

A Work Project, presented as part of the requirements of the Award of A Master Degree
in Finance from the NOVA - School of Business and Economics.

Asset Allocation using Machine Learning

Julius Nitz - 30619

A Project carried out on the Master in Finance Program under the supervision of:

Pedro Lameira

04.01.2019

Abstract

This paper, Asset Allocation using Machine Learning, proposes a two step model, forecasting first volatility through an GJR-GARCH model and using a Support Vector machine to do the investment decision between the market portfolio and a risk parity portfolio. Besides the volatility forecast, the Support Vector Machine is based on economic, price, fundamental and sentiment data. It manages to outperform both the market (S&P 500) and a risk parity portfolio in terms of returns and risk adjusted returns.

Keywords: Support Vector Machine, Asset Allocation, Risk Parity, Volatility Forecast

Contents

1 Motivation	3
2 Literature Review	3
2.1 Risk Parity	3
2.2 Machine Learning	5
2.3 Support Vector Machine	5
2.4 GARCH	9
3 Description of Model	12
3.1 Construction of the Risk Parity Portfolio	12
3.2 Input Data for the Support Vector Machine	13
3.3 First Step: Predicting Volatility	16
3.4 Second Step: Support Vector Machine	17
4 Results and Comparison	18
5 Conclusion	20
6 Annex	22
6.1 Tables	22

6.2	Figures	25
-----	-------------------	----

List of Tables

1	Kernel evaluation	22
2	Buy and hold comparison	22
3	Penalty parameter evaluation	23
4	Parameter evaluation	24

List of Figures

1	SVM with linear kernel on a simple data set	25
2	SVM with Gaussian kernel on a simple data set	25
3	SVM with linear kernel on an advanced data set	25
4	SVM with linear kernel son an advanced data set	26
5	GARCH model	26
6	GJR-GARCH model	27
7	GARCH model with Student's t distribution	27
8	GJR-GARCH model with Student's t distribution	27

1 Motivation

Adequate risk management is one of the challenging parts of asset management. Active investors are facing the challenge of deciding on an asset allocation as well as on the timing of their investment. This has been proven very difficult, even for the most sophisticated investors ([Andreu et al., 2018](#)). To reduce miss-allocation and timing, more and more investors have turned to risk parity approaches to optimize their portfolios. Rather than investing fixed portions of the portfolio into an asset class or varying them discrete over time, multiple assets are combined and weighted by their risk contribution to the portfolio. All assets should contribute the same amount of risk. A simple implementation can be done by using the S&P 500 and 10-year US Treasury bonds which yields a higher Info Sharpe Ratio than a traditional 40/60 allocation for bonds and equity ([J.P.Morgan, 2012](#)) or a plain investment in the market portfolio, the S&P 500. Nevertheless, returns are lagging the market portfolio in absolute terms([J.P.Morgan, 2012](#)). Data collected from Bloomberg shows, that the annualized return of the market portfolio was 9.4% from the 20.12.1988. A risk parity portfolio achieved just 5.1% over the same time frame. However, the market portfolio is subject to higher volatility and experiences larger draw downs. Thus, it would be desirable to invest in rising markets into the market portfolio and in less stable times in the more resolute risk-parity portfolio. The rest of this paper is structured as followed: First, a literature review covering risk parity, Machine Learning and Support Vector Machines as well as GARCH models is conducted. Second, the proposed two-step model and the underlying data is presented. Third, results are compared and finally a conclusion is drawn.

2 Literature Review

2.1 Risk Parity

Risk parity is a novel approach in the allocation of assets and a center of debate by academics and practitioners. First mentioned by [Lörtscher \(1990\)](#) and [Kessler and Günter \(1996\)](#), the concept has gained in popularity after the global financial crisis, promising

higher returns at a lower risk level than traditional allocation methods, like the market portfolio following the Capital Asset Pricing Model ([Sharpe, 1964](#)) or a 60/40 allocation between equities and bonds. The novelty of the approach is to allocate capital based on the risk contribution to the portfolio instead of allocating nominal dollar amounts. As a simplification, we can look at a two asset case, taking the U.S. equity market and U.S. Treasury bonds. A risk parity portfolio of these two would significantly invest more in bonds than the other allocation methods as it is less risky. Furthermore, many risk parity strategies require leverage to increase the low returns of an non-levered risk parity strategy to satisfy the return demands of investors ([Anderson et al., 2012](#)).

Both [Asness et al. \(2012\)](#) and [Anderson et al. \(2012\)](#) find, that a risk parity portfolio without leverage outperforms when evaluated on Info Sharpe Ratio traditional allocation methods, even after adjusting for trading costs. [Anderson et al. \(2012\)](#) argues, that after trading costs levered risk parity portfolios would be worse off than the traditional 60/40 allocation. However, [Asness et al. \(2013\)](#) points out to the fact, that Anderson et al. used an odd way to implement risk parity by allowing the target volatility to fluctuate instead of keeping it fixed. Both represent two sides of a controversy about the benefits of risk parity. Important for this paper are their findings and agreement on the unlevered risk parity portfolio and theoretical explanations backing the observed data.

Risk parity portfolios experience a lower absolute return compared to nominal composed portfolios while having a higher Sharpe Ratio. [Frazzini and Pedersen \(2014\)](#) showed, that low risk assets usually experience better risk adjusted returns as investors are restrained in leverage. Thus, investors able and comfortable to use leverage could exploit the benefits of risk parity, gaining higher returns than the market portfolio with a similar risk profile [Asness et al. \(2013\)](#). Furthermore, this also leads to the effect that the risk premium paid for high risk assets is not large enough to offset the increased risk investors are taking. If the equity risk premium would be sufficiently large, a portfolio that is dominated by equity risk, such as a 60/40 portfolio, should perform better than a risk parity portfolio ([Asness et al., 2013](#)).

2.2 Machine Learning

A wide body of research has been produced over the last two decades in machine learning application in financial markets and finance in general. In stock market prediction, a lot of work was done on predicting stock prices and the direction of share price movements. While this offers high financial gains, it is also one of the most complicated tasks due to non-linearity and non-stationariness of the underlying time series (Nair et al., 2017). Much work focused on Neural Networks to predict the market (Gerlein et al., 2016). Examples include Boyacioglu and Avci (2010), Ortega and Khasahanah (2014) and Hsu et al. (2009) which all had promising results. jae Kim (2003) was one of the first to apply a Support Vector Machine (SVM), forecasting the KOSPI while Huang et al. (2005) did similar work on the Nikkei 225. Recently, work focused on also predicting volatility (Donaldson and Kamstra (1997), Chen et al. (2010), Kim and Enke (2016), Hajizadeh et al. (2012), Monfared and Enke (2014), Kristjanpoller et al. (2014)) and using volatility as an input forecasting prices (Kristjanpoller and Minutolo, 2016). However, little work has been done in choosing the optimal investment strategy deciding between assets instead of predicting the direction of a single assets with Nazemi et al. (2014) being an exception solving portfolio optimization problems.

