University of Warsaw
Faculty of Economic Sciences

Working Papers
No. 8/2021 (356)

GARCHNET - VALUE-AT-RISK FORECASTING WITH NOVEL APPROACH TO GARCH MODELS BASED ON NEURAL NETWORKS

Mateusz Buczyński
Marcin Chlebus

Warsaw 2021

WORKING PAPERS 7/2021 (355)

GARCHNet - Value-at-Risk forecasting with novel approach to GARCH models based on neural networks

Mateusz Buczyński[a] *, Marcin Chlebus[b]

[a] *Interdisciplinary Doctoral School, University of Warsaw*
[b] *University of Warsaw, Faculty of Economic Sciences*
** Corresponding author: mp.buczynski2@uw.edu.pl*

Abstract: This study proposes a new GARCH specification, adapting a long short-term memory (LSTM) neural network's architecture. Classical GARCH models have been proven to give substantially good results in the case of financial modeling, where high volatility can be observed. In particular, their high value is often praised in the case of Value-at-Risk. However, the lack of nonlinear structure in most of the approaches entails that the conditional variance is not represented in the model well enough. On the contrary, recent rapid advancement of deep learning methods is said to be capable of describing any nonlinear relationships prominently. We suggest GARCHNet - a nonlinear approach to conditional variance that combines LSTM neural networks with maximum likelihood estimators of probability in GARCH. The distributions of the innovations considered in the paper are: normal, t and skewed t, however the approach does enable extensions to other distributions as well. To evaluate our model, we have executed an empirical study on the log returns of WIG 20 (Warsaw Stock Exchange Index) in four different time periods throughout 2005 and 2021 with varying levels of observed volatility. Our findings confirm the validity of the solution, however we present several directions to develop it further.

Keywords: Value-at-Risk, GARCH, neural networks, LSTM

JEL codes: G32, C52, C53, C58

1 Introduction

For decades the uncertainty at the financial markets has been the main point of research related to risk (Segal et al., 2015; Vorbrink, 2014). A market standard that was established more than 30 years ago to be used as a measure of risk is Value-at-Risk (VaR) (Duffie and Pan, 1997). It is the most straightforward way of expressing potential losses over a target horizon with a specific statistical confidence. The simplicity of VaR doesn't restrain numerous approaches from being proposed (Engle and Manganelli, 2004; Barone-Adesi et al., 2008; Wang et al., 2010). Such scientific abundance imposes that various approaches to calculation of VaR might be in use and still be considered 'good'. Whether one can address the model as of good quality is the point of the debate of many researchers in this field (Abad et al., 2014; Nozari et al., 2010; Ergün and Jun, 2010; Degiannakis et al., 2012; Escanciano and Olmo, 2010; Abad and Benito, 2013). Even though the models have been thoroughly tested and are deemed statistically correct, temporal changes in financial time series characteristics are still not given enough attention, which can lead to over- or underestimation of risk. The best example of such situation is a financial crisis from 2008 (Degiannakis et al., 2012) or more recent market crash caused by COVID-19 (Omari et al., 2020). Therefore financial industry - both regulators and financial institutions - turns towards a better, probabilistic way to estimate risk from the past events that is able to adapt to recent shocks rapidly (So and Philip, 2006). As of writing, the official estimate of market risk is Expected Shortfall (ES) proposed by Basel Committee. It estimates the expected value of potential loss if such loss on the considered asset will be lower than VaR.

One of the most influential factors that drives the risk is the variance, in particular its changing temporal structure or tendency to cluster (Cont, 2002). There exists a broad family of models that is aimed at capturing such effect, while the most common one is Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model, proposed by Bollerslev (1986). Due to the fact that financial markets usually exhibit the known stylized facts, a more fitting approach is to use fat-tailed distribution (Aloui and Mabrouk, 2010). The introduction of distributions such as t or GED distributions that allow to model skewness and heavy tails has dispelled any doubts regarding the validity of GARCH models (BenSaïda, 2015; Bonato, 2012). Another extension of these models, proposed by Francq and Zakoïan (2004), assumes that not only the variance exhibits temporal

changes, but also the mean, however in terms of financial returns the mean is usually irrelevant in the long term (Fama, 1998).

On the other hand, financial researchers are much keen on implementing machine learning methods (Sezer et al., 2020). The deep neural networks (deep NNs) are considered to be a good substitute to the conventional statistical methods, not only in financial markets field, but also in other areas of science (Mnih et al., 2013; Devlin et al., 2018; Cho et al., 2014). However in the case of time series data, recurrent approaches, such as long short term memory (LSTM) NNs are better (Goodfellow et al., 2016). In addition to that, NNs offer a nonlinear estimator of the probability function (Chen and Billings, 1992). In case of GARCH models a conditional variance function usually assumes a linear or a very simple nonlinear relationship between the probability function (as well as distribution's moments) and the observables (Glosten et al., 1993a; Nelson and Cao, 1992).

According to Lim et al. (2019), the best approach to exploit machine learning in the time series domain is not to fully replace statistical and econometric approaches. They rather propose to combine the best of both worlds, hence the idea of this paper is to model the conditional variance with NN. There have been a few studies in the intersection of GARCH models and NNs already. For example, Arnerić et al. (2014) have proposed to model time series with GARCH model, but with an extension to RNN by the name of Jordan NNs. A similar research by Kristjanpoller and Minutolo (2015, 2016) proposes a ANN-GARCH model and their results indicate a 25% reduction in mean absolute percentage error (MAPE). A research by Kim and Won (2018) takes a step further by incorporating a LSTM layer into the neural network, reporting a decrease in mean absolute error (MAE) of 37.2%. Yet another approach, proposed by Jeong and Lee (2019) considers a RNN model to specify the autoregressive moving average (ARMA) process that drives not the conditional variance, but the conditional mean. Their results reveal that such an approach leads to a decrease in MAPE by about 10%.

However, the aforementioned research doesn't focus particularly on the implementation of the NNs into conditional variance itself. For example the research by Kristjanpoller and Minutolo (2015, 2016) utilizes GARCH estimates of the volatility as an input to the NN model, whereas Kim and Won (2018) build a NN with covariates being the parameters of artificially generated GARCH models. Our approach leans toward the estimation of conditional moments of the assumed

distribution with the usage of NNs, as e.g. in Rothfuss et al. (2019). The first to propose such approach were Nikolaev et al. (2011), who have studied an approach with recurrent NNs (RNNs) to represent the conditional variance and have found out that the inclusion of nonlinear methods (RNN-GARCH) decreases the model uncertainty. Further on, Liu and So (2020) consider usage of LSTM NN to model the conditional variance directly by the maximum likelihood approach of the assumed distribution's density function. They have shown that the method can successfully specify both standard deviation and variance of the financial returns. Another advantage of their approach is the option to use explainable artificial intelligence (XAI) methods. However, instead of using an estimate, they have assumed the values of additional (besides first and second moments) parameters of the distribution. Another research by Nguyen et al. (2019) proposes a fairly similar approach, but to a stochastic volatility (SV) model, which is related to GARCH. In their research they propose a SV-LSTM model, which instead of applying AR(1) process to model volatility, uses a LSTM NN. Their results indicate that the proposed approach can give better out of sample estimates than standard SV models.

In this paper we propose **GARCHNet** - conditional specification for GARCH models based on the NNs with the extensive use of a LSTM layer. Our incentives are based on the previously raised GARCH drawbacks and that the LSTM NN is able to adequately represent any nonlinear relationships that are present in the financial time series data. We also extend previous research in this field by proposing another distributions - we propose GARCHNet with normal, t and skewed t distributions and we provide necessary negative log likelihood functions for all of them to be used as cost functions in the backpropagation algorithms of NN optimization.

