EC331: Research in Applied Economics

Old Dogs and New Tricks: Log Return Density Forecasting Using a Novel Neural Network Architecture and ARMA-GARCH Models

Joseph Basford*
ID: u1909163

Abstract

Uncertainty is fundamental in economics. Imperfect or asymmetric information can lead to marked resource misallocations, and their ubiquity is a hard truth in many economies. Whilst in the past, econometricians focused on point and interval forecasts, developments in quantitative finance and risk management have increased demand for quick and accurate forecasts of the whole probability distribution. However, accuracy and speed can be conflicting goals, particularly when nonlinearities in the data mean complex models are required to predict the probability density function accurately. In this paper, I develop a novel semiparametric neural network architecture and compare its predictive accuracy to a variety of ARMA-GARCH models on the log returns of the NASDAQ composite, Nikkei 225, and DAX stock indices. Using a multivariate extension of the Giacomini-White test of equal predictive accuracy, a large initial model set of neural network and ARMA-GARCH models is synthesised down to a model confidence set. ARMA-GARCH models are found to outperform the neural network models across all three series, and results are robust to variations in the testing specification and choice of scoring rule.

Word Count: 4,993

*I would like to thank my supervisor, Han Zhang, for her invaluable help and guidance throughout the project. I would also like to thank Daniel Borup for making the Matlab code for constructing the MCS using the multivariate Giacomini-White test available on his website: `https://sites.google.com/view/danielborup/research`. Additionally, I thank the GitHub user "ogrnz" whose repository translates the Matlab code into Python (available here: `https://github.com/ogrnz/feval`). All code used, and some supplementary technical details, is available on the companion Github repository for this project: `https://github.com/Joe-Basford/RAE`.

Contents

1 Introduction

Forecasting is inherent in almost all economic decision-making. Whether it is macroeconomic regulators setting interest rates or parents saving for their child's tuition, expectations of the future form the basis of agents' decisions. Inaccurate forecasts, even under a rationality assumption, can lead to market failure as agents cannot accurately and confidently formulate a model for the future state of the world[1]. Therefore, forecasting stochastic events well is particularly interesting in econometrics and finance, as good forecasts can dampen information effects.

Previously, the forecasting literature focused on point and interval forecasts. Little practical use was seen in density forecasting, and initial attempts to create tractable density forecasting techniques required restrictive assumptions, such as no parameter uncertainty or Gaussian innovations (Diebold et al., 1998). However, with the emergence of quantitative finance and risk management, the appetite for accurate density forecasts has grown, leading to a burgeoning literature on the subject. Computer technology and simulation advancements have resulted in usable density forecasts under minimal requirements. Boero and Marrocu (2004) noted that at the time of their writing, over half of all inflation-targeting banks were using some form of density forecasting. The number of macroeconomic policymakers utilising density forecasting has likely grown significantly since then, and Value-at-Risk analysis - with density forecasting at its core - has become foundational in modern risk management.

Large datasets coupled with new computer technologies have meant that complex and highly nonlinear models for density forecasting can now be accurately estimated. Intuitively, one would expect nonlinear models to produce better forecasts than linear models, given thresholds, capacity constraints, and other asymmetries which engender nonlinear decision boundaries (Dahl and Hylleberg, 2004). Despite this, nonlinear models do not significantly outperform linear models, such as ARMA-GARCH, and may produce systematically worse forecasts (De Gooijer and Kumar, 1992; Clements et al., 2004). The issue with highly nonlinear models is re-

[1] Although an inability to forecast well may represent a violation of rationality rather than imperfect information, I find it more plausible that many agents do not know how to quickly and easily forecast well do would do so if they could.

lated to the "curse of dimensionality" as slight perturbations to the underlying data-generating process can impact forecast accuracy significantly more in high-dimensional models than in smaller models (Diakonikolas et al., 2018).

However, previous comparisons of linear and nonlinear models are restricted as they focus on point forecasts (Dahl and Hylleberg, 2004), only consider regime-changing models (Clements et al., 2004), or arbitrarily discretise the outcome space to convert a function approximation problem to a classification task (Yeo et al., 2018). To fill this literature gap, I develop a novel neural network (NN) architecture based on a series of Long Short-Term Memory (LSTM) blocks (Hochreiter and Schmidhuber, 1997) which allows for a continuous outcome space. The paper compares ARMA-GARCH models of order $(p, p) - (p, p)$, for $p = 1, 2, 3, 4, 5$, with variants of the NN on the log returns of three globally significant stock indices, viz. the NASDAQ composite ($\hat{}IXIC$), the DAX ($\hat{}GDAXI$), and Nikkei 225 ($\hat{}N225$). The NN model differs from other techniques used for density forecasting, which use nonparametric, fully parametric, or classification models (Zhang et al., 2022; Guo et al., 2018; Yeo et al., 2018). Instead, the NN model I propose assumes an underlying parametrisation of the distribution a priori and parameters of the distribution are then estimated using a NN. Thus, the NN architecture is semiparametric; although an underlying distribution for the data is assumed in each model, increasing the breadth of assumed distributions (e.g. assuming a parametric family of distributions rather than individual ones) and the width of the NN decreases the restrictions on the data generating process. Indeed, as the width of a NN tends to infinity, it can be considered a Gaussian Process: a nonparametric model (Lee et al., 2018).

At the inference stage, I test the null hypothesis (H_0) of equal conditional performance of the *class* of NN models, and the *class* of ARMA-GARCH models against the two alternative hypotheses that either the NN class performs better (H_1) or the ARMA-GARCH class performs best (H_2). I use a multivariate version of the Giacomini-White test (Borup et al., 2022; Giacomini and White, 2006) of equal predictive performance to construct a model confidence set (MCS) (Hansen et al., 2011). If both ARMA-GARCH and NNs are in the MCS (H_0) will be accepted, whilst if only NN or ARMA-GARCH models remain in the MCS, the null will be rejected in favour of (H_1) or (H_2), respectively.

2 Literature Review

Bollerslev (1986) first proposed GARCH models as an extension of the ARCH models proposed by his PhD supervisor Engle (1982). Whilst ARCH models struggle to model long-memory series due to a linearly declining lag structure, GARCH models permit a parsimonious estimation of more flexible lag structures for long-memory series. Further, ARMA-GARCH models combine ARMA and GARCH models by simultaneously estimating parameters for the conditional mean and variance of the time series. Although originally formulated under normally distributed innovations, ARCH and GARCH models can accommodate other distributions. Perhaps most relevant for my purposes, Bollerslev (1987) formulated the GARCH model under T-distributed innovations, allowing for kurtosis and skew to be parsimoniously estimated.