2.3 Support Vector Machine

Support Vector Machines (SVM) are a basic machine learning application for classification task introduced through Vapnik (1995) and Vapnick (1998). In principal, SVM are a classification method which maps the input vectors into a high-dimensional feature space, followed by a construction of a linear model implementing non-linear class boundaries. The aim is to find the optimal hyper-plane which is the maximum margin hyperplane yielding the maximum separation between classes. The closest samples in the training data to the hyperplane are the support vectors used to construct the optimal hyperplane. This procedure offers an effective way to classify data binary with a small sized training sample (Cao et al., 2006). SVM are designed in such a way that they can handle non-linear problems by using linear regression if the dimension of the feature space is high

enough. Furthermore, just a subset, which should represent stable characteristics, of the training data serves as support vectors in the model, giving the technique the name (Chen et al., 2010).

The classification through the optimal hyperplane separates the training data with a maximal margin, yielding the minimum distance from the hyperplane to the closest data points in the training set. Given a data set $\{(X_i, y_i)\}_{i=1}^l$, $X_i \in \mathfrak{R}^n$, $y_i = \pm 1$ where X_i is the n-dimensional input vector and y_i is the class label a SVM approximates a separating hyperplane by

$$f(X) = (W \cdot \phi(X) + b) \quad (1)$$

where $\phi(X)$ is the high dimensional feature space mapped nonlinearly from the input space X

To find the optimal hyperplane in equation(1), the value for the margin is set to $\frac{2}{|W|}$. The maximum margin is achieved by estimating W and b through minimization:

$$\min : C \sum_{i=1}^l \varepsilon_i + \frac{1}{2}|W|^2 \quad (2)$$

subject to

$$y_i(W \cdot \phi(X) + b) \geq 1 - \varepsilon_i \quad , i = 1, \dots, l \quad (3)$$

In equation(2) $C \sum_{i=1}^l \varepsilon_i$ sums the training errors, with C determining the trade-off between training error and margin, while $\frac{1}{2}|W|^2$ maximizes the margin. Both put together are making up this risk function. C is a user-defined parameter, unique to each classification and usually optimized by trial and error. Increasing the C value will reduce the margin accepted by the decision function increasing usually the ability to correctly classify. However, it increases the the complexity of the decision function (Pedregosa et al., 2011). Equation(1) takes the following form after introducing Lagrange multiplies a_i to exploit the optimality constraints:

$$f(X, a_i) = \sum_{i=1}^l a_i y_i K(X_i, X) + b \quad (4)$$

a_i has a positive value and can be obtained by maximizing the Equations 2 and 3 yielding combined the following form:

$$R(a_i) = \sum_{i=1}^l a_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l a_i a_j y_i y_j K(X_i, X_j) \quad (5)$$

which is constrained by

$$\sum_{i=1}^l a_i y_i = 0 \quad , 0 \leq a_i \leq C \quad (6)$$

$K(X_i, X_j)$ is know as the kernel function. Users can determine their own kernel function and the most common ones will be discussed below. Most widely used however is the Gaussian kernel (Cao et al., 2006). The Gaussian kernel is defined as

$$K(X_i, X_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (7)$$

with gamma being a user input, usually tested through trial and error but constrained to $\gamma > 0$. High values of γ mean, that the influence of a single training example has small reach while low values grant a larger reach (Pedregosa et al., 2011).

The linear kernel is given by

$$K(X_i, X_j) = (x_i - x_j) \quad (8)$$

The polynomial kernel is defined as

$$K(X_i, X_j) = (\gamma(x_i - x_j) + r)^d \quad (9)$$

d , r and γ are chosen by the user. d defines the degree of the polynomial function and varying this affects the result

The sigmoid kernel, given by

$$K(X_i, X_j) = \tanh(\gamma(x_i - x_j) + r) \quad (10)$$

with γ and r are again user inputs.

While the linear kernel dates back to the original work by [Vapnik \(1995\)](#), the other kernels tend to perform better to map higher dimensional feature spaces ([Chen and Shih, 2006](#)). To illustrate this effect, [Figures 1 to 4](#) display different data sets and each will be classified with a linear and a Gaussian kernel. Choosing a more sophisticated kernel might improve the precision of the classification model, depending on the data. However, a Gaussian kernel demands more computational power and needs more adjustments to work. Offering more input options to the user makes this more prone to overfitting than a linear model([Raschka, 2016](#)). Besides the danger of overfitting, researchers should keep in mind, that the best suited kernel depends on the properties of the data set, but also involves a trade off between precision and computational resources ([Sharma et al., 2016](#)).

[Figure 1](#) displays a simple data set and how a linear kernel would separate it. Note, that a two dimensional data set is chosen for illustration, the dimension has no impact on the performance of the kernel per se. The linear kernel is able to separate the data set in a good manner. Comparing the output with [Figure 2](#) below, we notice no material difference in the classification using a Gaussian kernel. The only thing changing is the colouring of some areas, so outliers in the bottom right corner might be attributed to a different category in a forecast. As the Gaussian kernel requires more computational power, there might be no need to use it in this case.

However, if we now examine a more sophisticated data set as shown in [Figure 3](#) and [Figure 4](#), it becomes obvious that this data set can not be separated linear, so a linear kernel would do a bad job. In this case it makes more sense to use a Gaussian kernel as showcased in [Figure 4](#). While there is no solution to find the best fitting kernel, expect testing for them, [Raschka \(2016\)](#), who also created the figures, advises as a rule of thumb to use linear kernels for linear problems and non-linear kernels such as the Gaussian kernel for non-linear problems.

So far, SVM for binary classification was examined. And while the presented model is dealing with a binary classification problem, the following provides a brief overview of

further developed models. The basic SVM model was developed into multi-class support vector machines (**MSVM**). According to [Kim and Ahn \(2012\)](#), there are two fundamental ways of approaching this problem. Either, several binary classifiers can be constructed or all data is considered at once. For the first approach, they propose four techniques, namely one-against-all, one-against-one ([Krebel, 1999](#)), directed acyclic graph SVM (**DAGSVM**) ([Platt et al., 2000](#)) and Error-correcting output codes ([Dietterich and Bakiri, 1995](#)).

One-against-all is the simplest method, creating n binary SVM classifiers for a n -classification. First, class 1 is separated from all other classes, second class 2 is separated against all other class till the n -class is separated against the other classes. When One-against-one is used to create a MSVM, binary classifiers for all pairs of classes are created, yielding in $C_{1v1}=n(n-1)/2$ classes.

DAGSVM is similar to one-against-one in the training phase but uses a rooted binary decision directed acyclic graph with N classifiers which resembles a decision tree. In every step, a binary option is given, either it is not a or not b . Error-Correcting Output Coding, with origins from digital communication theory, creates a matrix comparing attributes for classes with classifiers. The row, which represents the classes, with the lowest error is selected as class.

The second approach is to consider all data at once for which [Crammer and Singer \(2000\)](#) and [Weston and Watkins \(1999\)](#) provided methods based on quadratic optimization problems. These are very complex and require more computational power, therefore they are utilized less often than other methods ([Kim and Ahn, 2012](#)).