We also propose an empirical experiment to verify the usefulness of the GARCHNet model. The experiment is to estimate one day ahead forecasts of Value-at-Risk over a horizon of 250 testing instances (roughly one trading year) and compare them with equivalent GARCH models. The experiment was conducted on the log returns from the index WIG20 (Warsaw Stock Exchange Index; Poland) over four different time frames (both training and testing samples) throughout 2005 - 2021. The experiment was written and conducted in Python and pytorch (Paszke et al., 2019).

The paper is organized as follows. In section 2 we introduce the theoretical foundations for GARCHNet and the necessary background for VAR backtesting. In section 3 we describe the em-

pirical experiment with data and model's descriptions. In section 4 we report the results of the experiment and in section 5 we have entered our concluding remarks and paths to extend our framework in the future research.

2 Methodology

2.1 GARCH models

GARCH model with no mean (pure GARCH process) can be specified as:

$$\begin{aligned} r_t &= \mu_t + \epsilon_t, \\ \epsilon_t &= \sigma_t z_t, \end{aligned} \tag{1}$$

where r_t is observed time series, μ_t is conditional mean of the process and σ_t is the conditional standard deviation of the observed time series process. z_t is an innovation process and is considered to be i.i.d with unit variance, in the most straightforward approach the assumed distribution is normal: $z_t \sim \mathcal{N}(0, 1)$.

In VaR area, many conditional variance definitions have been proposed already: standard GARCH (Bollerslev, 1986), EGARCH (Nelson, 1991), IGARCH (Engle and Bollerslev, 1986) or GJR-GARCH (Glosten et al., 1993b). However in this paper we only utilize standard GARCH(p, q) process, which defines conditional volatility as:

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \beta_i z_{t-i}^2 + \sum_{i=1}^{p} \gamma_i \sigma_{t-i}^2, \tag{2}$$

where p and q are numbers of lags of conditional variance and innovation respectively, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are parameter vectors to be estimated. The stationarity assumption of GARCH process is met with $\sum_{i=1}^{q} \beta_i + \sum_{i=1}^{p} \gamma_i < 1$.

When it comes to the optimization of this process, one of the possible procedures is to use quasi maximum likelihood (QML). Given that the innovations are assumed to be independent, the conditional log likelihood of the vector of demeaned observed time series $\boldsymbol{\epsilon}$ of length T can be denoted as a sum of all log conditional densities of particular innovations ϵ_t (see Francq and Zakoïan,

2004):

$$\ell(\boldsymbol{\theta};\boldsymbol{\epsilon}) = \sum_{t=1}^{T} \ell_t(\boldsymbol{\theta};\epsilon_t) = \sum_{t=1}^{T} log f(\epsilon_t|\epsilon_{t-1},...,\epsilon_1;\boldsymbol{\theta}) = \sum_{t=1}^{T} log f(\epsilon_t;\boldsymbol{\theta}), \tag{3}$$

where $\boldsymbol{\theta} = (\omega, \beta_1, ..., \beta_q, \gamma_1, ..., \gamma_p)$ is a vector of parameters, $f(\epsilon_t|\epsilon_{t-1},...,\epsilon_1;\boldsymbol{\theta})$ is conditional density function of innovation ϵ_t, however, given that the innovations are independent it will reduce to $f(\epsilon_t;\boldsymbol{\theta})$.

Quasi maximum likelihood estimation of the parameters vector $\boldsymbol{\theta}$ is a solution $\hat{\boldsymbol{\theta}}$ of:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} \ell(\boldsymbol{\theta},\boldsymbol{\epsilon}) \tag{4}$$

$$\ell(\boldsymbol{\theta},\boldsymbol{\epsilon}) = \frac{1}{T}\sum_{t=1}^{T} \ell_t(\boldsymbol{\theta},\epsilon_t) \tag{5}$$

In the case z_t is normally distributed, conditional log likelihood function for one observation is equal to:

$$\ell_t(\boldsymbol{\theta},\epsilon_t) = -\frac{1}{2} log\sigma_t^2 - \frac{1}{2}\frac{\epsilon_t^2}{\sigma_t^2}, \tag{6}$$

which comes down to a logarithm of a normal density function.

In the case z_t is t distributed, an additional parameter is necessary to be estimated - η - number of degrees of freedom of this distribution, with an assumption of $\eta > 2$. Therefore the parameter vector is $\boldsymbol{\theta} = (\omega, \beta_1, ..., \beta_q, \gamma_1, ..., \gamma_p, \eta)$, and conditional log likelihood for one observation is:

$$\ell_t(\boldsymbol{\theta},\epsilon_t) = \log\Gamma\left(\frac{\eta+1}{2}\right) - \log\Gamma\left(\frac{\eta}{2}\right) - \frac{1}{2}\log(\pi\,(\eta-2)\,\sigma_t^2) - \frac{\eta+1}{2}\log\left(1+\frac{\epsilon_t^2}{\sigma_t^2(\eta-2)}\right), \tag{7}$$

where $\Gamma(\cdot)$ is a gamma function and the log likelihood is a logarithm of density of t distribution.

In the last case we consider z_t as skewed t distributed. Yet another parameter is introduced - λ, responsible for the skewness of the distribution. The particular analytic implementation of skewed t distribution was proposed by Hansen (1994). In this case an additional assumption is that $-1 < \lambda < 1$. Parameter vector is once again extended to $\boldsymbol{\theta} = (\omega, \beta_1, ..., \beta_q, \gamma_1, ..., \gamma_p, \eta, \lambda)$.

$$\ell_t = \ln\left[\frac{bc}{\sigma}\left(1+\frac{1}{\eta-2}\left(\frac{a+bx/\sigma}{1+sgn(x/\sigma+a/b)\lambda}\right)^2\right)^{-(\eta+1)/2}\right], \tag{8}$$

where

$$a = 4\lambda c\frac{\eta-2}{\eta-1}, \quad b^2 = 1+3\lambda^2-a^2, \quad c = \frac{\Gamma\left(\frac{\eta+1}{2}\right)}{\sqrt{\pi\,(\eta-2)}\Gamma\left(\frac{\eta}{2}\right)}, \tag{9}$$

All of the log likelihood functions are obtainable numerically. In addition to that, specific form of the conditional variance does not influence the QML in the above form. The assumed distribution has much more impact on it. This opens up a possibility to use much more complicated, nonlinear forms, such as NNs (Goodfellow et al., 2016).

2.2 LSTM neural networks

Long Short Term Memory (LSTM) neural networks are an extension of recurrent neural networks (RNNs), proposed by Rumelhart et al. (1986). RNNs are a special kind of neural networks that introduce recurrency by enabling the use of sequential, autocorrelated data. The sequence (or observed time series) is accompanied by a hidden input, some sort of a memory state that stores the information provided with previous timesteps. The next input in the sequence is predicted with that recurrent hidden state:

$$h_t = g(W_x x_t + W_h h_{t-1} + b_h) \tag{10}$$

where $g(\cdot)$ is an activation function (e.g., logistic sigmoid, hyperbolic tangent or Rectified Linear Unit (ReLU)), $\boldsymbol{x} = (x_1, x_2, ..., x_T)$ is the sequence of observed time series of length T, while $\boldsymbol{h} = (h_1, h_2, ..., h_T)$ represents a random vector - hidden state of the same length T. W_x and W_h are weight matrices (parameters) of the neural network, corresponding to $\boldsymbol{x}$ and $\boldsymbol{h}$ respectively and b_h is a bias vector. Such equation assumes that the sequence can be of infinite length, or at least of arbitrarily large number T, however due to computational obstacles (such as the problem of vanishing or exploding gradients (Pascanu et al., 2012)) the sequence length T is practically limited to only a few timesteps.

The problem mentioned above is practically solved by introduction of LSTM (Hochreiter and Schmidhuber, 1997). LSTMs expand the idea of hidden states by introducing gating mechanisms, which tell whether to preserve or ignore the input from the hidden state. Given that, LSTMs can ”remember” or ”forget” particular timesteps if necessary, building the long-term dependency parameter matrix. In detail, there are three gates: forget, input and output.