There is an extensive literature studying ARMA-GARCH models, and variants abound. Several seek to account for leverage effects, whereby negative innovations have larger impacts on future volatility relative to positive innovations (Nelson, 1991; Engle, 2015; Engle and Bollerslev, 1986; Glosten et al., 1993, *inter alia.*). The presence of nonlinearities in the series appears necessary for leverage effect models to offer tangible benefits in predictive accuracy. Hansen and Lunde (2005) found no evidence that a GARCH(1,1) model was outperformed by models including leverage effects when comparing 330 ARCH-type models on DM-$ exchange rate data. However, they found that leverage effect GARCH models significantly outperformed the GARCH(1,1) model on IBM return data, indicating that the specifics of the data-generating process are relevant when choosing between leverage effect models and standard ARMA-GARCH models.

There are two categories of test for equal predictive performance:

1. Unconditional Tests
    - Which forecast historically had the highest predictive accuracy on average?
2. Conditional Tests
    - Which forecast will have the highest predictive accuracy given what we know about the process?

Initial tests for equal predictive performance restricted attention to particular loss functions and unconditional predictive performance (Granger and Newbold, 1977; Leitch and Tanner, 1991; West et al., 1993; Harvey et al., 1997). Later, Diebold and Mariano (2002) generalised these tests by allowing for non-symmetric, non-quadratic loss functions and non-Gaussian, non-mean zero, as well as serially and contemporaneously correlated forecast errors. Giacomini and White (2006) further generalised the Diebold-Mariano test to one of equal conditional predictive performance and accounted for the effect of estimation uncertainty on predictive performance whilst allowing for the forecasts to come from nested and non-nested models. Borup et al. (2022) introduced a natural multivariate extension of the Giacomini-White test, which simultaneously tests many competing forecasts. As one would intuitively expect, the multivariate Giacomini-White test reduces down to the multivariate Diebold-Mariano test when conditioning on the trivial σ-algebra (see Section 4.3 for details).

There are several scoring rules/loss functions (see Definition 1) used to rate forecast accuracy, and the most common are summarised in Gneiting and Raftery (2007). A minimal requirement is **strict properness** of the score, i.e. the true density should be ranked strictly higher than any other (cf. Definition 2). Other desirable properties include consistency and elicitability, which ensure forecasts are coherent with decision theory (Savage, 1971). However, as consistency and elicitability concern point forecasts, I do not focus on these criteria when selecting a scoring rule. An important example of a strictly proper score is the logarithmic score (Definition 3) (Mitchell and Hall, 2005; Amisano and Giacomini, 2007; Bao et al., 2007). The logarithmic score is equivalent to the Kullback-Leibler information criterion which converts the Giacomini-White test statistic into the likelihood ratio test of Berkowitz (2001), with the desirable power properties following from Neyman-Pearson theory. Other strictly proper scores include the spherical and quadratic scores (Definitions 4 and 5) (Boero and Marrocu, 2004; Diks et al., 2011; Amisano and Giacomini, 2007). Unlike the logarithmic score, these scores can accommodate null events, which can help during NN training. I restrict attention to the logarithmic, quadratic, and spherical scores, as they are computationally tractable and common in the literature.

Empirical studies find that nonlinear models do not consistently outperform linear models in

point, interval, or density forecasting (De Gooijer and Kumar, 1992; Clements et al., 2004). However, there may be particular periods over which nonlinear models significantly outperform linear models due to nonlinearities (Clements et al., 2004; Tong et al., 1995). Some applied papers posit that NNs may outperform ARMA-GARCH models in density forecasting due to nonlinearity in the time series itself or in its dependency structure over time (Goulet et al., 2022; Park et al., 2017; Taylor and Buizza, 2004). However, Park et al. (2017) use squared prediction error as their loss function, which is known not to be strictly proper since any forecast with the same conditional mean will attain the same expected score. Goulet et al. (2022) do not directly compare ARMA-GARCH and NN models. Yeo et al. (2018) adapt an LSTM model for density forecasting by discretising the estimated density function to transition from a function approximation problem to a classification problem. Dahl and Hylleberg (2004) compared a linear model to a series of nonlinear models using US employment data and found that the nonlinear models outperformed the linear model, but their linear model only modelled the conditional mean of the series and did not account for higher moments, which may be relevant to the series. Hence, my paper fits into the literature comparing linear and nonlinear models by directly comparing ARMA-GARCH predictive performance to a broad class of nonlinear models due to the expressiveness of neural nets.

3 The Data

The three stock indices considered are the NASDAQ composite ($\hat{}IXIC$), the DAX ($\hat{}GDAXI$), and Nikkei 225 ($\hat{}N225$). Data was obtained from Yahoo Finance on 27/11/2021. As is standard in the literature (Wei et al., 2022), I focus on the log returns of adjusted closing prices since these best reflect the market valuation of the asset by eliminating noise due to variation in dividend structure. The log return series, $(\text{LOG-RETURN}_t)_{t\geq 1}$, can be calculated using the adjusted close price series, $(\text{ADJ-CLOSE}_t)_{t\geq 0}$, through the following relation

$$\text{LOG-RETURN}_t = \log\left(\frac{\text{ADJ-CLOSE}_t}{\text{ADJ-CLOSE}_{t-1}}\right)$$

I use the log return transform rather than other transforms of ADJ-CLOSE as log returns are stationary (otherwise, growth is exponential) and relevant for those invested in the stock market and agents in the wider macroeconomy. Although general market conditions will likely affect log returns (Fama et al., 1969), I do not factor these into the models due to limits on computational power whilst training the NN model. The Nikkei 225 was available from 05/01/1995-27/11/2021, the NASDAQ composite from 05/02/1971-27/11/2021, and the DAX from 30/12/1987-27/11/2021. The data were split on 2015/01/01 as this gave a training/validation set split of around 80/20 in all three series, with log returns before this constituting the training set and data after 2015/01/01 constituting the validation set for inference. Summary statistics for the three series across the training and validation sets can be found in Table 1, and the data are illustrated in Figure 1.