2.4 GARCH

Heteroscedasticity in time series of financial markets instrument pose a fundamental problem for the implementation of model based approaches for investing. With Volatility varying over time, it has to be forecasted before it could be used as an input in a model, as using historical averages might lead to a wrong result. Thus forecasting volatility has kept academics and practitioners occupied. A comprehensive review was created by [Franses and McAleer \(2002\)](#). The dominant model is the GARCH model developed by [Bollerslev](#)

(1986), which is a generalisation of the ARCH by Engle (1982). GARCH based models have the convenient properties to adapt to volatility clustering and varying volatility over time, meaning that a time series experiences periods of higher and lower volatility. To model volatility, two effects, moving average and autoregression of the error term, are captured. A GARCH model is given by the following formulas.

$$y_t = x_t' b + \epsilon_t \quad (11)$$

$$\epsilon_t | \psi_{t-1} \sim \mathcal{N}(0, \sigma_t^2) \quad (12)$$

$$\sigma_t^2 = \omega + \alpha_1 \epsilon_{t-1}^2 + \dots + \alpha_q \epsilon_{t-q}^2 + \beta_1 \sigma_{t-1}^2 + \dots + \beta_p \sigma_{t-p}^2 = \omega + \sum_{i=1}^q \alpha_i \epsilon_{t-i}^2 + \sum_{i=1}^p \beta_i \sigma_{t-i}^2 \quad (13)$$

In (13), the first sum captures the autoregressive ARCH effect with the order of q while the second sum captures the GARCH effect with the order of p , which means it captures the moving average.

While the GARCH model experienced multiple further developments, this paper limits itself to the GJR-GARCH developed by Glosten et al. (1993). This model has the same properties as a normal GARCH, but adds a conditional indicator term. This helps to model the error term if it is not evenly distributed but skewed to one side. In financial markets, practitioners often experience higher levels of volatility in decreasing markets while appreciating markets are usually associated with lower volatility. A GJR-GARCH is able to capture these asymmetric effects. A GJR-GARCH is given by

$$\sigma_t^2 = K + \delta \sigma_{t-1}^2 + \alpha \epsilon_{t-1}^2 + \phi \epsilon_{t-1} I_{t-1} \quad (14)$$

where

$$I_{t-1} = 0 \text{ if } \epsilon_{t-1} \geq 0, \text{ and } I_{t-1} = 1 \text{ if } \epsilon_{t-1} < 0 \quad (15)$$

Evaluation of appropriate Model For the task of forecasting volatility in financial markets time series, plenty of models and even more variations of these models are available. As GARCH models are based on a log-likelihood estimator, R^2 is no longer a reliable indicator on the performances of a model. Two models are usually used to

evaluate the performance of the model, first the Bayesian information criterion (**BIC**) (Schwarz, 1978) and second the Akaike information criterion (**AIC**) (Akaike, 1974). Both evaluate the fit of a model but penalize overfitting by adding more variables to explain dependent variable. In both cases, the model with the lowest value is preferred. The **BIC** developed by Schwarz (1978) is defined as:

$$BIC = \ln(n)k - 2\ln(\hat{L}) \quad (16)$$

where n is the number of observations, k is the number of parameters estimated by the model and \hat{L} is the maximized value of the likelihood function. According to Giraud (2014) the **BIC** has two limitations, first, it is necessary that the sample size n is significantly larger than the number of parameters k . Second, it can not handle complex models in high-dimensions.

The Akaike criterion is similar to the **BIC** and formally given by:

$$AIC = 2k - 2\ln(\hat{L}) \quad (17)$$

The **AIC** is similar to the **BIC**, as we can see from the similarity of the formula, but differentiating in the penalty given to the numbers of parameters k included in the model. The **AIC** penalty is $2k$ while the **BIC** is $\ln(n)k$. The **BIC** is based on Bayesian. According to Verize (2012), the **BIC** tends to select the true model if the true model is amongst the evaluated models by assigning it the lowest value if the number of observation is asymptotically large. the **AIC** does not perform as good under such circumstances- However, it will choose the model that is closest to the real model if the true model is not evaluated. As in the real world the true model is usually not found, the **AIC** tends to perform better. It also offers the minimization of the risk to pick a wrong model in a finite sample. However, a researcher should always consider both evaluation methods to attribute the different behaviours of both criterions.

3 Description of Model

In this paper, we present a two-step approach to decide whether to invest in a risk parity portfolio or the market portfolio, represented by the S&P 500. This decision depends on the future risk assessment. As risk in the financial world is usually assessed by volatility of the underlying asset, the presented model emphasizes on volatility. As the first step of the model, volatility is forecasted using an GJR-GARCH model to capture its properties. In the second step, the volatility forecast is used along with return data for equities and government bonds, sentiment data and economic indicator to tackle the binary decision to invest in one of the portfolios in the following period. These inputs are used in a Support Vector Machine

Structuring the investment decision in such a binary, simplified question eases the decision making process and enhances the ability of machine learning tools to support it. While there is a large body of research predicting market directions and returns, this is a new approach. Furthermore, the combination of fundamental data and price related data is different as most researchers were just using one of the two.

3.1 Construction of the Risk Parity Portfolio

The presented model, a Support Vector Machine, is trained to decide between investing in the S&P 500 or a risk parity portfolio in the following period. The risk parity portfolio is constructed through combining 10 year US treasuries as a bond investment and the S&P 500 as an equity investment. For 10-year treasuries, the rolling future with the corresponding maturity was used as a proxy for price data while the returns of the S&P 500 Total return index were used as the equity investment, mirroring an investment in a portfolio tracking the market and receiving dividends. The weights of the two assets are determined by their relative contribution to the overall desired risk level. This level was arbitrarily set at 5% annual volatility. Changing this does not influence the performance of the portfolio materially.

The relative contribution to the risk is set as their annualized daily volatility over the last 250 trading days, which approximates one year. The annual volatility of each asset

is afterwards divided by the target volatility for the overall portfolio. If the sum of both weights exceeds 100%, they are standardized to sum up to 100% to prevent the use of leverage. The weights w_i for this two asset case is thus given by:

$$w_i = \frac{\sigma_T}{\sigma_i} \quad i = 1, 2 \quad (18)$$

If $w_1 + w_2 > 1$:

$$w_i = \left(\frac{\sigma_T}{\sigma_i}\right)/(w_1 + w_2) \quad (19)$$

with σ_T being the targeted volatility of the portfolio and σ_i being the individual volatility of the asset. The weights of the risk parity portfolio are adjusted on every trading day. Of both, the S&P500 Total Return Index and the risk parity, daily log returns are calculated. Based on this, a target vector is composed, being filled with 1, if the next day return of the S&P500 Total Return Index is larger than the risk parity portfolio, or 0 if this is not the case. Predicting a 1 or a 0 is the aim of the model, thus compromising the investment decision. This classification is done through a Support Vector Machine (SVM).