The following equations calculated iteratively build up LSTM network:

$$i_t = g(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i) \tag{11}$$

$$f_t = g(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f) \tag{12}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \tag{13}$$

$$o_t = g(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_t + b_o) \tag{14}$$

$$h_t = o_t \odot h(c_t) \tag{15}$$

$$y_t = W_{yh}h_t + b_y \tag{16}$$

where W terms denote weight matrices (e.g.: W_{ix} is a matrix of weights from the input gate to the input x), the b terms denote bias vectors (e.g. b_i is the input gate bias vector), $g(\cdot)$ and $h(\cdot)$ denote sigmoid and hyperbolic tangent activation functions respectively here, i, f and o denote input, forget and output gates respectively, c_t is another hidden state vector, specifically named cell activation vector (responsible for activating specific gates). The output of the neural network can be any distribution $p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta})$, however most often some particular moment of this distribution is estimated directly - in our case we would like it to be conditional variance.

2.3 GARCHNet

Our idea for the GARCH process specification is to use neural network as an approximation of the true conditional variance specification. In order to optimize the NN, likelihood functions described in section 2.1 come in handy. They are used as cost functions - in a negative form (negative log likelihoods). In GARCHNet, the GARCH specification is as follows:

$$l_n = W_{l_n l_{n-1}} l_{n-1} + b_{l_n} \tag{17}$$

and

$$_1 = W_{ly}y_t + b_{l_1}, \tag{18}$$

where y_t is calculated as in equation 16 and n determines the number of following fully connected layers. Given that, conditional variance is the function of the last n fully connected layer:

$$\sigma_t^2 = g(W_{Vl} l_n + b_V), \tag{19}$$

where $g(\cdot)$ is a function with non-negative output (e.g. softplus), while W_{Vl} is a matrix of weights from the last hidden layer to the output layer and b_V denotes its bias. The input to such LSTM neural network is p last observed time series realizations (chosen beforehand). Its output will be an estimate of conditional variance. Due to the specific mechanism that drives the LSTM layers' forgetting mechanism, we don't have to worry that the sequence, which is fed to the model might be too long. NNs are usually optimized using the backpropagation algorithm (Goodfellow et al., 2016), that considers calculation of gradients for each neuron in a layer and then applying changes to weights iteratively based on the value of the cost function.

However, in the density functions of t and skewed t distributions, there are two additional parameters that are necessary to be estimated or assumed. In our scenario these parameters are estimated with the same NN as a function of time. Therefore degrees of freedom η and skewness λ are estimated as:

$$\eta = g(W_{El} l_n + b_E) + 2, \tag{20}$$

$$\lambda = h(W_{Sl} l_n + b_S), \tag{21}$$

where $g(\cdot)$ is a function with non-negative output (e.g. softplus), $h(\cdot)$ is a function with output in the range $(-1, 1)$, while W are matrices of weights from the last hidden layer to the particular output layer (degrees of freedom η and skewness λ respectively) and b vectors denote their biases. Please note that we are adding two units to the output of degrees of freedom η to meet the assumption that $\eta > 2$. A complementary approach would imply changes in the log likelihood function.

This means that in the most advanced scenario, for skewed t distribution, there are three last hidden layers (one for conditional variance σ_t^2, one for degrees of freedom η and one for skewness λ), each resulting in one different output neuron.

Originally, conditional variance's parameters (ω, β and γ) should be non-negative (Bollerslev, 1986), which together with non-negativity of random variables (σ_t^2 and z_t^2) suffices for the condi-

tional variance to be non-negative as well. In the case of neural network, such assumption could lead to the worsening of the accuracy of estimated solution (Chorowski and Zurada, 2014). Instead of using such limitation, we have proposed to use softplus function (or any other that outputs non-negative values and is easily differentiable). Softplus function is defined as:

$$\text{Softplus}(x) = log(1 + exp(x)). \tag{22}$$

In the case of skewness we have proposed to use hyperbolic tangent function so that the output meets the assumption that $-1 < \lambda < 1$. Hyperbolic tangent function is defined as:

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}. \tag{23}$$

2.4 Value-at-Risk

Value-at-Risk (VaR) defines the worst possible loss with a given probability α, assuming normal market conditions for a specific time period t (Philippe, 2006). In other words, VaR is a quantile of the distribution of the observed financial time series. In our case, these are log returns of the price quotations of the respective stock index.

$$P(r_t < VaR_\alpha(t)|\Omega_{t-1}) = \alpha, \tag{24}$$

where r_t is the realization of the observed financial time series and Ω_{t-1} is an information set given at the time $t-1$.

When GARCH models are employed, VaR is calculated as an α quantile of the assumed innovation distribution weighted by the estimate of the conditional standard deviation σ_t, plus an estimate of conditional mean μ_t (Angelidis et al., 2004):

$$VaR_\alpha = \mu_t + \sigma_t F^{-1}(\alpha). \tag{25}$$

2.4.1 Quality of VaR forecasts

The basic tool that allows to assess the quality of the VaR forecast is the count of the instances, in which VaR forecast was lower (in absolute terms) than the realization of the observed time series - excess count or proportion of failures (Chlebus, 2017):

$$\hat{\alpha} = \frac{1}{N} \sum_{t=1}^{N} I_{VaR_\alpha > r_t}, \tag{26}$$

where N is the number of testing instances and $\sum_{t=1}^{N} I_{VaR_\alpha > r_t}$ is the number of exceedances $= n$.

Statistically, this number comes from binomial distribution (assuming that the exceptions are IID). Basel Committee strictly regulates what values constitute to the 'safe zone' or require the model to be looked upon. Specifically the name of such test is Traffic Light Test (Costanzino and Curran, 2018). In the case of VaR at 2.5% significance level and 250 testing instances the 'safe' (green) zone ends with 10 exceptions (95% cumulative probability) and yellow (warning zone) ends with 16 exceptions (99.99% cumulative probability).

The unconditional coverage (UC) test by Kupiec (1995) builds up on the idea that the overall number of exceptions should follow the binomial distribution. To test that a likelihood ratio test is proposed:

$$LR_{UC} = -2ln\left(\frac{(1-\alpha)^{N-n}\alpha^n}{(1-\hat{\alpha})^{N-n}\hat{\alpha}^n}\right). \tag{27}$$

There is also a conditional coverage (CC) test by Christoffersen (1998). In addition to the unconditional coverage, Christoffersen test measures the likelihood of unusually frequent VaR exceptions - an effect of exceptions clustering.

Let us assume that the exceptions follow a first-order Markov chain with transition probability matrix:

$$\Pi_1 = \begin{bmatrix} 1-\pi_{01} & \pi_{01} \\ 1-\pi_{11} & \pi_{11} \end{bmatrix}, \tag{28}$$

where $\pi_{ij} = P(I_t = j | I_{t-1} = i)$ and I_t is an indicator variable that takes 1 if there was an exceptions

and 0 otherwise. The likelihood of such a process is:

$$L(\Pi_1; I_1, ..., I_N) = (1 - \pi_{01})^{n_{00}} \pi_{01}^{n_{01}} (1 - \pi_{11})^{n_{01}} \pi_{11}^{n_{11}}. \tag{29}$$

The maximum likelihood estimate of parameters π_{ij} is:

$$\Pi_1 = \begin{bmatrix} \frac{n_{00}}{n_{00}+n_{01}} & \frac{n_{01}}{n_{00}+n_{01}} \\ \frac{n_{10}}{n_{10}+n_{11}} & \frac{n_{11}}{n_{10}+n_{11}} \end{bmatrix}. \tag{30}$$

Now, let us assume that the exceptions are independent. In terms of first-order Markov chain model the transition matrix would be:

$$\Pi_2 = \begin{bmatrix} 1 - \pi_2 & \pi_2 \\ 1 - \pi_2 & \pi_2 \end{bmatrix}. \tag{31}$$

In this scenario likelihood function becomes:

$$L(\Pi_2; I_1, ..., I_N) = (1 - \pi_2)^{(n_{00}+n_{10})} \pi_2^{(n_{01}+n_{11})}. \tag{32}$$

Finally, using likelihood ratio test the CC's independence test can be concluded:

$$LR_{ind} = -2ln\left(\frac{L(\Pi_2; I_1, ..., I_N)}{L(\Pi_1; I_1, ..., I_N)}\right). \tag{33}$$

The conditional coverage test consists of both unconditional coverage and independence tests: $LR_{cc} = LR_{uc} + LR_{ind}$.

