rate and the activation function. Despite distinct dropout rates and activation functions at each layer level when applicable, the algorithm returns one best value for the dropout rate as well as the best activation function, applicable to the entire model.

Hyperparameter	Minimum	Maximum	Step
Neurons in layer 1	4	32	4
Neurons in layer 2	4	32	4
Neurons in layer 3	4	32	4
Dropout rate 1	0%	50%	10%
Dropout rate 2	0%	50%	10%

Hyperparameter	Choice 1	Choice 2	Choice 3
Learning rate	0.01	0.001	0.0001

Activation function	Choice 1	Choice 2
Layer 1	relu	elu
Layer 2	relu	elu
Layer 3	relu	elu

Table 1: Eligible hyperparameter values

5 Results and discussion

5.1 Model I: SP500 Index

5.1.1 Hyperparameters

Model evaluation and results of Baseline, LSTM and GRU models are based on the following set of hyperparameters, obtained with Bayesian optimization.

Hyperparameter	Baseline	LSTM	GRU
Neurons in layer 1	N/A	32	8
Neurons in layer 2	N/A	32	4
Neurons in layer 3	N/A	32	20
Dropout rate	N/A	0%	40%
Learning rate	N/A	0.0001	0.01
Activation function	N/A	elu	elu

Table 2: Model I hyperparameters

5.1.2 Model evaluation

Results below illustrate the performance of the baseline, LSTM and GRU models. Confusion matrix, accuracy score and ROC AUC score serve as support to evaluate model performance [12].

	Baseline	LSTM	GRU
True Positive (TP)	25	36	53
True Negative (TN)	219	372	332
False Positive (FP)	233	80	120
False Negative (FN)	30	19	2

Table 3: Model I confusion matrix

As mentioned above, the ROC curve is particularly helpful in a classification exercise, as it illustrates how well the model is successful in distinguishing between two classes. The higher the AUC score, the better.

	Baseline	LSTM	GRU
ROC auc score	47%	74%	85%
Train accuracy	48%	92%	89%
Test accuracy	48%	80%	76%
Convergence	N/A	100 epochs	20 epochs

Table 4: Model I evaluation

5.1.3 Discussion

From all perspectives, LSTM and GRU models largely outperform the baseline model once run with best hyperparameters, which confirms their validity and relevance. At first sight, both LSTM and GRU models seem to perform equally well, as they show similar train and test accuracy. Both succeed in limiting overfitting, as illustrated by an out-of-sample accuracy close to 80% relatively close to the in-sample accuracy.

Confusion matrix results indicate that GRU predicts better the minority class (class 1, positive RV-IV spread) than LSTM, as illustrated by the higher number of True Positive. This is however to the detriment of the majority class (class 0, negative RV-IV spread), as illustrated by a lower number of True Negative.

The ROC AUC score illustrates GRU's ability to better distinguish between the two prediction classes than LSTM, as both respectively have a ROC auc score of 85% and 74%. Finally, GRU is faster to converge than LSTM.

5.1.4 Conclusion

LSTM and GRU models proved to be solid candidates for classification prediction models, as they successfully predict the class of the SP500 RV-IV spread on the next trading day, with an out-of-sample accuracy around 80%. The joint evaluation of LSTM and GRU models showed that GRU performed better than LSTM in the classification prediction task.

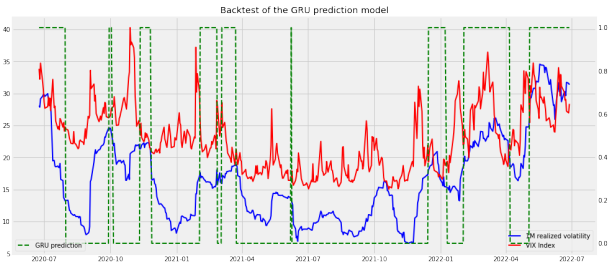

Figure 11: GRU model backtest on SP500 Index (*source: Python notebook M1, section 6.5*)

5.2 Model II: Eurostoxx 50 Index

5.2.1 Hyperparameters

Model evaluation and results of Baseline, LSTM and GRU models are based on the following set of hyperparameters, obtained with Bayesian optimization.

Hyperparameter	Baseline	LSTM	GRU
Neurons in layer 1	N/A	32	12
Neurons in layer 2	N/A	24	20
Neurons in layer 3	N/A	32	12
Dropout rate	N/A	0%	30%
Learning rate	N/A	0.0001	0.01
Activation function	N/A	elu	elu

Table 5: Model II hyperparameters

5.2.2 Model evaluation

Results below illustrate the performance of the baseline, LSTM and GRU models. Confusion matrix, accuracy score and ROC AUC score serve as support to evaluate model performance [12].