3.2 Input Data for the Support Vector Machine

16 different input series are used for the prediction, combining price data, sentiment indicators, economic data, valuation of the market, technical indicators and the aforementioned volatility forecast. The first group contains returns of the S&P 500, the 10-year Treasury future and two currencies, the Australian Dollar and the Japanese Yen.

For the S&P 500 a non-dividend-adjusted time series is used to ensure that no noise is generated through the payment of dividends. Three different time horizons are used, the returns over the current period, the last 5 periods, equalling a normal trading week, and the last 50 periods, equalling 10 weeks of trading are fed into the algorithm. This is done to represent short, medium and long term returns investors are observing, enriching the value of information given to the algorithm.

For the Treasury bond future, the return over the current period and the last 50 days is used. The longer time horizon to calculate the return for treasuries is chosen in this

way as bond markets tend to be less volatile and move slower. Thus capturing a specific, meaningful trend requires a longer time horizon. The two currencies, the Australian Dollar and the Japanese Yen, are both feed in with the return over the current period. While the former was picked to represent a proxy on a commodity based economy, the later serves as a risk proxy and safe haven asset in financial markets.

Two sentiment indicators are also used in the model, the Credit Suisse Fear Barometer and the Summation Index. The Credit Suisse Fear Barometer ([Hoffman, 2017](#)) is comparing the income of selling a 10% out-of-the-money call to the price of a put that can be financed with this premium. The more expensive the puts are, the higher the index as just a put more out of the money could be brought. Thus a high index, reading the %OTM of the put strike, signals higher insurance cost and thus nervousness in the market. The Summation index was developed and is published by [McClellan \(2019\)](#). It is the sum of all previous McClellan Oscillators and indicates over the long term the trending direction of the stock market and its power, with a reading of 1000 to be considered neutral. The McClellan Oscillator is calculated as the difference between two exponentially moving averages of the Advance-Dcline Line, one with a 10% smoothing constant and one with a 5% smoothing constant. The Advance-Dcline line is calculated daily by adding the number of rising stocks and subtracting the number of declining stocks of the previous days reading. An increase in the advance-Dcline line and thus over the long term in the McClellan Oscillators is seen as a positive signal as investors are buying in the market, investing fresh money. However, extreme readings might also identify overbought or oversold markets ([McClellan, 2019](#)).

For economic indicators, three variables were selected. First, the latest ISM PMI ([PMI](#)) reading was included, which gauges confidence of the business community and offers an outlook into the future. A reading above 50 signals expansion while a reading below 50 is interpreted as contraction. The number is published every month, so it is a comparably less volatile time series in the model. Second, the latest reading of the US hourly wage growth is included which is published on the first friday of every month by the Bureau of Labor Statistics by the US Department of Labor ([BLS](#)). Finally, the Libor future

maturing one year ahead is include, a contract also known as Eurodollar. A Eurodollar contract offers the buyer to borrow money at maturity for 90 days at the traded rate. The longer maturity of the future is picked to gain some insight into the market expectations for the risk free rate in one-years time. As the level of the risk free rates varied over time and is not fixed, participants in financial markets are more interested in changes in the rate than the actual level. Thus, we calculate the 125 day return of the future, implying the change in market expectations over the last half year of trading. As a valuation gauge of the equity market, the rolling 12 month forward Price Earnings ratio was used. The forward ratio takes into account market expectations and carries more information than a backward looking valuation method. It is calculated by Bloomberg as the Consensus of all financial analyst contributing estimates to Bloomberg.

The first technical indicator is the Relative Strength Index (**RSI**) value of the S&P 500, calculated by Bloomberg. It indicates whether the market is overbought or oversold at current levels. The RSI was developed by [Wilder \(1978\)](#) and is defined as:

$$RSI = 100 - \frac{100}{1 + RS} \quad ; \quad RS = \frac{\bar{UP}}{\bar{D}} \quad (20)$$

with \bar{UP} being the sum of up closes of the market over the last 14 days divided by 14 and \bar{D} being the sum of down closes of the market over the last 14 days divided by 14. The number of days observed can be varied to derive different RSIs, but as we just use one in this model we decided to stick with the original calculation method. The second technical indicator is binary in nature. It is constructed to indicate, whether the current index reading of the S&P 500 is above or below its 50 day simple moving average. If this is the case, a 1 indicates yes while otherwise a 0 indicates no. Practitioners attribute much weight to the 50 day moving average and see it as an important indicator of the trending of a market. Furthermore, due to it binary nature, it does not have to be preprocessed before being used in the model. The last input into the model is the volatility forecast for the next period obtained from the GJR-GARCH model presented in section 2.3. A more detailed overview how this model was picked is shown in section 3.3, presenting the first step of the model.

As it is common in the literature, all data is preprocessed to be in a numerical range appropriate for SVMs. This is done by normalizing all values into the so called Z-score, given by:

$$Z_i = \frac{X_i - \bar{X}}{\sigma_X} \quad (21)$$

where X_i is the given value, \bar{X} is the mean of the time series and σ_X is the standard deviation of the time series. This process scales the data into a normal distributed series with a mean of 0 and a variance of 1. The normalisation should be done separately for the training and test data to avoid forward looking bias.

3.3 First Step: Predicting Volatility

As the first step of the model, we estimate the volatility one period ahead through a GJR-GARCH model based on the work of [Glosten et al. \(1993\)](#). For this, the model has been configured on daily data since the 01.01.1970 with 7249 observations till the 14.10.1997. For the next 5490 observations till the 30.10.2018, a sliding window forecast was used. This enables the model to update and adjust to more recent data and represents a more realistic forecasting behaviour, as any forecaster would have used all available data till that point in time. A larger sliding window is chosen as a balancing act. The model should be allowed to incorporate new data and changes in the underlying true behaviour of the time series. However, a large sample also enables us to estimate the true behaviour better than a small sample. This is especially important as we want to observe volatility shocks, which are not evenly distributed through time. Furthermore, we want to cover multiple cycles, both in the economy as in the financial market, to find the true behaviour. A short data sample might be skewed and overfitted just to one specific market period.

Using the same data set, four different models were evaluated to find the best fitting one. A normal GARCH model was estimated first, using standard parameters. Following that, a GJR-GARCH model was used to test on asymmetric volatility distribution which is often observed in financial markets. Both models were also tested with a different distribution of error terms. Instead of assuming a normal distribution, the Student's t distribution was used to fit the model. Figures 5, 6, 7 and 8 are showing the statisti-

cal summaries of the models mentioned above. First, we can see from the the GARCH model is significant as all parameters would not be rejected at the 95% confidence interval. Both, the moving average and autocorrelation effects are significant, although the autocorrelation expressed through β overweights. Using both BIC and AIC, we can assume that volatility shocks are not evenly distributed and rather occur in declining markets as both criterions are smaller in the GJR-GARCH model than the GARCH model. Both evaluations methods penalize the additional estimator but still improve.