Even more restrictive is dynamic quantile (DQ) test by Engle and Manganelli (2004). They define another random variable $Hit_t = I_t - \alpha$. The null hypothesis of this test is that the expected value of the Hit_t explained with the information available at $t - 1$ is zero. To test that, they implement a linear regression model:

$$Hit_t = \delta + \sum_{k=1}^{K} \beta_k X_{t-k} + \epsilon_t, \tag{34}$$

where matrix X might include both lags of Hit, r or VaR. DQ test statistic is then:

$$DQ = \frac{Hit'X(X'X)^{-1}X'Hit}{\alpha(1-\alpha)} \tag{35}$$

Another interesting dimension to compare models are loss functions (LFs). Their value determines the loss in case of the occurrence of the model's failure. There are two sides that are usually interested in these values - regulator and the companies themselves. Both of them weight some business aspects differently. From the regulatory point of view the most important aspect is the value lost on the occurrence of VaR exception, whereas from the point of view of a company - the alternative cost of holding excessive reserves.

To compare the models we have chosen following loss functions, from the proposed by Abad et al. (2015):

- Lopez quadratic LF (LLF):

$$LLF_t = \begin{cases} 1 + (VaR_t - r_t)^2 & \text{if } r_t < VaR_t, \\ 0 & \text{otherwise;} \end{cases} \tag{36}$$

- Caporin regulator's LF (CRLF):

$$CLF_t = \begin{cases} |1 - |r_t/VaR_t|| & \text{if } r_t < VaR_t, \\ 0 & \text{otherwise;} \end{cases} \tag{37}$$

- Caporin firm's LF (CFLF):

$$CFLF_t = |1 - |r_t/VaR_t|| \text{ for all } r_t; \tag{38}$$

- Abad, Benito, Lopez's LF (ABLLF):

$$ABLLF_t = \begin{cases} (VaR_t - r_t)^2 & \text{if } r_t < VaR_t, \\ \beta(r_t - VaR_t) & \text{otherwise,} \end{cases} \tag{39}$$

where β is a parameter that represents a cost of capital, originally an interest rate.

3 Data and model specifications

3.1 Data

GARCHNet's performance was measured empirically by backtesting on the log returns of price quotations of WIG20 (Warsaw Stock Exchange; Poland). Therefore our observed time series is:

$$r_t = logp_t - logp_{t-1}. \tag{40}$$

Such data is openly available, e.g.: from Stooq (2021). As a reference we have also estimated corresponding GARCH models on the same data samples.

We have subjectively chosen four different time periods consisting of 1250 observations each (1000 for the training sample and 250 for the testing sample). The start dates of specific periods are as follows: (i) 2005-01-01 (testing on year 2009), (ii) 2007-01-01 (testing on year 2011), (iii) 2013-01-01 (testing on year 2017), (iv) 2016-01-01 (testing on year 2020). In our opinion these periods provide a full view of possible volatility levels in training and testing samples - training and testing samples both show low volatility (sample starting in 2013 respectively) or the volatility is different for training and testing samples (low volatility training samples starting in 2005 and 2016; and high volatility training sample starting in 2007). Such a spectrum allows us to test the model in varying market conditions.

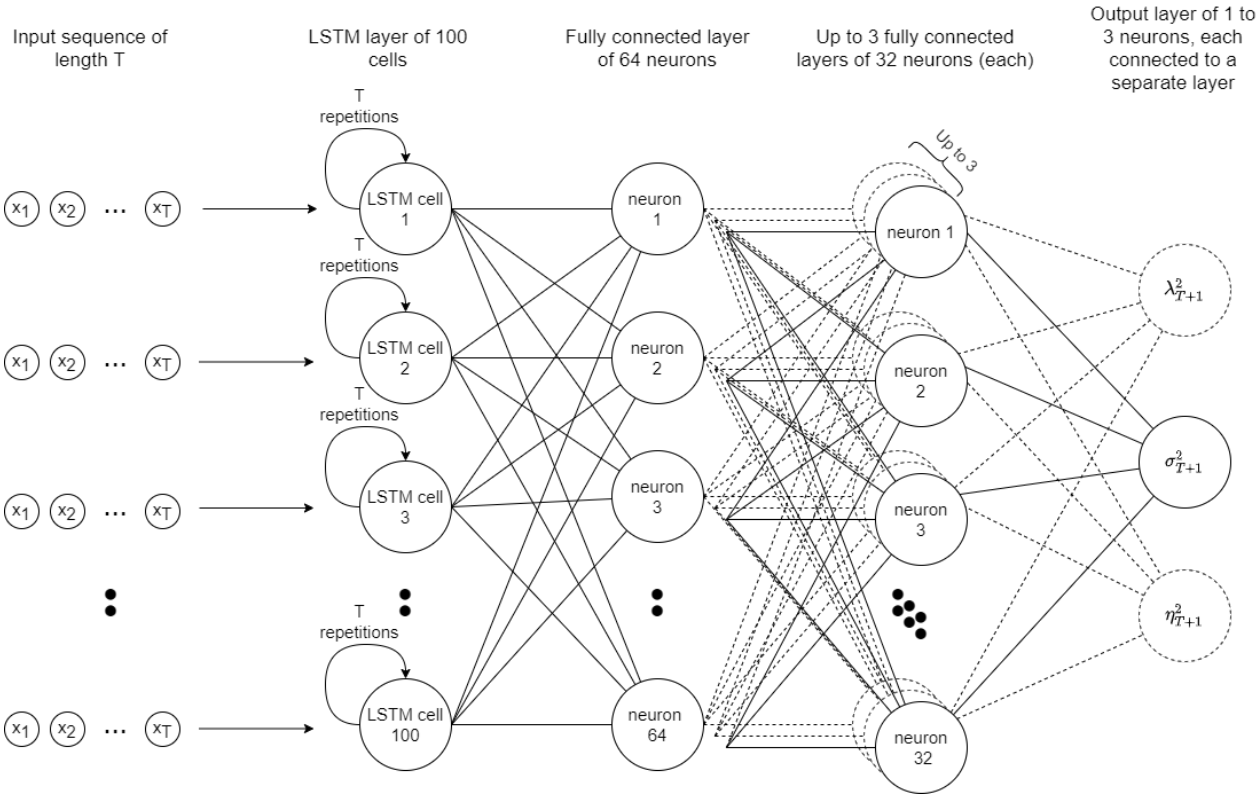
3.2 Models

We have compared proposed GARCHNet specifications with corresponding standard GARCH models. In order to do that we have also had to choose the number of observations p that constitutes a sequence for LSTM model. Due to resemblance to the original meaning of p in GARCH model (number of lagged conditional variances in its model) we have controlled both these parameters with p. The test includes $p \in \{5, 10, 20, 100\}$.

We have used a rolling-window estimation approach (Zanin and Marra, 2012). For each timestep in the testing sample, we have prepared a new model with new randomly initialized weights and trained it using last 1000 observations.