	Baseline	LSTM	GRU
True Positive (FP)	59	85	98
True Negative (TN)	224	361	333
False Positive (FP)	242	105	133
False Negative (FN)	56	30	17

Table 6: Model II confusion matrix

Here again, the ROC curve provides good insights on model II ability to distinguish between the two classes.

	Baseline	LSTM	GRU
ROC auc score	50%	76%	78%
Train accuracy	49%	91%	93%
Test accuracy	49%	77%	74%
Convergence	N/A	150 epochs	100 epochs

Table 7: Model II evaluation

5.2.3 Discussion

From all perspectives, LSTM and GRU models largely outperform the baseline model once run with best hyperparameters, which confirms their validity and relevance. As first sight, both LSTM and GRU models seem to equally perform, as they show similar train and test accuracy. Both succeed in limiting overfitting, as illustrated by the out-of-sample accuracy close to 75% relatively close to the in-sample accuracy.

Confusion matrix results indicate that GRU predicts better the minority class (class 1, positive RV-IV spread) than LSTM, as illustrated by the higher number of True Positive. This is however to the detriment of the majority class (class 0, negative RV-IV spread), as illustrated by a lower number of True Negative.

The ROC AUC score illustrates GRU's ability to better distinguish between the two prediction classes than LSTM, as both respectively have a ROC AUC score of 78% and 76%. Finally, GRU is faster to converge than LSTM.

5.2.4 Conclusion

Once again, LSTM and GRU models proved to be solid candidates for classification prediction models, as they successfully predict the class of the Eurostoxx 50 RV-IV spread class on the next trading day, with an out-of-sample accuracy around 75%. The joint evaluation of LSTM and GRU models showed that GRU performed better than LSTM in the classification prediction task.

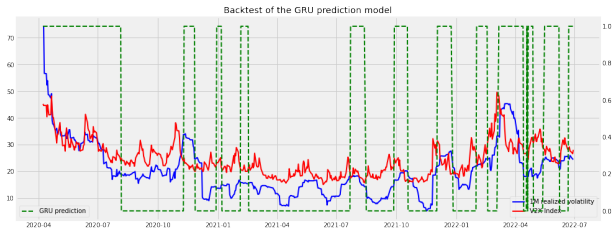

Figure 12: GRU model backtest on Eurostoxx 50 (*source: Python notebook M2, section 6.5*)

6 Applications

Section 5 demonstrated the ability of LSTM and GRU models to successfully predict the sign (e.g. class) of the RV-IV spread between the realized and implied volatility on the next trading day. The objective of this section is to fit the results of the volatility predictions into investment and risk management strategies, and illustrate how LSTM and GRU models can support business decisions.

6.1 Investment Management

6.1.1 Strategy

Implied volatility is generally greater than realized volatility, as corroborated by Figure 6 and Figure 9 above, which results in a negative RV-IV spread over time. The mostly negative RV-IV spread illustrates that the market overestimates the volatility yet to come. However, realized volatility tends to be above the implied volatility in times of market stress, as the market incorporates a large amount of news in reaction to recent events. From there, it is reasonable to assume that a positive RV-IV spread indicates a bear market territory, as a result of higher upcoming volatility. A natural investment strategy comes: Go long when the model predicts a negative RV-IV spread, go cash when the model predicts a positive RV-IV spread, e.g. when it predicts a bear market. This strategy is illustrated on the SP5000 Index with the GRU classification prediction model obtained in section 5.1 above.

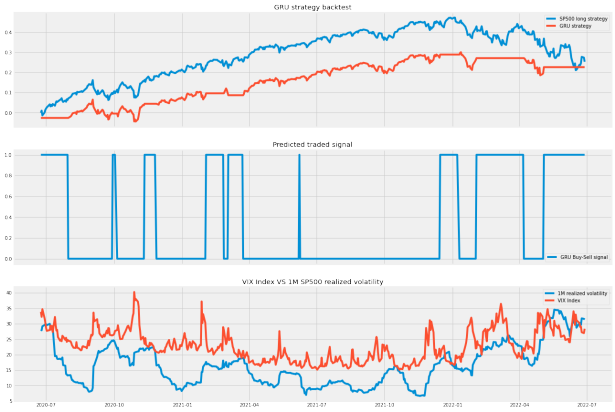

Figure 13: GRU SP500 investment strategy (*source: Python notebook M1, section 8.2*)

6.1.2 Discussion

The SP500 investment strategy based on GRU predictions performs just as well as the long SP500 strategy, as illustrated by the first subplot on Figure 13.