In a second step, the two models with an assumed Student's t distribution are evaluated (Bollerslev, 1987). Again, we observe that all parameters are significant, indicating that the model is suited to forecast volatility. A strong autocorrelation effect is observed again in both models. The GARCH model also is better fitted than the previous two models assuming normal distribution. However, the GJR-GARCH is again slightly better in both, BIC and AIC. Thus we assume that error terms are not normal distributed but are experiencing fatter tails. As the GJR-GARCH with the Student's t distribution is best fitted to model volatility from the examined models, it is picked to forecast volatility going forward.

3.4 Second Step: Support Vector Machine

Due to data availability, the first date all above mentioned parameters are available is the 17.10.1997, marking the start of our data sample. It includes all data till the 30.10.2018, thus yielding a long-term data set including bull and bear markets, most notably the dotcom bubble crash and the Global Financial Crisis of 2008. As it is common in the literature, 80% of the data set is used to train the SVM while 20% is used to test it. Thus, our testing period begins on the 9.09.2014. The SVM was implemented using the scikit-learn (Pedregosa et al., 2011) package for python, available on their website. Different kernels and penalty parameters were tested, for which results are presented in the following section. All other inputs were left on default of the package.

Results are evaluated by precision, recall and the F1-Score. Precision is defined as follow:

$$p = \frac{tp}{tp + tf} \quad (22)$$

with tp being true positives and tf being true negatives. Intuitively speaking, it measures the ability of the classifier to not label a neagive value as positive. Recall on the other hand measures the ability to find all true values and is defined as:

$$r = \frac{tp}{tp + fn} \quad (23)$$

where fn are false negatives. The F1 score is the harmonic mean of these two measures, calculated as:

$$F_1 = 2 \cdot \frac{p \cdot r}{p + r} \quad (24)$$

Besides these standard measures to evaluate a classifier model, this paper also puts an emphasize on evaluation method used by practitioners, the Info Sharpe Ratio, defined as:

$$SR_i = \frac{r_i}{\sigma_i} \quad (25)$$

where r_i is the annualized return of the forecast and σ_i is the annualized volatility of the forecast. In doing so, the real life performance over the testing period of the model is evaluated. Practitioners use this ratio to evaluate an investment strategy as it puts returns into comparison with the risk, described as volatility, of the strategy. Thus, even a low yielding strategy might achieve a better Info Sharpe Ratio if volatility is minimized. This does not constrain sophisticated investors with access to leverage.

4 Results and Comparison

In the following, results of testing different setups of the model are presented. In this binary classification, the decision was made between investing in the market portfolio, represented by a 1, and investing in the risk parity portfolio, represented by a 0. First, we ran the model with all parameters described above and evaluate them in the testing range, using different kernels to find the best fitting for the underlying data. Results are

summarized in Table 1.

We can interfere that the Gaussian kernel is best suited for our data set. It already shows how important it is to extent the scope of evaluation beyond just traditional classifier measures. While the difference in precision and F1 scores is minimal, the difference in Info Sharpe Ratio is not. It stands out, that the Gaussian kernel has the highest sharpe ratio and the highest recall value for 0, indicating an investment in the risk-parity portfolio. The further discussion will thus just involve a SVM using the Gaussian kernel.

Following that, first we test for different values of the penalty parameter C as given in equation (2) and than evaluate whether all parameters used as inputs do increase precision of predictions. For the variation of the penalty parameter, a value of 1 is selected first and than varied to find the best parameter with results presented in Table 3. Increasing the penalty values increases both precision and recall values as well as the Info Sharpe Ratio till a maximum is found for a value of $C = 8$. This kernel specification yields a Sharpe ratio of 2.21. Again a high recall value for the risk-parity classification is observed.

Finally, the different input parameters are each evaluated. Thus, they are compared with the model being run without this parameter. All use the same kernel specification that have been found optimal from above, results are presented in Table 4. Leaving out a parameter decreases the performance of most forecasts. Only three variables do not meaningful increase the performance of the model and are thus left out. For the Summation index, a small third digit increase in the Info Sharpe Ratio is recognized, leading to the conclusion that the presence of the data does not improve the forecast materially. The same is consequently observed for the Japanese Yen. While leaving both currencies, including the Australian Dollar, out, decreases the Sharpe Ratio, the presence of the Yen does not improve the forecast. However, the most meaningful increase is achieved by leaving out the 50 day bond market return. This increases the Info Sharpe Ratio to 2.28. This marks the final model with the highest sharpe ratio achieved, using a subset of 13 input parameters.

Noteworthy is also the effect of leaving out the Volatility forecast as it yields the worst performance observed. This highlights the importance of the two step model proposed in

this paper. However, it should be noted, that the forecast would have still beaten both the market portfolio and the risk parity portfolio when evaluated using the Sharpe ratio. The market portfolio achieved in this time an annualized return of 8.8% with an annualized volatility of 12.8%, yielding a Info Sharpe Ratio of 0.684, as summarized in Table 2. This compares with 2.7% return and 3.78% volatility for the risk parity portfolio which had a Info Sharpe Ratio of 0.722. It should be noted that the test period experienced an exceptionally high Info Sharpe Ratio for the market portfolio, as volatility was subdued. For the whole sample the Sharpe ratio was just 0.349 compared to 0.77 for the risk parity portfolio. Also, a 60/40 allocation would have had a better Sharpe ratio in the test than in the full sample, but still be below the risk parity portfolio. Overall, even a not well set up model seems to offer investors insights and upside potential compared to a buy and hold strategy. Nevertheless, the opportunities of fine tuning a model, clearly shown by the increases in Sharpe ratio observed, offers huge gains and should always be conducted.

5 Conclusion

In this paper we established a two step model to help investors with the crucial decision whether to invest in a risk parity portfolio or the higher returning market portfolio. We found a Support Vector Machine well suited for this classification task, especially if it is supported by a volatility forecast done before. Furthermore, the combination of economic, sentiment and price related indicators proved beneficial. This showcases, that a comprehensive approach to investing, factoring in various data sources, should be pursued by investors. In evaluating a classification model for financial markets, it is important not just to evaluate standard metrics such as precision but also take the Info Sharpe Ratio into account to find profitable strategies. After all, the aim of a good investment model should be to catch the most extreme moves of the invested assets, in both directions. Thus, even low scores on traditional evaluation methods can be suitable for investors as shown in this paper.

Overall, this shows that proper risk management is key for asset management. Alternating and timing the investment of a risky and less risky assets proves beneficial and

machine learning tools are able to identify the underlying patterns in the data. Furthermore, this solves problems raised in previous literature of low returns by just investing in a low risk assets without leverage. The presented model offers investors an opportunity to enjoy high Sharpe ratio investments at a high rate of return. Nevertheless, more work should be done in this area. This paper limited its scope to a binary classification problem. Future work might focus on a multivariate classification case to investigated, whether this provides even better investment opportunities. While SVM were intended as a binary classification tool, they can be adjusted to a multivariate case. The second limitation of this paper is regional scope, as just the U.S. market was investigated. Other geographies might be the scope of future work as well as other asset classes. Finally, other researchers might be inclined to do more work on allocating assets rather than forecasting price movements, as this would be more beneficial for large investors which are forced to invest their money somewhere and often suffer from limitations of leverage and short selling.