For the neural network that specifies the conditional variance's specification we have used a rather small architecture: one layer LSTM with 100 neurons (fed with sequence of length p), followed by three ($n = 3$) fully-connected layers with 64, 32 and 1 neuron(s) respectively. In the case of t and skewed t distributions there were two (and three respectively) output layers to correspond with the number of parameters to be optimized. The parameters were optimized using Adam optimizer with learning rate equal to 3e-4 and batch size of 512. Due to rolling-window approach it was hard to choose an automatic threshold for the number of epochs to avoid overfitting, so we have arbitrarily trained each model for 300 epochs.

Figure 1: The diagram with the architecture of GARCHNet model



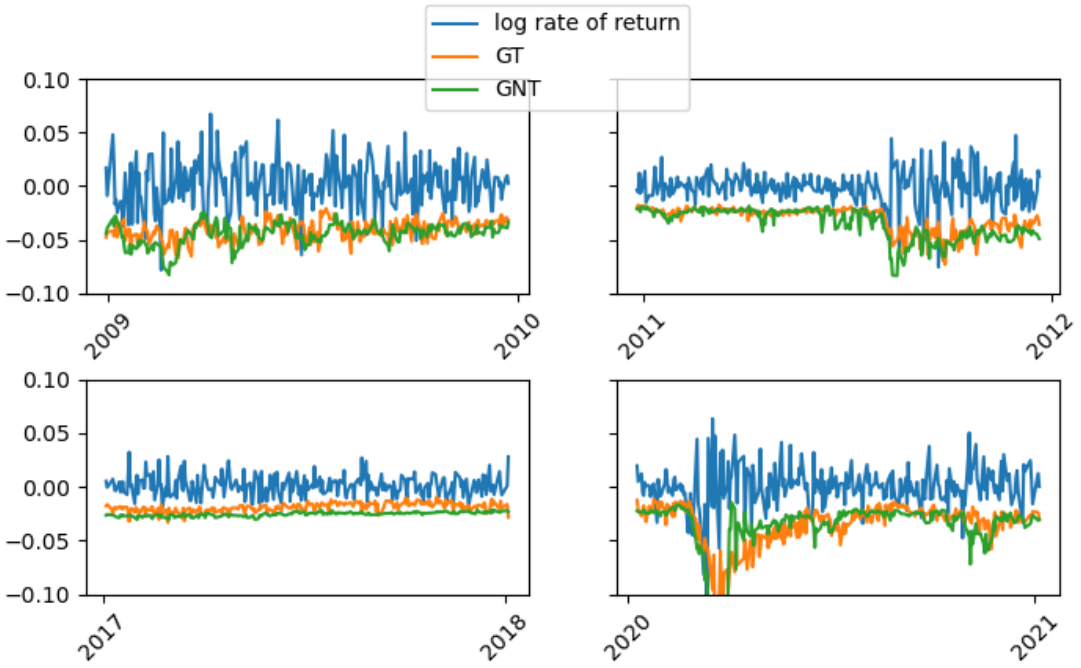
4 Results

The results of the experiment are satisfactory. In the figure 2 you can observe the relation between forecasts by GARCH and GARCHnet with t distributed innovations. The presented relationship is one of the best. One can notice that the GARCHNet's predictions do not depart from the rate of return, even more - for some of the timesteps GARCHNet is much quicker to acknowledge

the presence of volatility shocks. There is no GARCH model that would be better than its GARCH-Net equivalent in all tested periods and for all tested sequence lengths p in terms of number of exceptions. However GARCHNet with t distribution seems to have the highest overhead. Only in case of two periods it had the number of exceptions greater than GARCH with t distribution. In the case of the remaining models the overhead is much smaller and sometimes negative. However, we believe that the predictive power of such model can be improved with the better neural architecture. GARCHNet with skewed t distribution is worse by a small margin, which is not in line with its GARCH equivalent - GARCH with skewed t distribution is the best model compared to the rest of its family. This might indicate that the approach to the estimation of the distribution's parameters is inefficient with Adam optimizer or the approach that we have taken should be reconsidered. For example the parameter estimate should be changed for estimate over whole training sample, not based on the rather small sample (of length p) time series in prediction stage.

In the table 1 we have presented the results of statistical tests, as well as in the figure 3 you can observe the numbers of exceptions. Let's focus on the two first periods: the ones starting

Figure 2: Exemplary comparison for the GARCH and GARCHNet with t distribution for $p = 20$

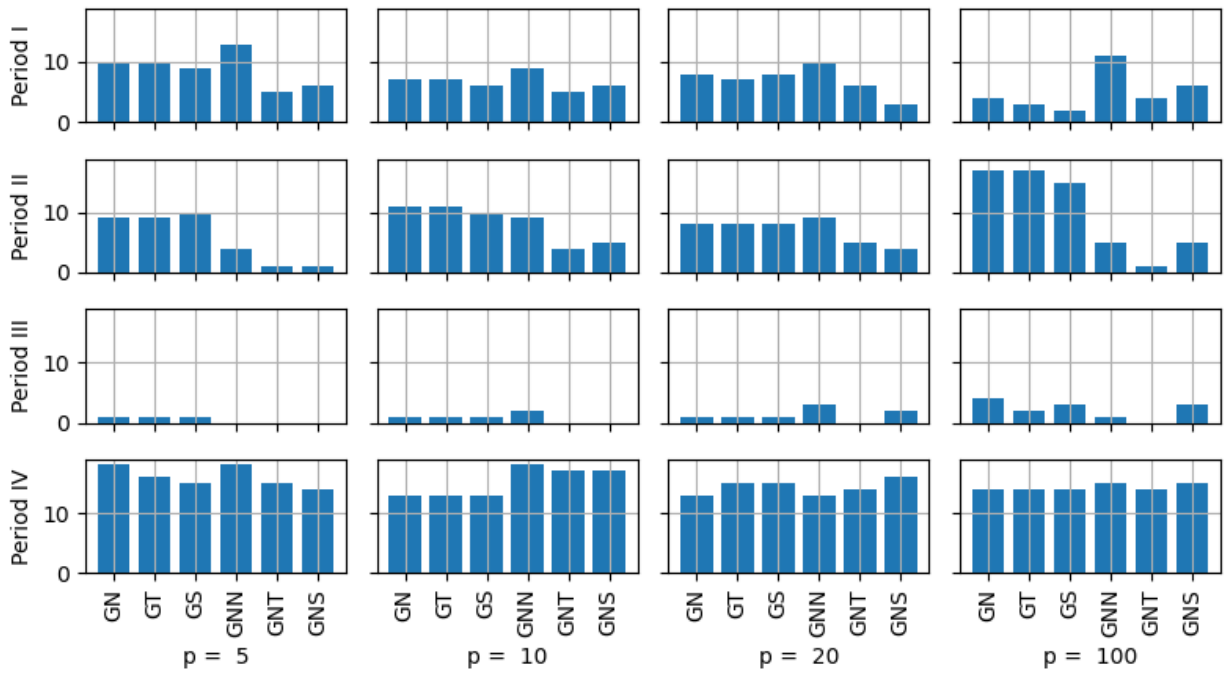


Note: GT - GARCH with t distributed innovations; GNT - GARCHNet with t distributed innovations.

in 2005 and 2007. Both these periods show a large number of failures to reject null hypotheses of considered tests. GARCHNet's numbers of exceptions do not show any outstanding features, however we notice that the DQ test was failed to be rejected much more often than in case of the standard GARCH approach. The nonlinear structure of the proposed conditional variance might not be fully explained by the linear structure of the DQ test. In terms of statistical tests GARCHNet approaches seem to provide models with greater quality on overall.