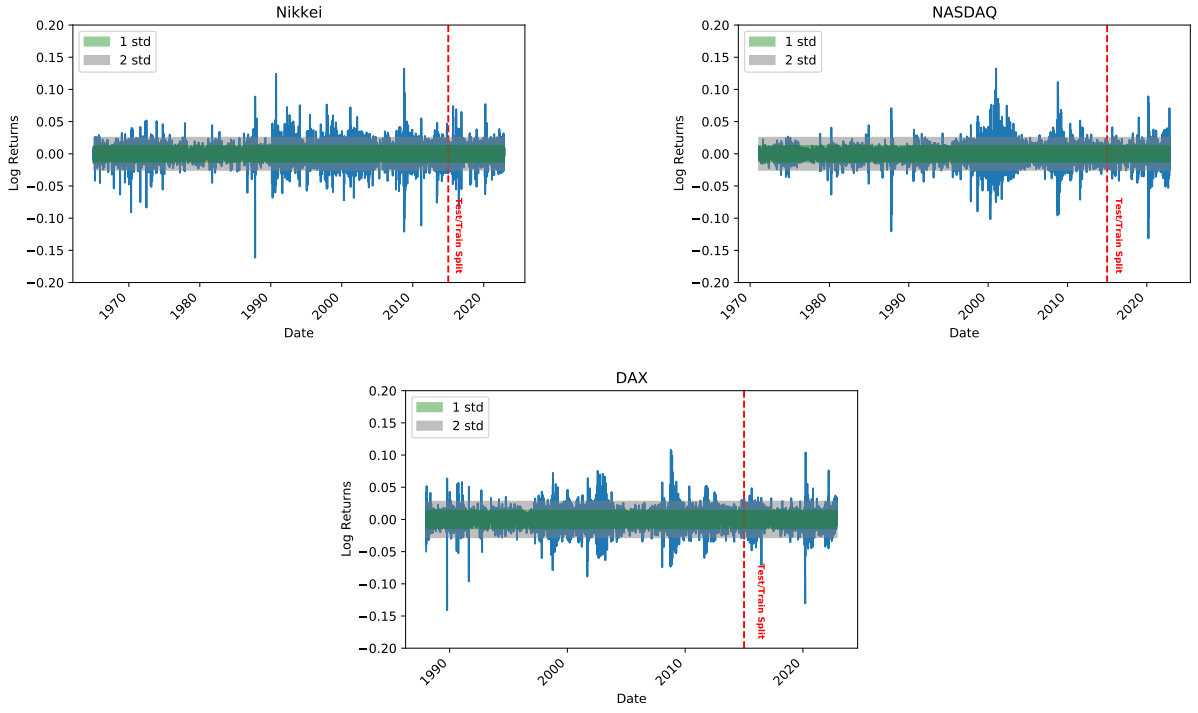
Splitting at 2015/01/01 means the three series have varying training data sizes: the Nikkei has a training/validation set split of about 86/14, whilst for the NASDAQ composite it is 85/15, and for the DAX it is 77/23. The difference in training set sizes should give some indication as to how data size affects predictive accuracy. I expect larger data to benefit the NN models more due to their size relative to the ARMA-GARCH models. Hence, if there are differences in the relative performances of models across the datasets, it is likely due to differences in training set size.

Table 1: Summary Statistics

	Nikkei			NASDAQ			DAX		
	Training	Validation	Combined	Training	Validation	Combined	Training	Validation	Combined
count	12307	1929	14236	11076	1990	13066	6818	2004	8822
mean	0.0002	0.0003	0.0002	0.0003	0.0004	0.0003	0.0003	0.0002	0.0003
std	0.0126	0.0129	0.0127	0.0125	0.0139	0.0127	0.0143	0.0130	0.0140
min	-0.1614	-0.0825	-0.1614	-0.1205	-0.1315	-0.1315	-0.1409	-0.1305	-0.1409
25%	-0.005	-0.0058	-0.0055	-0.0045	-0.0049	-0.0046	-0.0066	-0.0054	-0.0064
50%	0.0004	0.0007	0.0004	0.0011	0.0010	0.0011	0.0008	0.0008	0.0008
75%	0.0063	0.0069	0.0063	0.0060	0.0073	0.0062	0.0077	0.0068	0.0075
max	0.1323	0.0773	0.1323	0.1325	0.0893	0.1325	0.1080	0.1041	0.1080

Statistics given to 4 decimal places. For values with 'small' absolute value (<0.01), the log return value is approximately the percentage return.

Figure 1: Log Stock Return Data



Training and test sets split on 2015-01-01

One problem with splitting the data on 2015-01-01 is the potential of nonlinearities which are only present after 2015-01-01. These could result in the NN models performing worse than expected relative to the ARMA-GARCH models as NN models perform optimally after being trained on similar nonlinear data whilst the ARMA-GARCH model has no mechanism for nonlinear modelling. However, given the training sets include periods of likely regime change, such as the 2008 financial crisis, the 1997 Asian financial crisis, and the 1973 and '79 oil crises

(besides the DAX), the problem is lessened. Nevertheless, if the regime changes after 2015 are idiosyncratic and previous regime changes give no information about features of future regime changes, then conclusions drawn about the performance of the ARMA-GARCH and NN models may have high generalisation error. For this reason, I make some strong assumptions regarding the structure of the series for inference (see Appendix A).

4 Methodology

4.1 Description of the Environment

Suppose there is a complete probability space $(\Omega, \mathscr{F}, \mathbb{P})$ and a stochastic process $\{Z_t : \Omega \to \mathbb{R}^{n+1}\}_{t=1}^{T}$ comprised of $(\boldsymbol{X}_t, Y_t) \in \mathbb{R}^n \times \mathbb{R}$, where $Y_t : \Omega \to \mathbb{R}$ is the log return at time t and $\boldsymbol{X}_t : \Omega \to \mathbb{R}^n$ is an input sequence of predictors. Assuming all information available to the forecaster at time t is encoded into $(Z_i)_{i=1}^{t}$, I work with the natural filtration $(\mathscr{F}_t)_{t\geq 0} = (\sigma(\cup_{i=1}^{t}\sigma(Z_i)))_{t\geq 0}$, to which (Z_t) is adapted. Throughout I implicitly work on the filtered space $(\Omega, \mathscr{F}, (\mathscr{F}_t)_{t\geq 0}, \mathbb{P})$. I consider one-step ahead forecasts using a rolling window method since this is computationally tractable both from a model fitting and inference perspective.

It is necessary to make the following assumptions whilst modelling (see Appendix A for details).