- From the beginning of the test period in June 2020 until the end of 2021, the SP500 Index strongly recovered from Covid-19 Q1/Q2 2020 shortfall. The massive cash injection from central banks of major economies worldwide to balance lockdown consequences drove equity indices up, including the US. As a result of FOMO (Fear of Missing Out), getting out of the market during this period generally resulted in missed performance despite higher volatility.
- From January 2022 until June 2022, as a result of rising inflation, war in Ukraine and supply-chain disruptions due to Covid-19 strong recovery, the economy and financial markets went back to their fundamentals. During this period, the GRU model performs particularly well, by going cash when the SP 500 Index entered into a bear market in February - March 2022 as well as May - June 2022.

In summary, most of the time, the return of the GRU strategy was below the SP500 long strategy return up until June 2022 where they converged to similar values. It appears that the GRU stratey does well in identifying the bear market and avoiding losses while it initially struggled a little identifying the bull market after the Covid-19 shock.

Furthermore, such volatility classification model could also be used as support to option traders. From the relationship between implied and realized volatility can be derived the conclusions whether options are over or undervalued, and therefore motivate volatility arbitrage decisions.

6.2 Risk Management

6.2.1 Strategy

In 2019, Marcos López de Prado proposed a methodology to estimate Theory-Implied Correlation (TIC) matrices. TIC methodology introduces a machine learning algorithm to estimate forward-looking correlation matrices implied by economic theory. Given a particular theoretical representation of the hierarchical structure that governs a universe of securities, the method fits the correlation matrix that complies with that theoretical representation of the future [18]. The risk management strategy therefore consists in proposing a methodology to compute a forward-looking covariance matrix with the help of the TIC matrix and Eurostoxx 50 GRU volatility prediction model, and from there derive forward-looking portfolio risk measures such as VaR or Expected Shortfall - as follows:

- First, the TIC matrix is obtained from the historical correlation matrix as well as GICS sector information on underlying portfolio assets.
- Second, the TIC matrix is denoised [22] in order to remove any information or activity that confuses or misrepresents genuine underlying trends.
- Third, 1-month realized volatitilies of portfolio underlying assets incorporate the result of the GRU volatility prediction model: if it predicts a positive RV-IV spread, volatilities are increased by 25% to reflect the increase in volatility coming on the next trading day. If it predicts a negative RV-IV spread, then historical volatilities are taken as true estimates of present volatility [23].
- Fourth, the forward-looking covariance matrix is computed from the matrix product between the denoised TIC matrix obtained from step 2 and GRU prediction adjusted volatilities obtained from step 3. The forward-looking covariance matrix must be positive semi-definite.
- Fifth, forward-looking portfolio risk measures are estimated from the covariance matrix obtained in step 4.

6.2.2 Data

For illustration purposes, an investment model portfolio is made of 25 Eurostoxx 50 underlying assets with equal weight of 4% each as of 30 June 2022. The historical correlation matrix is obtained with 5 years of daily prices (1,282 observations). From there, historical, TIC and denoised TIC matrices are computed. The computation of TIC matrices requires the GICS sector information for all 25 underlying assets in the portfolio. All data is provided by Yahoo Finance. Dependencies between assets now appear much more stable on TIC and denoised TIC matrices.