Now let's move to the samples starting in years 2013 and 2016. In these two cases we see a notably larger number of null hypothesis rejections - both due to risk under- and overestimation. However, in these two periods results of GARCHNet models are in line with the results of standard GARCH. On average GARCHNet models have the same number of exceptions. Both families of models were unable to correctly respond to the COVID-19 financial market crashes, hence large numbers of exceptions in the last analyzed period. It needs to be noted that COVID 19 period should be treated as a stress-test to these models and given GARCHNet's very similar results we would

Figure 3: Number of exceptions in regard to the studied period and number of lags p for each of the models



Note: GN - GARCH with normally distributed innovations (d.i.); GT - GARCH with t d.i.; GST - GARCH with skewed t d.i.; GNN - GARCHNet with normally d.i.; GNT - GARCHNet with t d.i.; GNS - GARCHNet with skewed t d.i.

Table 1: Results of GARCH and GARCHNet models regarding p-values of considered statistical tests

p	Model	Period I (2009)			Period II (2011)			Period III (2017)			Period IV (2020)		
		UC	CC	DQ	UC	CC	DQ	UC	CC	DQ	UC	CC	DQ
5	GN	**0.16**	**0.10**	0.00	**0.30**	**0.20**	0.00	0.01	0.00	**0.62**	0.00	0.00	0.00
	GT	**0.16**	0.02	0.00	**0.30**	**0.20**	0.00	0.01	0.00	**0.62**	0.00	0.00	0.00
	GS	**0.30**	**0.20**	0.00	**0.16**	0.02	0.00	0.01	0.00	**0.62**	0.00	0.00	0.00
	GNN	0.02	0.00	0.00	**0.33**	**0.05**	**0.09**	0.00	-	-	0.00	0.00	0.00
	GNT	**0.60**	**0.16**	**0.26**	0.01	0.00	**0.62**	0.00	-	-	0.00	0.00	0.00
	GNS	**0.92**	**0.86**	**0.99**	0.01	0.00	**0.62**	0.00	-	-	0.01	0.00	0.00
10	GN	**0.77**	**0.74**	0.00	**0.08**	0.01	0.00	0.01	0.00	**0.96**	0.02	0.00	0.00
	GT	**0.77**	**0.74**	0.00	**0.08**	0.01	0.00	0.01	0.00	**0.96**	0.02	0.00	0.00
	GS	**0.92**	**0.86**	**0.35**	**0.16**	0.02	0.00	0.01	0.00	**0.96**	0.02	0.00	0.00
	GNN	**0.30**	**0.20**	**0.50**	**0.30**	**0.20**	0.02	0.05	0.02	**0.99**	0.00	0.00	0.00
	GNT	**0.60**	**0.16**	**0.72**	**0.33**	**0.37**	**0.53**	0.00	-	-	0.00	0.00	0.00
	GNS	**0.92**	**0.86**	**0.85**	**0.60**	**0.16**	**0.10**	0.00	-	-	0.00	0.00	0.00
20	GN	**0.50**	**0.31**	**0.05**	**0.50**	**0.31**	0.00	0.01	0.00	**1.00**	0.02	0.00	0.00
	GT	**0.77**	**0.36**	**0.34**	**0.50**	**0.31**	0.00	0.01	0.00	**1.00**	0.00	0.00	0.00
	GS	**0.50**	**0.31**	0.00	**0.50**	**0.31**	0.00	0.01	0.00	**1.00**	0.00	0.00	0.00
	GNN	**0.16**	0.02	**0.06**	**0.30**	**0.20**	0.00	**0.14**	**0.12**	**0.76**	0.02	0.00	0.00
	GNT	**0.92**	**0.30**	**0.81**	**0.60**	**0.69**	0.04	0.00	-	-	0.01	0.00	0.00
	GNS	**0.14**	**0.12**	**1.00**	**0.33**	**0.05**	**0.13**	0.05	0.02	**1.00**	0.00	0.00	0.00
100	GN	**0.33**	**0.37**	**1.00**	0.00	0.00	0.00	**0.33**	**0.37**	-	0.01	0.00	0.00
	GT	**0.14**	**0.12**	**1.00**	0.00	0.00	0.00	0.05	0.02	-	0.01	0.00	0.00
	GS	0.05	0.02	**1.00**	0.00	0.00	0.00	**0.14**	**0.12**	-	0.01	0.00	0.00
	GNN	**0.08**	0.04	**0.05**	**0.60**	**0.16**	-	0.01	0.00	-	0.00	0.00	0.04
	GNT	**0.33**	**0.05**	**1.00**	0.01	0.00	-	0.00	-	-	0.01	0.00	0.01
	GNS	**0.92**	**0.86**	**1.00**	**0.60**	**0.69**	-	**0.14**	**0.12**	-	0.00	0.00	**0.18**

Note: GN - GARCH with normally distributed innovations (d.i.); GT - GARCH with t d.i.; GST - GARCH with skewed t d.i.; GNN - GARCHNet with normally d.i.; GNT - GARCHNet with t d.i.; GNS - GARCHNet with skewed t d.i.; UC - unconditional coverage test; CC - conditional coverage test; DQ - dynamic quantile test.

All statistical tests that were failed to be rejected at the 5% significance level are in bold.

Table 2: Results of GARCH and GARCHNet models regarding the values of cost functions

p	Model	Period I (2009)				Period II (2011)				Period III (2017)				Period IV (2020)			
		Caporin R Cost	Lopez Cost	Caporin Cost	ABL Cost	Caporin R Cost	Lopez Cost	Caporin Cost	ABL Cost	Caporin R Cost	Lopez Cost	Caporin Cost	ABL Cost	Caporin R Cost	Lopez Cost	Caporin Cost	ABL Cost
5	GN	2.60	10.00	**2691.55**	3.05	1.46	9.00	**3564.53**	**2.32**	0.07	1.00	**2207.26**	**1.55**	6.19	18.01	1587.55	**2.13**
	GT	2.60	10.00	2791.28	**3.03**	1.45	9.00	3575.75	2.33	0.10	1.00	2284.92	1.58	4.68	16.01	1716.42	2.30
	GS	2.11	9.00	3145.73	3.28	1.54	10.00	3588.61	**2.32**	0.09	1.00	2294.27	1.59	4.38	15.01	1652.06	2.24
	GNN	2.07	13.00	2700.18	2.93	1.32	**4.00**	3917.20	2.37	**0.00**	**0.00**	2537.18	1.61	5.60	18.01	**1585.71**	2.17
	GNT	0.95	**5.00**	3423.03	3.65	**0.23**	1.00	5182.33	3.41	**0.00**	**0.00**	3372.26	2.07	4.29	15.01	1966.34	2.53
	GNS	**0.77**	6.00	3260.19	3.43	0.47	1.00	4758.94	3.05	**0.00**	**0.00**	2945.67	1.88	**4.24**	**14.01**	1872.40	2.38
10	GN	1.79	7.00	2823.16	**3.08**	1.36	11.00	3613.36	2.27	0.06	1.00	**2145.70**	**1.54**	5.71	13.02	1878.46	2.56
	GT	1.81	7.00	2966.01	3.15	1.39	11.00	3634.79	2.28	0.09	1.00	2168.49	1.56	5.06	**13.01**	1927.94	2.60
	GS	1.49	6.00	3247.63	3.45	1.56	10.00	3672.03	2.28	0.08	1.00	2216.31	1.59	5.19	**13.01**	1880.72	2.56
	GNN	1.39	9.00	**2780.38**	3.13	2.34	9.00	**3563.36**	**2.23**	0.15	2.00	2490.13	1.58	5.50	18.01	**1489.63**	**2.26**
	GNT	0.70	**5.00**	3051.24	3.51	0.60	**4.00**	4210.55	2.67	**0.00**	**0.00**	3331.98	2.04	**4.07**	17.01	1938.49	2.55
	GNS	**0.54**	6.00	2901.20	3.55	**0.55**	5.00	4562.11	2.92	**0.00**	**0.00**	2959.38	1.87	4.79	17.01	1816.66	2.34
20	GN	1.94	8.00	3099.76	3.19	1.60	8.00	3537.04	2.25	0.13	1.00	**1904.77**	1.49	5.55	13.01	1826.08	2.65
	GT	1.90	7.00	3165.70	3.22	1.78	8.00	3555.19	2.25	0.15	1.00	1953.75	1.51	4.85	15.01	1855.72	2.70
	GS	1.83	8.00	3459.50	3.50	1.80	8.00	3608.65	2.26	0.14	1.00	1986.81	**1.54**	5.04	15.01	**1789.58**	2.62
	GNN	1.78	10.00	**2935.17**	**3.03**	1.74	9.00	**3340.20**	**2.15**	0.45	3.00	2467.33	**1.54**	4.32	13.01	1817.48	**2.52**
	GNT	1.16	6.00	3382.37	3.46	0.91	5.00	3895.66	2.57	**0.00**	**0.00**	3249.45	1.99	**3.73**	**14.01**	2025.77	2.76
	GNS	**0.62**	**3.00**	3400.57	3.45	**0.64**	**4.00**	4303.16	2.93	0.08	2.00	2809.01	1.82	5.15	16.01	2232.47	2.61
100	GN	0.83	4.00	3368.76	3.72	4.02	17.00	**2750.14**	**1.91**	0.21	4.00	2017.47	**1.40**	5.82	14.02	2121.12	2.77
	GT	0.82	**3.00**	3384.61	3.73	3.99	17.00	2756.44	**1.91**	0.26	2.00	**1976.76**	1.42	5.75	14.02	2089.83	2.77
	GS	**0.37**	2.00	3667.47	4.11	4.22	15.00	2796.50	1.92	0.29	3.00	2031.83	1.47	6.06	14.02	1921.69	2.67
	GNN	1.84	11.00	**2432.55**	**2.99**	1.37	5.00	3852.90	2.51	0.10	1.00	2301.40	1.46	4.58	15.01	**1759.03**	2.48
	GNT	0.71	4.00	3317.81	3.76	**0.20**	**1.00**	6294.23	4.05	**0.00**	**0.00**	3122.89	1.88	**3.48**	**14.01**	1974.79	2.57
	GNS	0.90	6.00	2765.77	3.35	0.63	5.00	4760.92	3.15	0.27	3.00	2679.73	1.78	4.46	15.01	1925.93	**2.45**