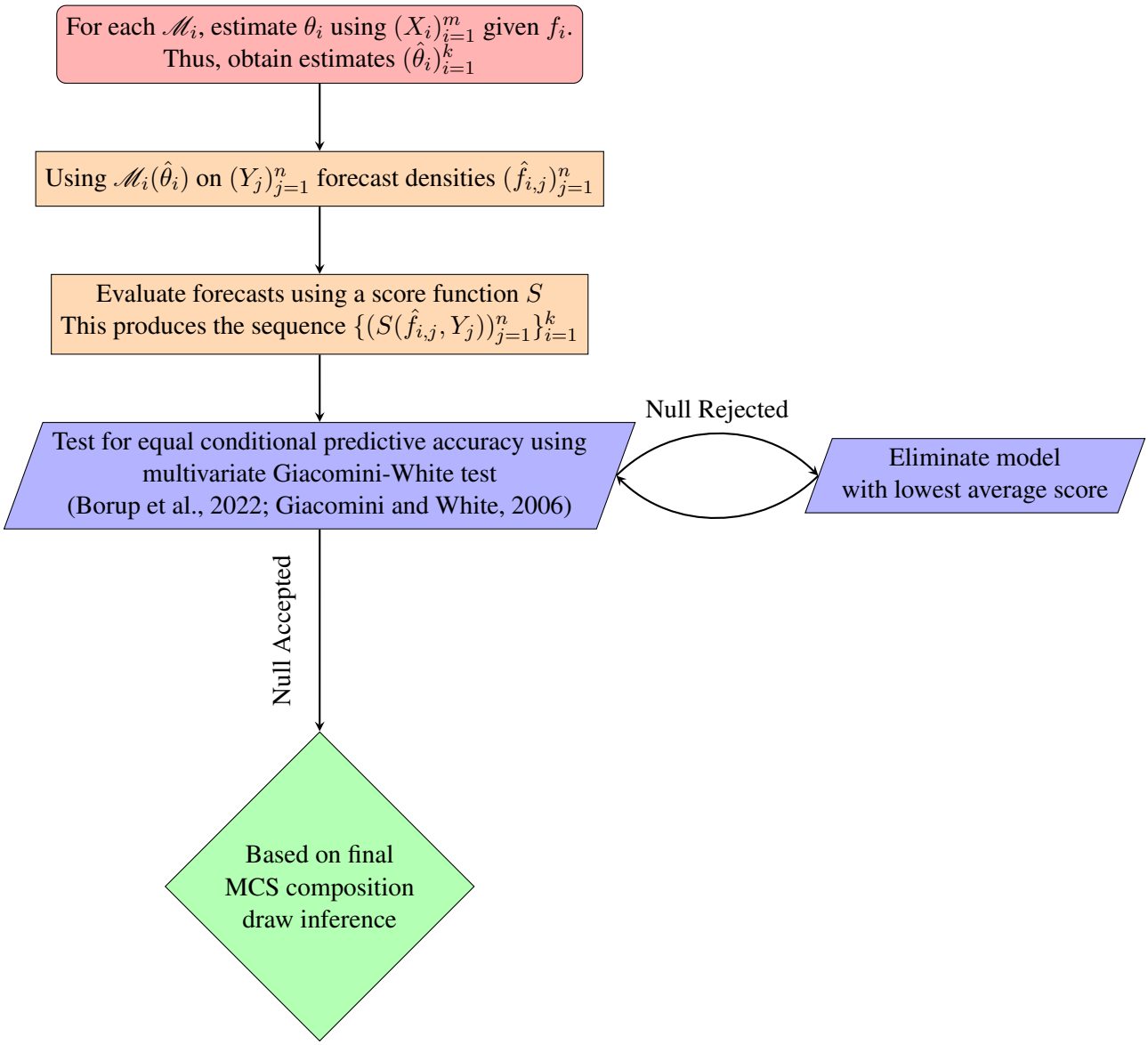
1. The distribution of log returns is absolutely continuous, i.e. there exists a probability density function by the Radon-Nikodym theorem.
2. For all time steps, the random variable representing log returns is $\mathscr{L}^2$
3. Structural breaks are not idiosyncratic over the validation set
4. There exists some parametrisation for the process distribution

In the ARMA-GARCH and NN models, I assume some distribution of the innovations. Let $\mathscr{C}$ denote a convex hull of probability density functions. I selected the Normal, Laplace, and noncentralised Student-T distributions as vertices of $\mathscr{C}$. These distributions offer variety in the flexibility of kurtosis estimation. The normal distribution has 0 excess kurtosis, whilst the Laplace has an excess kurtosis of 3, and the noncentralised Student-T has an excess kurtosis equal to $\frac{\nu}{\nu-2}$, assuming $\nu > 2$, where ν is the degrees of freedom of the distribution. Further, as the normal distribution has maximal entropy across all possible distributions over $\mathbb{R}$, it is a natural baseline as it represents the distribution with minimal structure beyond the second moment. These distributions provide sufficient variety in our density forecasts whilst also balancing computational tractability, particularly in the ARMA-GARCH model where a maximum likelihood function must be derived.

A high-level overview of my methodology is provided in Figure 2. See the rest of the methodology for details.

Figure 2: Organisation of Methodology

Suppose we have a set of models $(\mathscr{M}_i)_{i=1}^k$ with corresponding assumed distributions $(f_i)_{i=1}^k$ and parametrisations $(\theta_i)_{i=1}^k$. Suppose also that we have some training data $(X_i)_{i=1}^m$ and validation data $(Y_j)_{j=1}^n$.



4.2 The Score Function

I fit the model to training data for each model and assumed distribution. For the ARMA-GARCH model, I use maximum likelihood estimation whilst the NN model is trained via back-propagation through time (BPTT, see 4.5 for details). One must specify a loss function to perform BPTT and evaluate forecast performance. As a loss function, I use the negative of a scoring rule. A scoring rule is defined as follows,

Definition 1. (Scoring Rule) A **scoring rule** is a function $S : \mathscr{C} \times \mathbb{R} \to \mathbb{R}$ assigning a value indicating the accuracy of the density forecast versus the observed value.

I select the score function based on two criteria. First, the score function should satisfy the following strict properness property,

Definition 2. (Properness and Strict Properness) Suppose the true density forecasting function is $p_t \in \mathscr{C}$. A scoring rule $S : \mathscr{C} \times \mathbb{R} \to \mathbb{R}$ is **proper** if $\forall f_t \in \mathscr{C}$,

$$\mathbb{E}[S(p_t, y_{t+1}) - S(f_t, y_{t+1})] \geq 0$$

S is **strictly proper** if $\forall f_t \in \mathscr{C}$ such that $p_t \neq f_t$ on some set of positive Lebesgue measure,

$$\mathbb{E}[S(p_t, y_{t+1}) - S(f_t, y_{t+1})] > 0$$

And $\mathbb{E}[S(p_t, y_{t+1}) - S(f_t, y_{t+1})] = 0 \iff f_t = p_t$ a.e..

Second, the score function should maximise the power when constructing the MCS (see Section 4.3). The log score function, $S(f, y) = \log f(y)$, satisfies these conditions and has the desirable property that the difference of the log scores for any two competing density forecasts is the difference of their Kullback-Leibler Information Criteria (Amisano and Giacomini, 2007). Therefore, the log score will be the primary score function used throughout. Strict properness of the log-score is derived in Appendix B.2.1 and the power properties of the log-score in the Giacomini-White tests follows from Neyman-Pearson theory of likelihood ratio tests (Berkowitz, 2001; Diks et al., 2011). Although the score function used to fit the NN may

differ from the one used for inference, I have chosen to primarily consider the case when these are the same to ease the computations. Although the ARMA-GARCH and NN models seek to maximise the log score when fitting, results may have high generalisation error when using other score functions due to the potentially unbounded log loss for any given observation. To test the robustness of my results to different score functions, I additionally use the spherical and quadratic scores at the hypothesis testing stage. Strict properness of the spherical and quadratic scores is shown in Appendix B.2.2 and Appendix B.2.3, respectively. However, all three of these scores, suffer from the same problem when focusing on a particular region of interest (for example the left tail in insurance claims) as they place excessive mass in the region of interest. Whilst there exist scores bypassing this issue (e.g. the Continuous Ranked Probability Score (Gneiting and Raftery, 2007) or censored likelihood scores (Diks et al., 2011)), I do not consider them in this paper.