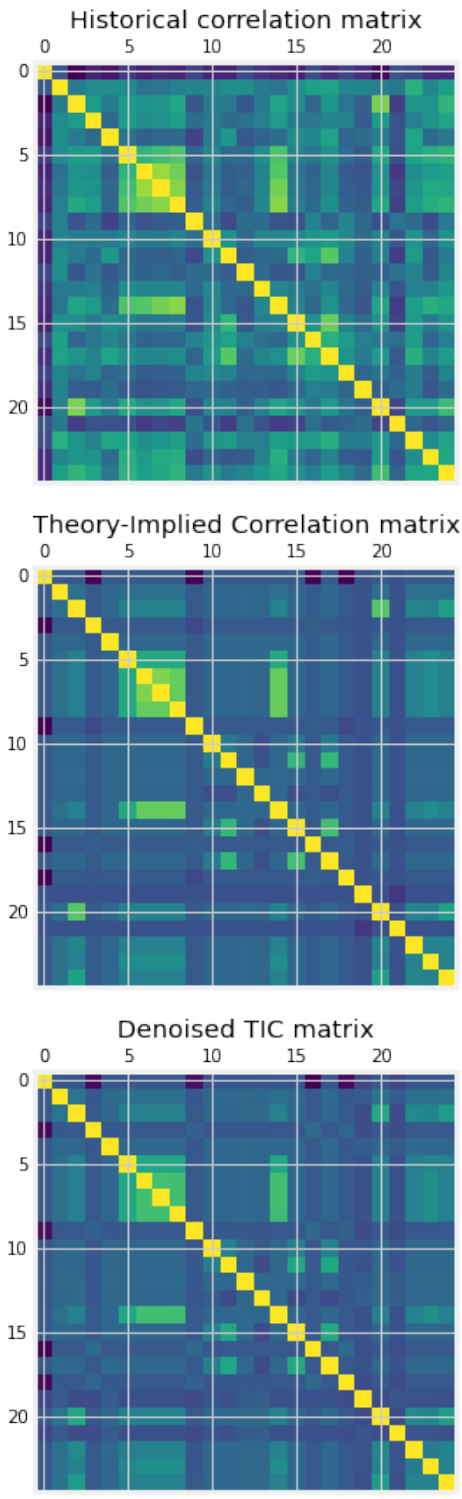

Figure 14: Historical, TIC and denoised TIC matrices of the Eurostoxx 50 portfolio (*source: Python notebook M2, section 8.2 and 8.3*)

6.2.3 Risk measures

Risk measures are the result of 10,000 Monte-Carlo simulations on an horizon of one month at 99% confidence level. Risk measures are estimated according to three scenarios. A first scenario computes the risk measures based on historical volatilities and historical correlations only. A second scenario computes the risk measures with historical volatilities and denoised TIC matrix. Finally, a third scenario computes the risk measures with GRU stressed volatilities and denoised TIC matrix.

Scenario	Volatility	Correlation	VaR	ES
Scenario 1	Historical	Historical	23.0%	24.8%
Scenario 2	Historical	Denoised TIC	17.6%	18.6%
Scenario 3	GRU stressed	Denoised TIC	19.6%	20.9%

Table 8: Eurostoxx 50 portfolio risk measures

6.2.4 Discussion

An efficient risk management framework must be guided by a forward-looking view, in order to rely and communicate on accurate forecasts of the risks coming ahead. In that regards, the denoised TIC and GRU volatility prediction model can help: the latter indicates the direction of the volatility of the next trading day, whether it will be above or below what the market expects. The former proposes a correlation structure that complies with a theoretical representation of the future. In this manner, a forward-looking view on volatilities and correlations can be factored into a forward-looking covariance matrix, which can in turn be used as support to the calculation of forward-looking risk measures.

Scenarios 2 and 3 result in lower risk measures than scenario 1, which is simply derived from historical volatilities and correlations. Nevertheless, this result does not invalidate the model, but the contrary, inasmuch as it reflects the model predictions of normal volatility and weaker correlations to come.

The approach developed across section 6.2 shows several advantages: Beyond offering a forward-looking estimation of portfolio risk, it gives an ideal stress-testing framework in order to build scenario analysis. The denoised TIC matrix and volatilities can be stressed to reflect the expectations of risk managers for plausible economic and financial scenarios. The stress-testing volatility factor of 25% adopted earlier is simply taken as a proof of concept and would require agile calibration to reflect a more realistic situation. Forward-looking correlation matrices could also be derived with another approach than TIC, as, for instance, get the implied correlations by calibrating a model using market prices of correlation-dependent derivatives in same way implied volatilities are obtained from option quotes. This approach has certainly less available data than the TIC approach (which relies on historical data at start), but has the benefit of being forward-looking too.

7 Conclusion and further research

This paper showed promising results from many aspects. First, and conditional to an adequate and robust feature selection process, Long-Short Term Memory (LSTM) and Gated Recurrent Units (GRU) models are able to predict with a good degree of accuracy the spread between the realized volatility and the implied volatility on the next trading day. In particular, the GRU model proved to be more efficient on predicting the label (positive RV-IV spread)

than the LSTM model. Interestingly, Model II volatility classification prediction was not better than Model I, despite the incorporation of forward-looking GARCH volatility estimates or commodity prices into the feature base set. This observation highlights the importance of feature selection and limiting the explanatory variables when building a prediction model. Second, classification prediction models can be used as a support to investment and risk management strategies: an investment strategy can indeed be built around the assumption that a realized volatility above its related implied volatility indicates a bear market; a risk strategy can also be build in factoring the volatility prediction results into the calculation of portfolio risk measures. More particularly, classification prediction models and machine learning techniques proved to be an important support in the design of forward-looking risk management and scenario analysis frameworks.