6 Annex

6.1 Tables

Kernel		Correct/False	Precision	Recall	F1 -	SR	AR
Linear	1	62/461	0.52	0.42	0.47	1.06	12.1%
	0	48/502	0.53	0.63	0.58		
	Total	110/963	0.53	0.53	0.53		
Gaussian	1	220/303	0.53	0.42	0.47	1.47	15.0%
	0	195/335	0.54	0.65	0.54		
	Total	415/658	0.53	0.53	0.53		
Sigmoid	1	241/282	0.49	0.46	0.47	0.80	7.7%
	0	251/299	0.51	0.54	0.53		
	Total	492/581	0.51	0.54	0.53		
Polynomial	1	246/277	0.5	0.47	0.48	0.87	8.7%
	0	249/301	0.52	0.55	0.53		
	Total	495/578	0.51	0.51	0.51		

Table 1: Kernel evaluation

Allocation strategy	Annualized Return	Annualized Volatility	SR
Market Portfolio - full sample	6.6%	18.8%	0.349
Market Portfolio - test sample	8.8%	12.8%	0.684
60/40 Portfolio- full sample	5.22%	10.8%	0.484
60/40 Portfolio - test sample	5.4%	7.3%	0.548
Risk Parity Portfolio- full sample	3.8%	4.9%	0.770
Risk Parity Portfolio - test sample	2.7%	3.7%	0.722

Table 2: Buy and hold comparison

Gamma		Correct/False	Precision	Recall	F1	SR	AR
1	1	220/303	0.53	0.42	0.47	1.47	10.2%
	0	195/335	0.54	0.65	0.54		
	Total	415/685	0.53	0.54	0.53		
0.5	1	169/354	0.5	0.32	0.39	0.99	10.5%
	0	166/384	0.52	0.7	0.6		
	Total	335/738	0.51	0.52	0.5		
3	1	234/289	0.53	0.45	0.48	1.76	16.5%
	0	211/339	0.54	0.62	0.58		
	Total	445/628	0.53	0.53	0.53		
5	1	237/286	0.55	0.45	0.5	2.09	19.6%
	0	191/359	0.56	0.65	0.6		
	Total	428/645	0.56	0.56	0.55		
6	1	226/297	0.55	0.43	0.49	2.10	19.8%
	0	182/368	0.55	0.67	0.61		
	Total	408/665	0.55	0.55	0.55		
7	1	228/295	0.56	0.44	0.49	2.11	19.7%
	0	181/369	0.56	0.67	0.61		
	Total	409/664	0.56	0.56	0.55		
8	1	228/295	0.56	0.44	0.49	2.21	20.5%
	0	181/369	0.56	0.67	0.61		
	Total	409/664	0.56	0.56	0.55		
9	1	229/294	0.55	0.44	0.49	2.02	18.8%
	0	189/361	0.55	0.66	0.6		
	Total	418/655	0.55	0.55	0.54		
10	1	228/295	0.54	0.44	0.48	1.80	17.1%
	0	192/358	0.55	0.65	0.6		
	Total	420/653	0.55	0.55	0.54		
15	1	226/297	0.54	0.43	0.48	1.73	16.8%
	0	194/356	0.55	0.65	0.59		
	Total	420/653	0.54	0.54	0.54		

Table 3: Penalty parameter evaluation

Gamma		Correct/False	Precision	Recall	F1	SR	AR
No Volatility	1	210/313	0.51	0.4	0.45	0.91	10.3%
	0	199/351	0.53	0.64	0.58		
	Total	409/664	0.52	0.52	0.52		
No CSFB	1	199/324	0.56	0.38	0.45	2.02	18.8%
	0	156/394	0.55	0.72	0.62		
	Total	355/718	0.55	0.55	0.54		
No Eurodollar	1	231/292	0.54	0.44	0.49	1.57	15.5%
	0	196/354	0.55	0.64	0.59		
	Total	427/646	0.54	0.55	0.54		
No Summation Index	1	251/272	0.54	0.48	0.51	2.22	18.9%
	0	213/337	0.55	0.61	0.58		
	Total	464/609	0.55	0.55	0.55		
No Wage growth	1	246/277	0.52	0.47	0.49	1.78	15.2%
	0	230/320	0.54	0.58	0.56		
	Total	476/597	0.53	0.53	0.53		
No PMI	1	264/259	0.53	0.5	0.52	1.81	16.1%
	0	235/315	0.55	0.57	0.57		
	Total	498/574	0.54	0.54	0.54		
No P/E	1	194/329	0.53	0.37	0.44	1.68	16.8%
	0	174/376	0.53	0.68	0.6		
	Total	368/705	0.53	0.53	0.53		
No Currencies	1	264/259	0.53	0.5	0.52	1.83	16.2%
	0	232/318	0.55	0.58	0.56		
	Total	496/577	0.54	0.54	0.54		
No JPY	1	255/268	0.54	0.49	0.51	2.22	19.4%
	0	215/335	0.56	0.61	0.58		
	Total	470/603	0.55	0.55	0.55		
No AUD	1	264/259	0.53	0.5	0.52	1.83	16.2%
	0	232/318	0.55	0.58	0.56		
	Total	496/577	0.54	0.54	0.54		
No Moving Average	1	258/265	0.54	0.49	0.51	1.84	16.5%
	0	224/326	0.55	0.59	0.57		
	Total	482/591	0.54	0.54	0.54		
No 5d return	1	251/272	0.52	0.48	0.5	1.83	15.6%
	0	230/320	0.54	0.58	0.56		
	Total	481/592	0.53	0.53	0.53		
No 50d return	1	245/278	0.53	0.47	0.5	2.06	18.5%
	0	216/334	0.55	0.61	0.57		
	Total	461/612	0.54	0.54	0.54		
No RSI	1	239/284	0.54	0.46	0.49	1.93	17.1%
	0	206/344	0.55	0.63	0.58		
	Total	445/628	0.54	0.54	0.54		
No 50d bond return	1	240/283	0.54	0.46	0.5	2.28	20.3%
	0	201/349	0.55	0.63	0.59		
	Total	441/632	0.55	0.55	0.55		