4.3 Misspecification and the MCS

A potential source of bias is model misspecification. Pairwise model comparisons from the NN or ARMA-GARCH model classes are problematic as any individual model could be misspecified. Further, iterative pairwise comparisons can lead to erroneous conclusions as the set of models considered 'best' depends on the comparison order. To limit the misspecification issue, I construct a sizable initial model set, $\mathscr{M}^0$, with many ARMA-GARCH and NN models which may be individually misspecified. Then, following the procedure of Hansen et al. (2011), models are iteratively removed to obtain a sequence $\mathscr{M}^0 \supset \mathscr{M}^1 \supset \cdots \supset \mathscr{M}^n$, where $\mathscr{M}^n$ contains the set of models with the highest predictive accuracy from $\mathscr{M}^0$ at the $\alpha\%$ confidence level. There are two aspects to the MCS procedure: a test of equal predictive performance, $\delta_{\mathscr{M}}$, and an elimination rule, $e_{\mathscr{M}}$, that is used if the null hypothesis of equal predictive performance is rejected at each step. For $\delta_{\mathscr{M}}$, I use a multivariate version of the Giacomini-White test from Borup et al. (2022). The test compares the two hypotheses

$$H_0 : \mathbb{E}[\Delta \mathbf{L}_{t+1} | \mathscr{G}_t] = 0$$

$$H_1 : \mathbb{E}[\Delta \mathbf{L}_{t+1} | \mathscr{G}_t] \neq 0$$

Where $\Delta\mathbf{L}_{t+1} = (\Delta L^1_{t+1}, \ldots, \Delta L^k_{t+1})$ is the k-dimensional vector of out-of-sample, one-step-ahead, loss differentials for the k competing models and $\mathscr{G}_t$ is some σ-algebra of information capturing the level of predictability of the dependent variable at each time step. If $\mathscr{G}_t = \{\Omega, \emptyset\}$, the test is one of equal *unconditional* predictive performance, as in Diebold and Mariano (2002). However, such a test does not give information about future performance since it does not consider how difficult the process is to predict in the next time step. As I am interested in how the models will perform for forecasters, I focus on conditional tests of equal performance and use $\mathscr{G}_t = \mathscr{F}_t$. As in Giacomini and White (2006), I use $h^i_t = (1, \Delta L^i_t)'$ to represent conditioning on $\mathscr{F}_t$. Thus, to estimate the conditional expectation function $\mathbb{E}[\Delta\mathbf{L}_{t+1}|h_t]$, after calculating the losses for each model, I run the regressions

$$\Delta L^i_{t+1} = \varphi_0 + \varphi_1 \Delta L^i_t + \eta_{t+1} \qquad i = 1, \ldots, k$$

and perform the test using predicted losses. The resulting MCS compositions are shown in Table 2. As a robustness check on the choice of h^i_t, I repeat the MCS construction using the regressions

$$\Delta L^i_{t+1} = \beta_0 + \beta_1 y_t + \beta_1 \Delta L^i_t + \beta_2 \Delta L^i_{t-1} + \eta_{t+1} \qquad i = 1, \ldots, k$$

where y_t are the log returns at time t, and the results are show in Table C.2. For the elimination rule $e_{\mathscr{M}}$, I rank models according to their predicted losses and, if the null hypothesis of equal conditional predictive performance is rejected, eliminate the model with the highest average predicted loss.

I shall now outline the ARMA-GARCH and NN architectures.

4.4 ARMA-GARCH

Bollerslev (1986) constructed GARCH models whilst developing the ARCH model introduced by Engle (1982). The general ARMA(m, n)-GARCH(p, q) model is given by the following

system of equations, where $\mathbb{E}[\eta_t] = 0$ and $\mathbb{E}[\eta_t^2] = 1$,

$$y_t = \mu_t + \varepsilon_t = \mu_t + \sqrt{h_t}\eta_t \qquad \text{(Dependent Variable)}$$

$$\mu_t = \rho_0 + \sum_{j=1}^{m} \rho_j y_{t-j} + \sum_{z=1}^{n} \rho_z \varepsilon_{t-z} \qquad \text{(Conditional Mean)}$$

$$h_t = \omega + \sum_{k=1}^{p} \alpha_k \varepsilon_{t-k}^2 + \sum_{i=1}^{q} \beta_i h_{t-i} \qquad \text{(Conditional Variance)}$$

These equations model the conditional (with respect to $\mathscr{F}_n$) mean, μ_t, as an ARMA(m, n) process and model the conditional variance of the innovations, ε_t^2, as an ARMA(p, q) process. I estimate the parameters of the model using maximum likelihood estimation in `R` with the `rugarch` package. I use a sliding window technique to create forecasts, but it is important to emphasise that the model is only fitted over the data prior to 2015/01/01, and is not refitted to each window. I do this only due to computational restrictions, as the tests for equal predictive accuracy accommodate both fixed and rolling estimation windows.

There are two nuisances when fitting the ARMA-GARCH models to the Laplace and noncentralised Student-T distributions. First, these distributions are parameterised in a way other than the mean and variance. The Laplace distribution is not directly implemented in `rugarch`, so instead the generalised error distribution is used with the shape parameter fixed at 1. The equivalence of the generalised error distribution and the Laplace distribution is shown in Appendix B.1. The noncentralised Student-T distribution is a three-parameter distribution. Since the ARMA-GARCH model only forecasts two parameters, viz., the mean and variance, one of these three parameters is fixed during out-of-sample forecasting; otherwise, it is impossible to recover our density forecasts. Following Bollerslev (1987) I fix the degrees of freedom, thus the kurtosis does not vary whilst forecasting which is a disadvantage relative to the NN model.

As the introduction mentions, I estimate ARMA-GARCH models of order $(p, p) - (p, p)$ for $p = 1, 2, 3, 4, 5$. Although ideally I would estimate all ARMA-GARCH models up to order $(5, 5) - (5, 5)$, it is computationally intensive to calculate L^2 norms of the density functions for the quadratic and spherical scores. Regardless, this should not significantly affect results as Hansen and Lunde (2005) find that the GARCH(1,1) usually performs very well relative to other GARCH models. However, models incorporating leverage effects may outperform the

GARCH(1,1) model, but these are outside the scope of this paper.