Note: GN - GARCH with normally distributed innovations (d.i.); GT - GARCH with t d.i.; GST - GARCH with skewed t d.i.; GNN - GARCHNet with normally d.i.; GNT - GARCHNet with t d.i.; GNS - GARCHNet with skewed t d.i.;
Minimum cost values for each period and p pairs are in bold.

like to emphasize that it performs well in any conditions.

We notice that the length of the sequence has a nonlinear effect on the quality of the model. It mostly depends on the volatility of the sample that was used for the prediction, mostly in case of standard GARCH model - see outstanding exceptions for $p = 100$ in sample starting in 2007 and much lower values for remaining values of p. Such effect is diminished in the case of GARCHNet, however we still notice large discrepancies for different values of p. In terms of the proposed length p, we would suggest 20, which is roughly 4 weeks - one trading month and therefore it has given the most remarkable results.

In the table 2 we have presented the values of cost functions. We notice that due to a lower number of exceptions of GARCHNet models, the values of regulator's cost functions are lower than for its equivalents in most of the analyzed cases, furthermore - the worst GARCHNet approaches are often better than the best GARCH ones. This is a much desired trait of the VaR model, because in case of the exception the potential loss is not that severe. However from the firm's viewpoint GARCHNet models don't look so bright. In most of the cases the values of firm's cost functions are the worst - only for few instances the value of cost function for GARCHNet model was lower. This is rather undesirable behavior given the usage of nonlinear approach. GARCHNet with normal distribution seems to have the lowest value of ABLLF cost function among GARCHNet models and usually it can compete with the same cost function calculated for its equivalent from GARCH family. In conclusion, based on the results of cost functions, we assume that GARCHNet at this stage is relatively conservative model.

5 Conclusion

In this paper we have proposed a new approach to conditional variance's specification in GARCH models - a GARCHNet model that incorporates a simple long short term memory neural network. The idea behind GARCHNet is that the neural network can easily approximate nonlinear relations and these are by far the most often seen in the volatility of financial markets. In addition to that, the simplicity of the GARCH maximum likelihood estimation allows to use original log likelihood functions as cost functions in the GARCHNet neural network model. We have proposed three different GARCHNet models, each with a different assumed innovation distribution: normal; t; and skewed t. The neural network that is used as the specification of the conditional variance is rather

small. It incorporates a LSTM layer as an input followed by three fully connected layers. In case the assumed distribution requires parameters other than mean and variance, these are optimized by the very same neural network.

GARCHNet models were contrasted with the original GARCH models in an empirical study. We have created Value-at-Risk estimates using rolling window approach - we have trained the model using 1000 observations and estimated a forecast, then we moved one timestep forward and prapared another one. Such procedure was repeated 250 times. For the data we have used log returns from price quotations of WIG20 (Warsaw Stock Exchange; Poland).

Our results show that the GARCHNet is a prominent model that can explain conditional variance at least on the same level as traditional approaches. Value-at-Risk forecasts are rather conservative, however due to that fewer exceptions are observed. GARCHNet would more likely be chosen by regulatory bodies than by firm's management itself due to relatively higher alternative cost.

We can see several options to enhance the model already:

1. The best lenght of sequence p

   p is one of the most influential parameters for the model as it defines the amount of information one forecast includes, however we did not notice any trends that would determine its impact on the quality of the model.

2. Stopping criterion

   Given that the validation sample is absent in the case of an immediate forecast (timestep to timestep), the available options to objectively define the end of the training phase of the model are depleted. We believe that in the case of VaR a stopping criterion based on the statistical tests would be an accurate one.

3. Neural network architecture and hyper-parameter tuning

   The neural network proposed in here is rather small. We believe that an increase in the number of parameters (hence NN's depth) would positively affect the quality of the model. Moreover we haven't covered any hyperparameter tuning - most of them were assumed, not tested.

4. Another approach to estimation of the distribution's parameters

The parameters of the distribution are now estimated by a separate layer that is dependent on the previous layers. Worsened results of the t distributed and skewed t distributed GARCHNet might be only an effect of the assumed approach. Two other options that could be considered are either separate neural networks to estimate additional parameters optimized in one procedure (forecast sample dependent parameters); or inclusion of these parameters as separate weights to be optimized (training sample dependent parameters).

5. Possible extension of this approach to time-series

   Given that GARCH models are not only employed in VaR modeling, we are mostly interested in GARCHNet performance in traditional time-series forecasting.

6. 'One (model) to rule them all...'

   Our study assumes the rolling window estimation approach. However, we believe that frequent updates of the model might not necessarily improve its quality, while only increasing the time overhead in training. We believe that the model can be fully reset (random weights fully initialized) less often than with each timestep forecast. The model might be refitted between resets to include new information. The most extreme approach would assume that it is only trained fully once (on the first 1000 observations) and than it is only refitted to include new timesteps.

References

Abad, P. and S. Benito (2013). A detailed comparison of value at risk estimates. *Mathematics and Computers in Simulation 94*, 258–276.

Abad, P., S. Benito, and C. López (2014). A comprehensive review of value at risk methodologies. *The Spanish Review of Financial Economics 12*(1), 15–32.

Abad, P., S. Muela, and C. Lopez (2015). The role of the loss function in value-at-risk comparisons. *Journal of Risk Model Validation 9*, 1–19.

Aloui, C. and S. Mabrouk (2010). Value-at-risk estimations of energy commodities via long-memory, asymmetry and fat-tailed garch models. *Energy Policy 38*(5), 2326–2339.