Table 4: Parameter evaluation

6.2 Figures

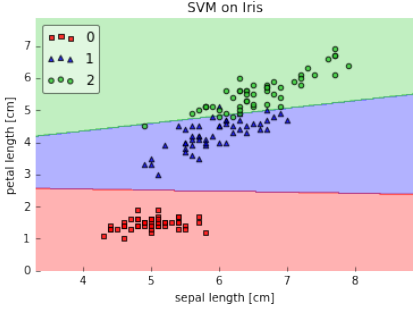


Figure 1: SVM with linear kernel on a simple data set

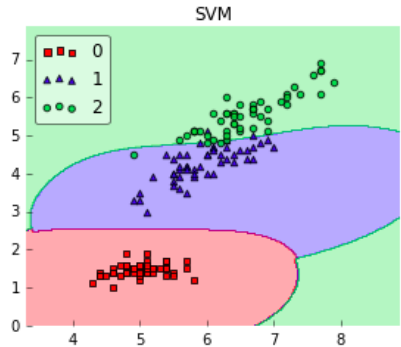


Figure 2: SVM with Gaussian kernel on a simple data set

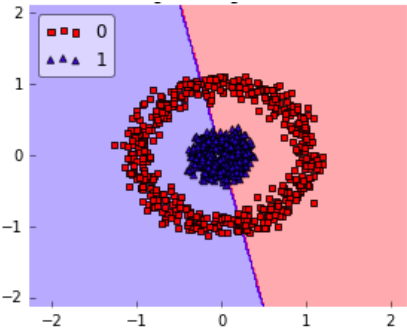


Figure 3: SVM with linear kernel on an advanced data set

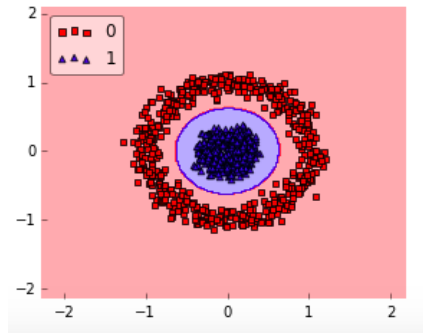


Figure 4: SVM with linear kernel son an advanced data set

```

Constant Mean - GARCH Model Results
=====
Dep. Variable:          SPX Index    R-squared:              -0.000
Mean Model:            Constant Mean  Adj. R-squared:         -0.000
Vol Model:             GARCH         Log-Likelihood:         -8700.05
Distribution:          Normal       AIC:                   17408.1
Method:               Maximum Likelihood  BIC:                   17435.6
No. Observations:     7249
Date:                 Tue, Dec 04 2018  Df Residuals:          7245
Time:                 22:32:56      Df Model:              4
Mean Model
=====
coef  std err      t    P>|t|     95.0% Conf. Int.
-----+-----
mu    0.0478    9.247e-03    5.171  2.329e-07  [2.969e-02,6.594e-02]
Volatility Model
=====
coef  std err      t    P>|t|     95.0% Conf. Int.
-----+-----
omega 7.7929e-03  2.665e-03    2.924  3.452e-03  [2.570e-03,1.302e-02]
alpha[1] 0.0572  1.614e-02    3.544  3.937e-04  [2.556e-02,8.882e-02]
beta[1] 0.9342  1.643e-02   56.860  0.000     [ 0.902, 0.966]
=====
Covariance estimator: robust

```

Figure 5: GARCH model

```

Constant Mean - GJR-GARCH Model Results
=====
Dep. Variable:          SPX Index      R-squared:              -0.000
Mean Model:            Constant Mean  Adj. R-squared:         -0.000
Vol Model:             GJR-GARCH      Log-Likelihood:         -8659.89
Distribution:          Normal         AIC:                    17329.8
Method:               Maximum Likelihood BIC:                    17364.2
No. Observations:    7249
Date:                 Tue, Dec 04 2018  Df Residuals:          7244
Time:                 22:34:21         Df Model:                5
Mean Model
=====
coef  std err      t      P>|t|     95.0% Conf. Int.
-----+-----
mu    0.0359  8.941e-03    4.018  5.876e-05 [1.840e-02,5.345e-02]
Volatility Model
=====
coef  std err      t      P>|t|     95.0% Conf. Int.
-----+-----
omega 8.6657e-03 3.007e-03    2.882  3.955e-03 [2.772e-03,1.456e-02]
alpha[1] 0.0250  5.469e-03    4.580  4.651e-06 [1.433e-02,3.577e-02]
gamma[1] 0.0571  1.937e-02    2.949  3.188e-03 [1.916e-02,9.508e-02]
beta[1] 0.9363  1.408e-02   66.509  0.000      [ 0.909, 0.964]
=====
Covariance estimator: robust

```

Figure 6: GJR-GARCH model

```

Constant Mean - GARCH Model Results
=====
Dep. Variable:          SPX Index      R-squared:              -0.000
Mean Model:            Constant Mean  Adj. R-squared:         -0.000
Vol Model:             GARCH         Log-Likelihood:         -8495.39
Distribution:          Standardized Student's t AIC:                    17000.8
Method:               Maximum Likelihood BIC:                    17035.2
No. Observations:    7249
Date:                 Tue, Dec 04 2018  Df Residuals:          7244
Time:                 22:32:10         Df Model:                5
Mean Model
=====
coef  std err      t      P>|t|     95.0% Conf. Int.
-----+-----
mu    0.0441  8.168e-03    5.398  6.739e-08 [2.808e-02,6.010e-02]
Volatility Model
=====
coef  std err      t      P>|t|     95.0% Conf. Int.
-----+-----
omega 6.1327e-03 1.545e-03    3.969  7.205e-05 [3.105e-03,9.161e-03]
alpha[1] 0.0450  5.190e-03    8.667  4.434e-18 [3.486e-02,5.524e-02]
beta[1] 0.9471  6.047e-03   156.621  0.000      [ 0.935, 0.959]
Distribution
=====
coef  std err      t      P>|t|     95.0% Conf. Int.
-----+-----
nu    6.6099    0.581    11.372  5.750e-30 [ 5.471, 7.749]
=====
Covariance estimator: robust

```

Figure 7: GARCH model with Student's t distribution

```

Constant Mean - GJR-GARCH Model Results
=====
Dep. Variable:          SPX Index      R-squared:              -0.000
Mean Model:            Constant Mean  Adj. R-squared:         -0.000
Vol Model:             GJR-GARCH      Log-Likelihood:         -8477.11
Distribution:          Standardized Student's t AIC:                    16966.2
Method:               Maximum Likelihood BIC:                    17007.6
No. Observations:    7249
Date:                 Tue, Dec 04 2018  Df Residuals:          7243
Time:                 22:31:41         Df Model:                6
Mean Model
=====
coef  std err      t      P>|t|     95.0% Conf. Int.
-----+-----
mu    0.0385  8.167e-03    4.712  2.458e-06 [2.247e-02,5.449e-02]
Volatility Model
=====
coef  std err      t      P>|t|     95.0% Conf. Int.
-----+-----
omega 6.7850e-03 1.791e-03    3.788  1.520e-04 [3.274e-03,1.030e-02]
alpha[1] 0.0253  4.206e-03    6.010  1.853e-09 [1.704e-02,3.353e-02]
gamma[1] 0.0437  9.310e-03    4.699  2.615e-06 [2.550e-02,6.199e-02]
beta[1] 0.9443  6.934e-03   136.189  0.000      [ 0.931, 0.958]
Distribution
=====
coef  std err      t      P>|t|     95.0% Conf. Int.
-----+-----
nu    6.8751    0.624    11.013  3.306e-28 [ 5.652, 8.099]
=====
Covariance estimator: robust