4.5 Neural Network Model

The building block of my architecture is a series of LSTM layers interspersed with dropout layers for regularisation to limit overfitting. Hochreiter and Schmidhuber (1997) developed LSTM blocks to combat the vanishing gradient problem of RNNs by introducing a cell state variable ($\mathbf{c}_t$) in addition to the process state variable ($\mathbf{h}_t$). An LSTM block is comprised of the following system of equations

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{V}_f \mathbf{h}_{t-1} + \mathbf{b}_f)$$

$$\boldsymbol{\iota}_t = \sigma(\mathbf{W}_\iota \mathbf{x}_t + \mathbf{V}_\iota \mathbf{h}_{t-1} + \mathbf{b}_\iota)$$

$$\mathbf{g}_t = \tanh(\mathbf{W}_g \mathbf{x}_t + \mathbf{V}_g \mathbf{h}_{t-1} + \mathbf{b}_g)$$

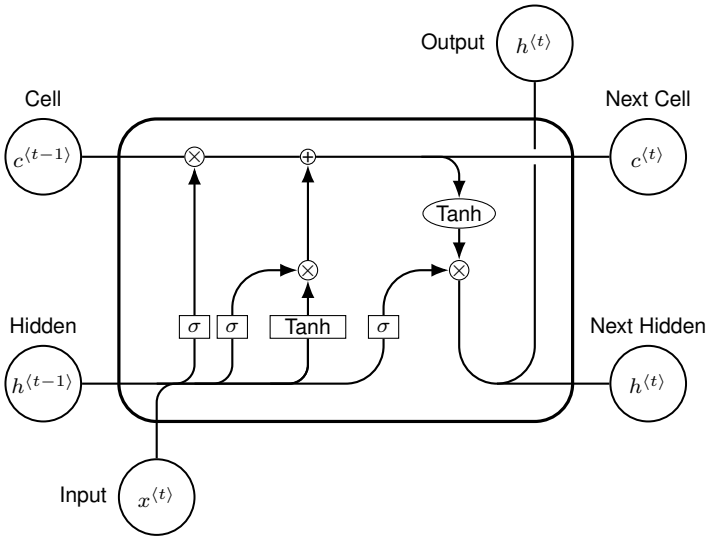
$$\boldsymbol{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{V}_o \mathbf{h}_{t-1} + \mathbf{b}_o)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \boldsymbol{\iota}_t \odot \mathbf{g}_t$$

$$\mathbf{h}_t = \boldsymbol{o} \odot \tanh(\mathbf{c}_t)$$

Where $\odot$ is the element-wise product, and σ is the sigmoid activation. Each block can be visualised as shown in Figure 3.

Figure 3: LSTM Cell Architecture

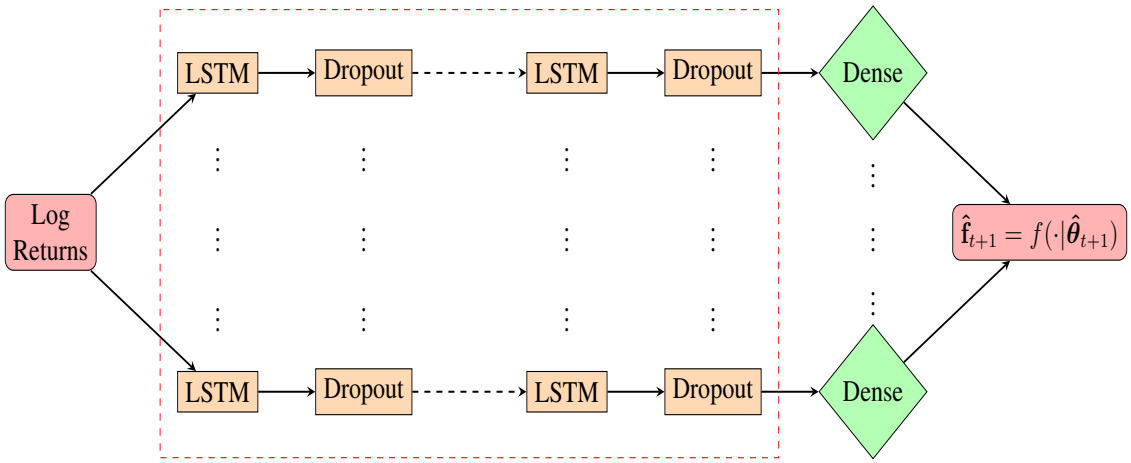


Dropout layers stochastically remove input units during training at a predefined rate. These lay-

ers are not used during forecasting but help prevent overfitting since any node could be dropped out during training meaning over-reliance on particular nodes is penalised. I combine LSTM blocks in sequence with dropout layers before a fully connected layer outputs the parameters for the forecasted distribution in the next time step. Initial attempts to predict all parameters with one LSTM-dropout-dense channel failed to produce forecasts matching the observed mean and variance. Therefore, I use different channels for each distribution parameter which performs more favourably. Features relevant to one parameter appear to be significantly different from those relevant to another parameter.

My NN architecture is outlined in Figure 4.

Figure 4: NN Architecture



The red dotted box denotes the feature extraction layers. Dots indicate that multiple LSTM-dropout-dense channels are used. The LSTM layers use a ReLU activation due to favourable universal approximation results following Cybenko (1989). The Dense layers use ReLU or sigmoid activations depending on the parameters to be estimated. I use 3 LSTM and dropout blocks in each channel, with LSTM widths of 16-32-16 and dropout rates of 0.3-0.4-0.3

After initialising a given NN architecture and with an assumed probability density function, f, one obtains a previsible forecasting strategy $(V_n)_{n\geq 1}$ (i.e. V_n is $\mathscr{F}_{n-1}$-measurable) which constitutes a fitted NN. Letting $(X_i)_{i=1}^n$ be training data, the empirical risk for the model can be calculated through a sliding window technique as

$$L((V_n)_{n\geq 1}, (X_i)_{i=1}^n) = -\frac{1}{n}\sum_{i=1}^{n} S(f(V_n), X_n)$$

I use a window size of 10 days, with a batch size of 64, as this appears to balance learning efficiency with accuracy in finding minima. Stochastic gradient descent is performed using BPTT

with the Adam optimizer (Kingma and Ba, 2014) and a custom learning rate scheduler to fit the NN. BPTT is an iterative procedure for calculating NN layer sensitives after a forward pass through the model when the loss function is evaluated using the forecast values and observed outcomes. BPTT provides an efficient method to calculate the gradient of weight and bias terms. I use the Adam optimiser due to its widespread use in machine learning and since it performs well on the task. The custom learning rate schedule reduces the learning rate by a factor of 0.1 if no improvement in the validation loss is observed for 50 consecutive epochs, thus allowing for efficient initial learning with refinement occurring once the general area of a minimum is found. All NN models are trained for approximately 200 epochs depending on early stopping. Estimation is done in Python using Keras with the TensorFlow backend.

5 Results

5.1 Discussion of Results

The composition of the MCS at the 1,5, and 10 per cent levels is shown in Table 2. The density forecasts for the NN and ARMA-GARCH models are illustrated in Figure 5, and Figures 6-10, respectively. Robustness checks under the alternative test function $h_t = (1, y_t, \Delta L_t, \Delta L_{t-1})'$ are shown in Appendix C.2.