Angelidis, T., A. Benos, and S. Degiannakis (2004). The use of garch models in var estimation. *Statistical Methodology 1*(1), 105–128.

Arnerić, J., T. Šestanović, and Z. Aljinović (2014). Garch based artificial neural networks in forecasting conditional variance of stock returns. *Croatian Operational Research Review 5*, 329–343.

Barone-Adesi, G., R. F. Engle, and L. Mancini (2008). A garch option pricing model with filtered historical simulation. *The review of financial studies 21*(3), 1223–1258.

BenSaïda, A. (2015). The frequency of regime switching in financial market volatility. *Journal of Empirical Finance 32*, 63–79.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics 31*(3), 307–327.

Bonato, M. (2012). Modeling fat tails in stock returns: a multivariate stable-garch approach. *Computational Statistics 27*(3), 499–521.

Chen, S. and S. A. Billings (1992). Neural networks for nonlinear dynamic system modelling and identification. *International Journal of Control 56*(2), 319–346.

Chlebus, M. (2017). Ews-garch: New regime switching approach to forecast value-at-risk. *Central European Economic Journal 3*(50), 1–25.

Cho, K., B. van Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation.

Chorowski, J. and J. M. Zurada (2014). Learning understandable neural networks with nonnegative weight constraints. *IEEE transactions on neural networks and learning systems 26*(1), 62–69.

Christoffersen, P. F. (1998). Evaluating interval forecasts. *International economic review*, 841–862.

Cont, R. (2002). Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance 1*, 223–236.

Costanzino, N. and M. Curran (2018). A simple traffic light approach to backtesting expected shortfall. *Risks 6*(1).

Degiannakis, S., C. Floros, and A. Livada (2012). Evaluating value-at-risk models before and after the financial crisis of 2008: International evidence. *Managerial Finance 38*, 436–452.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*.

Duffie, D. and J. Pan (1997). An overview of value at risk. *Journal of derivatives 4*(3), 7–49.

Engle, R. F. and T. Bollerslev (1986). Modelling the persistence of conditional variances. *Econometric Reviews 5*(1), 1–50.

Engle, R. F. and S. Manganelli (2004). Caviar: Conditional autoregressive value at risk by regression quantiles. *Journal of business & economic statistics 22*(4), 367–381.

Ergün, A. T. and J. Jun (2010). Time-varying higher-order conditional moments and forecasting intraday var and expected shortfall. *The Quarterly Review of Economics and Finance 50*(3), 264–272.

Escanciano, J. C. and J. Olmo (2010). Backtesting parametric value-at-risk with estimation risk. *Journal of Business and Economic Statistics 28*(1), 36–51.

Fama, E. F. (1998). Market efficiency, long-term returns, and behavioral finance1the comments of brad barber, david hirshleifer, s.p. kothari, owen lamont, mark mitchell, hersh shefrin, robert shiller, rex sinquefield, richard thaler, theo vermaelen, robert vishny, ivo welch, and a referee have been helpful. kenneth french and jay ritter get special thanks.1. *Journal of Financial Economics 49*(3), 283–306.

Francq, C. and J.-M. Zakoïan (2004). Maximum likelihood estimation of pure garch and arma-garch processes. *Bernoulli 10*(4), 605–637.

Glosten, L. R., R. Jagannathan, and D. E. Runkle (1993a). 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.

Glosten, L. R., R. Jagannathan, and D. E. Runkle (1993b). 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.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. The MIT Press.

Hansen, B. E. (1994). Autoregressive conditional density estimation. *International Economic Review 35*(3), 705–730.

Hochreiter, S. and J. Schmidhuber (1997). Long short-term memory. *Neural Comput. 9*(8), 1735–1780.

Jeong, Y. and S. Lee (2019). Recurrent neural network-adapted nonlinear arma-garch model with application to s&p 500 index data. *Journal of the Korean Data and Information Science Society 30*(5), 1187–1195.

Kim, H. Y. and C. H. Won (2018). Forecasting the volatility of stock price index: A hybrid model integrating lstm with multiple garch-type models. *Expert Systems with Applications 103*, 25–37.

Kristjanpoller, W. and M. C. Minutolo (2015). Gold price volatility: A forecasting approach using the artificial neural network–garch model. *Expert Systems with Applications 42*(20), 7245–7251.

Kristjanpoller, W. and M. C. Minutolo (2016). Forecasting volatility of oil price using an artificial neural network-garch model. *Expert Systems with Applications 65*, 233–241.

Kupiec, P. (1995). Techniques for verifying the accuracy of risk measurement models. *The J. of Derivatives 3*(2).

Lim, B., S. Arik, N. Loeff, and T. Pfister (2019). *Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting*.

Liu, W. and M. So (2020). A garch model with artificial neural networks. *Information 11*, 489.

Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller (2013). Playing atari with deep reinforcement learning.

Nelson, D. and C. Cao (1992). Inequality constraints in the univariate garch model. *Journal of Business & Economic Statistics 10*, 229–35.

Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica 59*(2), 347–370.

Nguyen, N., M.-N. Tran, D. Gunawan, and R. Kohn (2019). *A long short-term memory stochastic volatility model*.

Nikolaev, N., P. Tino, and E. Smirnov (2011). Time-dependent series variance estimation via recurrent neural networks. In T. Honkela, W. Duch, M. Girolami, and S. Kaski (Eds.), *Artificial Neural Networks and Machine Learning – ICANN 2011*, pp. 176–184. Springer Berlin Heidelberg.

Nozari, M., S. Raei, P. Jahangiry, and M. Bahramgiri (2010). A comparison of heavy-tailed estimates and filtered historical simulation: Evidence from emerging markets. *6*, 347–359.

Omari, C., S. Mundia, and I. Ngina (2020). Forecasting value-at-risk of financial markets under the global pandemic of covid-19 using conditional extreme value theory. *Journal of Mathematical Finance Vol.10No.04*, 28.

Pascanu, R., T. Mikolov, and Y. Bengio (2012). On the difficulty of training recurrent neural networks. *30th International Conference on Machine Learning, ICML 2013*.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala (2019). Pytorch: An imperative style, high-performance deep learning library. pp. 8024–8035.

Philippe, J. (2006). *Value at Risk, 3rd Ed*. McGraw-Hill.

Rothfuss, J., F. Ferreira, S. Walther, and M. Ulrich (2019). *Conditional Density Estimation with Neural Networks: Best Practices and Benchmarks*.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning representations by back-propagating errors. *Nature 323*(6088), 533–536.

Segal, G., I. Shaliastovich, and A. Yaron (2015). Good and bad uncertainty: Macroeconomic and financial market implications. *Journal of Financial Economics 117*(2), 369–397.

Sezer, O. B., M. U. Gudelek, and A. M. Ozbayoglu (2020). Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. *Applied Soft Computing 90*, 106181.

So, M. K. and L. Philip (2006). Empirical analysis of garch models in value at risk estimation. *Journal of International Financial Markets, Institutions and Money 16*(2), 180–197.

Stooq (2021). Historical data: Wig20 (wig20). Data retrieved from Stooq, `https://stooq.pl/q/d/l/?s=wig20&i=d`.

Vorbrink, J. (2014). Financial markets with volatility uncertainty. *Journal of Mathematical Economics 53*, 64–78.

Wang, Z.-R., X.-H. Chen, Y.-B. Jin, and Y.-J. Zhou (2010). Estimating risk of foreign exchange portfolio: Using var and cvar based on garch–evt-copula model. *Physica A: Statistical Mechanics and its Applications 389*(21), 4918–4928.

Zanin, L. and G. Marra (2012). Rolling regression versus time-varying coefficient modelling: An empirical investigation of the okun's law in some euro area countries. *Bulletin of Economic Research 64*(1), 91–108.

University of Warsaw
Faculty of Economic Sciences
44/50 Długa St.
00-241 Warsaw
www.wne.uw.edu.pl