```

Figure 8: GJR-GARCH model with Student's t distribution

References

- [employment situation summary]. URL <https://www.bls.gov/news.release/empsit.nr0.htm>.
- [ism manufacturing pmi]. URL <https://www.instituteforsupplymanagement.org/ISMReport/MfgROB.cfm?SSO=1>.
- H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974.
- R. M. Anderson, S. W. Bianchi, and L. R. Goldberg. Will my risk parity strategy outperform?. *Financial Analysts Journal*, 68(6):75 – 93, 2012.
- L. Andreu, J. C. Matallin-Saez, and J. L. Sarto. Mutual fund performance attribution and market timing using portfolio holdings. *International Review of Economics and Finance*, 57:353 – 370, 2018.
- C. Asness, A. Frazzini, and L. H. Pedersen. ”will my risk parity strategy outperform?”: A comment. *Financial Analysts Journal*, 69(2):12–15, 2013.
- C. S. Asness, A. Frazzini, and L. H. Pedersen. Leverage aversion and risk parity. *Financial Analysts Journal*, 68(1):47 – 59, 2012.
- T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307 – 327, 1986.
- T. Bollerslev. A conditionally heteroskedastic time series model for speculative prices and rates of return. *The Review of Economics and Statistics*, 69(3):542–547, 1987.
- M. A. Boyacioglu and D. Avci. An adaptive network-based fuzzy inference system (anfis) for the prediction of stock market return: The case of the istanbul stock exchange. *Expert Systems with Applications*, 37(12):7908 – 7912, 2010.
- L. Cao, L. K. Guan, and Z. Jingqing. Bond rating using support vector machine. *Intell. Data Anal.*, 10(3), 2006.

- S. Chen, W. K. Härdle, and K. Jeong. Forecasting volatility with support vector machine-based garch model. *Journal of Forecasting*, 29(4):406–433, 2010.
- W.-H. Chen and J.-Y. Shih. A study of taiwan’s issuer credit rating systems using support vector machines. *Expert Systems with Applications*, 30(3):427 – 435, 2006.
- K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. pages 35–46, 2000.
- T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.
- R. Donaldson and M. Kamstra. An artificial neural network-garch model for international stock return volatility. *Journal of Empirical Finance*, 4(1):17 – 46, 1997.
- R. F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.
- P. H. Franses and M. McAleer. Financial volatility: an introduction. *Journal of Applied Econometrics*, 17(5):419–424, 2002.
- A. Frazzini and L. H. Pedersen. Betting against beta. *Journal of Financial Economics*, 111:1 – 25, 2014.
- E. Gerlein, M. McGinnity, A. Belatreche, and S. Coleman. Evaluating machine learning classification for financial trading: An empirical approach. *Expert Systems with Applications*, (54):193–207, 2016.
- C. Giraud. *Introduction to High-Dimensional Statistics*. Chapman and Hall/CRC, 2014.
- L. R. Glosten, R. Jagannathan, and D. E. Runkle. On the relation between the expected value and the volatility of the nominal excess return on stocks. *The Journal of Finance*, 48(5):1779–1801, 1993.
- E. Hajizadeh, A. Seifi, M. F. Zarandi, and I. Turksen. A hybrid modeling approach for forecasting the volatility of s&p 500 index return. *Expert Systems with Applications*, 39(1):431 – 436, 2012.

- G. Hoffman. Credit suisse has its own 'fear barometer,' and it's flashing red (vix), 2017. URL <https://markets.businessinsider.com/news/stocks/credit-suisse-has-its-own-fear-barometer-and-its-flashing-red-2017-4-1001921106-1001921106>.
- S.-H. Hsu, J. P.-A. Hsieh, T.-C. Chi, and K.-C. Hsu. A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression. *Expert Systems with Applications*, (36):7947–7951, 2009.
- W. Huang, Y. Nakamori, and S.-Y. Wang. Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 32(10):2513 – 2522, 2005.
- K. jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1):307 – 319, 2003.
- A. M. J.P.Morgan, 2012. URL https://am.jpmorgan.com/blobcontent/1378404528937/83456/11_77%20Risk%20Parity.pdf.
- C. Kessler and S. Günter. Investment managment for private investors at sbc - the benchmark, 1996.
- K. Kim and H. Ahn. A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach. *Computers and Operations Research*, 39:1800–1811, 2012.
- Y. Kim and D. Enke. Using neural networks to forecast volatility for an asset allocation strategy based on the target volatility. *Procedia Computer Science*, 95:281 – 286, 2016.
- U. Kreßel. Advances in kernal methods: support vector learning. pages 255–268, 1999.
- W. Kristjanpoller and M. C. Minutolo. Forecasting volatility of oil price using an artificial neural network-garch model. *Expert Systems with Applications*, 65:233 – 241, 2016.
- W. Kristjanpoller, A. Fadic, and M. C. Minutolo. Volatility forecast using hybrid neural network models. *Expert Systems with Applications*, 41(5):2437 – 2442, 2014.

- R. Lörtscher. Investment management for private investors at sbc - basic premises and model, 1990.
- McClellan. The mclellan oscillator & summation index, 2019. URL https://www.mcoscillator.com/learning_center/kb/mcclellan_oscillator/the_mcclellan_oscillator_summation_index/.
- S. A. Monfared and D. Enke. Volatility forecasting using a hybrid gjr-garch neural network model. *Procedia Computer Science*, 36:246 – 253, 2014.
- B. B. Nair, P. S. Kumar, N. Sakthivel, and U. Vipin. Clustering stock price time series data to generate stock trading recommendations: An empirical study. *Expert Systems with Applications*, 70:20 – 36, 2017.
- A. Nazemi, B. Abbasi, and F. Omidi. Solving portfolio selection models with uncertain returns using an artificial neural network scheme. *Applied Intelligence*, 10 2014.
- L. Ortega and K. Khasahanah. A neuro-wavelet model for the short-term forecasting of high-frequency time series of stock returns. *Journal of Forecasting*, (33):134–146, 2014.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- J. Platt, N. Christianini, and J. Shawe-Taylor. Large margin das for multiclass classification. 12:547–553, 2000.
- S. Raschka. How do i select svm kernels, 2016. URL https://github.com/rasbt/python-machine-learning-book/blob/master/faq/select_svm_kernels.
- G. Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 03 1978.
- V. Sharma, D. Baruah, D. Chutia, P. Raju, and D. K. Bhattacharya. An assessment of support vector machine kernel parameters using remotely sensed satellite data. In *2016*

- IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 1567–1570, 2016.
- W. F. Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19(3):425–442, 1964.
- V. N. Vapnick. *Statistical learning theory*. Adaptive and learning systems for signal processing. New York [etc.] : John Wiley and Sons. cop. 1998., 1998.
- V. N. Vapnik. *The nature of statistical learning theory*. New York; Heidelberg and London:, 1995.
- S. Verize. Model selection and psychological theory: a discussion of the differences between the akaike information criterion (aic) and the bayesian information criterion (bic). *Psychological Methods*, 17:228–243, 2012.
- J. Weston and C. Watkins. Support vector machines formulticlass pattern recognition. pages 219–224, 1999.
- W. Wilder. *New Concepts In Technical Trading Systems*. Hunter Publishing Company, 1978.