Table 2: MCS Compositions at $\alpha = 0.01, 0.05$, and 0.1 for $h_t = (1, \Delta L_t)'$

	MCS Size	NN	ARMA-GARCH	NN Models	ARMA-GARCH Models
$\alpha = 0.01$					
Nikkei					
Log Score	1	NO	YES	-	(3,3)(3,3)(N)
Quadratic Score	1	NO	YES	-	(2,2)(2,2)(N)
Spherical Score	1	NO	YES	-	(1,1)(1,1)(N)
NASDAQ					
Log Score	1	NO	YES	-	(4,4)(4,4)(T)
Quadratic Score	1	NO	YES	-	(2,2)(2,2)(T)
Spherical Score	2	YES	YES	L	(2,2)(2,2)(T);(4,4)(4,4)(T)
DAX					
Log Score	1	NO	YES	-	(1,1)(1,1)(T)
Quadratic Score	1	NO	YES	-	(1,1)(1,1)(T)
Spherical Score	2	YES	YES	L	(1,1)(1,1)(T)
$\alpha = 0.05$					
Nikkei					
Log Score	1	NO	YES	-	(3,3)(3,3)(N)
Quadratic Score	1	NO	YES	-	(2,2)(2,2)(N)
Spherical Score	1	NO	YES	-	(1,1)(1,1)(N)
NASDAQ					
Log Score	1	NO	YES	-	(4,4)(4,4)(T)
Quadratic Score	1	NO	YES	-	(2,2)(2,2)(T)
Spherical Score	1	NO	YES	-	(4,4)(4,4)(T)
DAX					
Log Score	1	NO	YES	-	(1,1)(1,1)(T)
Quadratic Score	1	NO	YES	-	(1,1)(1,1)(T)
Spherical Score	2	YES	YES	L	(1,1)(1,1)(T)
$\alpha = 0.1$					
Nikkei					
Log Score	1	NO	YES	-	(3,3)(3,3)(N)
Quadratic Score	1	NO	YES	-	(2,2)(2,2)(N)
Spherical Score	1	NO	YES	-	(1,1)(1,1)(N)
NASDAQ					
Log Score	1	NO	YES	-	(4,4)(4,4)(T)
Quadratic Score	1	NO	YES	-	(2,2)(2,2)(T)
Spherical Score	1	NO	YES	-	(4,4)(4,4)(T)
DAX					
Log Score	1	NO	YES	-	(1,1)(1,1)(T)
Quadratic Score	1	NO	YES	-	(1,1)(1,1)(T)
Spherical Score	1	YES	NO	L	-

A YES or NO in the NN and ARMA-GARCH columns indicates that at least one model in that class is in the MCS. A T,L, or N in the NN Models column represents which NN model is in the final MCS (T for assumed T-distribution, L for Laplace, and N for normal). A (p,q)(m,s)(X) in the ARMA-GARCH Models column represents an ARMA(p,q)-GARCH(m,s) model with X=T,L,N assumed distributed innovations is in the MCS.

When using the log score, the MCS only consists of ARMA-GARCH models for all three series down to the 1% level. Therefore, the null hypothesis of equal conditional predictive performance (H_0) is rejected in favour of the alternative hypothesis (H_2) that the ARMA-GARCH

class has higher conditional predictive accuracy than the ARMA-GARCH class. These results are robust to different scoring functions. The only exception is when using the Spherical score on the DAX, where the NN with assumed Laplace distribution is the only model in the final MCS down to the 1% level. Changes in the testing function, h_t, also do not affect results significantly, as the ARMA-GARCH still dominates the NN models. The MCS using $h_t = (1, y_t, \Delta L_t, \Delta L_{t-1})'$ is smaller than when using $h_t = (1, \Delta L_t)'$, potentially indicating that the larger test function is better able to predict variations in the loss differentials and thus has higher power as it can more accurately identify which models are of higher conditional predictive accuracy. Such an interpretation is lent credibility by the simulation results of Giacomini and White (2006), where superfluous variables in the test function were found to have minimal adverse effects on the test's power whilst the inclusion of relevant variables dramatically increased the power.

The graphs in Figure 5 show that the NN performs worse when a normal distribution is assumed and performs best when the noncentralised T-distribution is assumed. This is expected since the normal distribution has no excess kurtosis, whilst the noncentralised T-distribution can flexibly estimate the kurtosis at each time step. However, the conditional mean of the forecasts is approximately constant for all assumed distributions and all series. The NN models cannot disentangle the effects on the loss function from the different distribution parameters, likely because the training dataset is too small.

On the other hand, the ARMA-GARCH models perform favourably across most specifications and series. Interestingly, the models of lower-order (i.e. ARMA$(1,1)$-GARCH$(1,1)$ or ARMA$(2,2)$-GARCH$(2,2)$) generally perform better than the higher-order specifications, and this is reflected in the MCS compositions for both test function specifications. These results align well with the literature as previous empirical studies found that lower-order ARMA-GARCH models often outperformed higher-order variants in point forecasting (Hansen and Lunde, 2005).

These results run contrary to some intuitive reasons why the NN model may outperform the ARMA-GARCH models:

1. The NN model can account for nonlinearities, whilst the ARMA-GARCH model cannot. Nonlinearities may take the form of
    - Regime Changes
    - Threshold Effects
2. The NN has a dynamic process of forgetting information, whilst for the ARMA-GARCH it is static.
3. Leverage effects may be large. The ARMA-GARCH model cannot differentiate between positive and negative shocks whilst modelling the volatility.
4. Large training data may benefit the larger NN model more than the smaller ARMA-GARCH models.

Points 1,2, and 3 are similar and could be placed under the point that NN models can model nonlinear relationships whilst ARMA-GARCH models cannot.

Despite these reasons, other literature has similarly found that linear models often outperform nonlinear models. Possible reasons include the dimensionality effect or because nonlinearities are not present in the validation set (Clements et al., 2004; Dahl and Hylleberg, 2004; Boero and Marrocu, 2004; Diebold and Nason, 1990; De Gooijer and Kumar, 1992). It appears that either the size of the training data, ranging from 6,181 for the DAX to 12,307 for the Nikkei 225, is not large enough or does not capture the underlying source of the nonlinearities so that the 'curse of dimensionality' effect in the NN models outweighs the benefits of nonlinear models relative to linear models.

5.2 Limitations and Potential Extensions

The ARMA-GARCH and NN models were univariate and included no control variables. The lack of control variables, such as macroeconomic indicators, is an issue for both the NN and ARMA-GARCH models. Therefore, given the flexibility of the NN model, I expect the lack of control variables to bias my results in favour of the ARMA-GARCH model. However, the

bias direction is not clear, given data size constraints. I used univariate models to keep them computationally tractable. Although it is not certain, I expect using univariate models to bias results in favour of the ARMA-GARCH models as the interplay between the log returns of the series may be complex and nonlinear, meaning the NN would likely perform relatively better in a multivariate setting. Future work could adapt the methodology taken here to a multivariate version of the ARMA-GARCH models (e.g. Bollerslev (1990) or Ling and McAleer (2003), inter alia.) and a multivariate extension of the NN architecture. The NN architecture could continue assuming some convex hull of probability density functions $\mathscr{C}$, and would then construct the assumed multivariate distribution as a copula with marginal distributions described by some permutation of density functions from $\mathscr{C}$ and the dependency structure coming from any of the distributions described by a density function in $\mathscr{C}$. This method would retain the expressiveness of the NN in a multivariate setting, since Sklar's theorem guarantees any continuous multivariate distribution can be uniquely represented as a copula.