This paper naturally opens the door to many applications and improvements:

- A first axis to consider would be around the multi-steps ahead classification predictions, where auto-correlation requires to be properly addressed in the model design. The exact same exercise could also be carried over in the case of a regression exercise, e.g. where the objective is to predict a value, not a class.
- A second axis to consider would be the better integration of neural networks' predictions into investment and risk management strategies. These predictions should be factored in properly-calibrated frameworks in order to allow for accurate scenario analysis.
- Last but not least, a third axis to consider would be to use neural networks to directly predict the future (positive semi-definite) correlation or covariance matrix, rather than the return or volatility of each individual asset in the portfolio, such that the machine learning model output could immediately be fed into some risk calculations.

References

[1] Stephen J Taylor. "Asset Price Dynamics, Volatility, and Prediction". Chap. 8. ISBN: 9780691134796.

[2] Stephen J Taylor. "Asset Price Dynamics, Volatility, and Prediction". Chap. 9.3. ISBN: 9780691134796.

[3] ShuiLing Yu and Zhe Li. "Forecasting Stock Price Index Volatility with LSTM Deep Neural Network". 29 (2018). DOI: 10.1007/978-3-319-72745-5_29.

[4] Zacharie Guibert. *Deep learning toolbox for classification prediction models.*

[5] Sepp Hochreiter. "The vanishing gradient problem during learning Recurrent Neural Nets and problem solutions" (1997).

[6] v7 labs. *Activation Functions in Neural Networks, 12 Types Use Cases.* URL: https://www.v7labs.com/blog/neural-networks-activation-functions.

[7] Gregory D. Hager Robert DiPietro. "Deep learning: RNNs and LSTM". *Handbook of Medical Image Computing and Computer Assisted Intervention* (2020).

[8] Zacharie Guibert. "Deep learning toolbox for classification prediction models". Chap. 6. Artificial Neural Networks.

[9] Zacharie Guibert. "Deep learning toolbox for classification prediction models". Chap. 3. Machine learning challenges.

[10] Srivastava and Hinton. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". *Journal of Machine Learning Research 15* (2014).

[11] Xue Ying. "An Overview of Overfitting and its Solutions". *Journal of physics* (2018).

[12] Zacharie Guibert. "Deep learning toolbox for classification prediction models". Chap. 7. Model evaluation and class imbalance.

[13] Zacharie Guibert. "Deep learning toolbox for classification prediction models". Chap. 4. Feature importance.

[14] R.T. Clarke H.R. Bittencourt. "Feature selection by using classification and regression trees (CART)". *Faculty of Mathematics, Pontifícia Universidade Católica do RS, Porto Alegre, Brazil* (2016).

[15] Jukka Iivarinen and Kimmo Valkealahti. "Feature Selection with Self-Organizing Feature Map". *Helsinki University of Technology* (1994).

[16] Witold R. Rudnicki Miron B. Kursa Aleksander Jankowski. "Boruta – A System for Feature Selection". *ICM, University of Warsaw* (2010).

[17] Embrechts McNeil Strautmann. "Correlation and dependency in risk management: properties and pitfalls". *Mathematics department, University of Zürich* (1998).

[18] Marcos López de Prado. "Estimation of theory-implied correlation matrices" (2019). DOI: https://dx.doi.org/10.2139/ssrn.3484152.

[19] Mico Loretan and William B English. "Evaluating changes in correlations during periods of high market volatility". *BIS quarterly review, June 2000* (2000).

[20] Alexander Eydeland Roza Galeeva Jiri Hoogland. "Measuring correlation risk" (2007).

[21] Zacharie Guibert. "Deep learning toolbox for classification prediction models". Chap. 8. Tuning of hyperparameters.

[22] Thijs van den Berg. "Practical machine learning case studies for finance". *CQF cohort 39 - Module 5 Lecture 8* (2022).

[23] Stephen Figlewski. "Forecasting Volatility Using Historical Data". *NYU Stern school of business* (1994).