Additionally, due to the troubles with computational intensity, the generalisation error of the results presented may be quite large. Estimation of the ARMA-GARCH models was constrained to models of order $(p, p) - (p, p)$ for $p = 1, 2, 3, 4, 5$, potentially biasing the results toward the NN models if there is an ARMA-GARCH model of a different order which significantly outperforms. Although Hansen and Lunde (2005) found that higher orders than GARCH(1,1) may not significantly improve point forecasts, it is still a shortcoming that the ARMA-GARCH model set was not larger.

Finally, the length of training data may not be sufficiently large to see the full benefits of a NN. At most, it is 12,307 trading days for the Nikkei 225. The NN model may perform much better with much longer series or high-frequency data. Therefore, future work could explore the effect of using larger datasets and whether this can overcome the 'curse of dimensionality'.

6 Conclusion

This paper developed a novel neural network architecture for density forecasting and compared it to a range of ARMA-GARCH models on three globally significant stock indices. The ARMA-GARCH models outperformed the NN models down to the 1% level across all three series, and this result is robust to changes in forecast scoring rules and test function specification.

These results extend earlier findings that linear models outperform nonlinear models in point and interval forecasts. Although one would expect nonlinear models to outperform linear models due to nonlinearities in the data, the size of data required to overcome the effects of using the larger models required for nonlinear techniques is large. Potential next steps in the research include using multivariate models, higher frequency data, or significantly longer periods of training data. These represent an exciting area for future research comparing nonlinear and linear models in density forecasting.

Appendices

A Modelling Assumptions

1. The distribution of log returns is absolutely continuous i.e. there exists a probability density function by the Radon-Nikodym theorem.
2. For all time steps the random variable representing log returns is $\mathscr{L}^2$
3. Structural breaks are not idiosyncratic over the validation set
4. There exists some parametrisation for the process distribution

Points (1) and (2) are weak assumptions needed to estimate the ARMA-GARCH and NN models. Points (3) and (4) are more constricting assumptions necessary for inference. Point (3) ensures that structural breaks are learnable. This is necessary as if the validation set contains particularly idiosyncratic structural breaks our results may be biased towards the ARMA-GARCH models as the benefit of the NN's non-linearity may not be appropriately presented. Point (4) prohibits structural breaks changing the underlying parametrisation or dependency structure of the process and is needed since our models do not allow for switching between different parametrisations e.g. from a Student-T to a Normal distribution. Although it would be possible to incorporate some switching behaviour to the models, to keep the models computationally tractable I abstract away from this.

B Mathematical Derivations

B.1 Equivalence of GED(μ, α, β) and Laplace(a, b)

Suppose X, Y are random variables on a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ such that $X \sim \text{GED}(\mu, \alpha, \beta)$ and $Y \sim \text{Laplace}(a, b)$. These are absolutely continuous random variables with probability density functions

$$f_X(z; \mu, \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)}\text{exp}\left(-\left[\frac{|z-\mu|}{\alpha}\right]^{\beta}\right)$$
$$f_Y(z; a, b) = \frac{1}{2b}\text{exp}\left(-\frac{|z-a|}{b}\right)$$

Thus,

$$f_X(z; a, b, 1) = \frac{1}{2b\Gamma(1)}\text{exp}\left(-\left[\frac{|z-a|}{b}\right]^{1}\right) = f_Y(z; a, b)$$

So, up to values of α, β and a, b the GED with shape parameter 1 has the same probability density function as the Laplace distribution.

Fixing $b \in \mathbb{R}$ arbitrarily, if $\tilde{X} \sim \text{GED}(a, b, 1)$

$$\begin{aligned}(\mathbb{P} \circ Y^{-1})((-\infty, b])) &= F_Y(b) \\ &= \int_{-\infty}^{b} f_Y(z)dz \\ &= \int_{-\infty}^{b} f_{\tilde{X}}(z)dz \\ &= F_{\tilde{X}}(b) \\ &= (\mathbb{P} \circ \tilde{X}^{-1})((-\infty, b]))\end{aligned}$$

So the laws of $\tilde{X}$ and Y agree on the π-system of half lines. Therefore, by Dynkin's uniqueness of extension lemma the laws of $\tilde{X}$ and Y agree on all Borel measurable sets.

B.2 Strict Properness of Scores

Across all scoring rules, I work on the filtered space $(\Omega, \mathscr{F}, (\mathscr{F}_n)_{n\geq 0}, \mathbb{P})$ with the convex hull of probability distributions $\mathscr{C}$, as described in Section 4.1. Let y_{t+1} be a random variable on the filtered space representing the log returns at time $t+1$ and suppose $p_t \in \mathscr{C}$ is the true density function for the distribution of y_{t+1}. Additionally, a forecast, $f_t \in \mathscr{C}$ is generated at time t using either an NN or ARMA-GARCH model. I drop time subscripts in favour of readability which I hope you can forgive.

B.2.1 Log Score

Definition 3. (Log Score) The log score is a scoring function $S_{\log} : \mathscr{C} \times \mathbb{R} \to \mathbb{R}$ given by (for $g \in \mathscr{C}$)

$$S_{\log}(g, y) = \log(g(y))$$

where $\log$ is the natural logarithm.

Theorem 1. *The log score rule is a strictly proper scoring rule.*

Take as given that $\log x \leq x - 1$. Therefore, $-\log x \geq 1 - x$. So,

$$\begin{aligned}
\mathbb{E}_p[S_{\log}(p, y) - S_{\log}(f, y)] &= \mathbb{E}_p\left[\log\left(\frac{p(y)}{f(y)}\right)\right] \\
&= \mathbb{E}_p\left[-\log\left(\frac{f(y)}{p(y)}\right)\right] \\
&\geq \mathbb{E}_p\left[1 - \frac{f(y)}{p(y)}\right] \\
&= 1 - \mathbb{E}_p\left[\frac{f(y)}{p(y)}\right] \\
&= 1 - \int_{\mathbb{R}} \frac{f(x)}{p(x)} dF_p(x) \\
&= 1 - \int_{\mathbb{R}} p(x)\frac{f(x)}{p(x)} dx \\
&= 1 - \underbrace{\int_{\mathbb{R}} f(x)dx}_{1 \text{ since } f \text{ is a pdf}} \\
&= 1 - 1 \\
&= 0
\end{aligned}$$

So,

$$\mathbb{E}_p\left[\log\left(\frac{p(y)}{f(y)}\right)\right] \geq 0$$

Then,

$$\begin{aligned}
\mathbb{E}_p\left[\log\left(\frac{p(y)}{f(y)}\right)\right] &\geq 0 \\
\mathbb{E}_p\left[\log(p(y)) - \log(f(y))\right] &\geq 0 \\
\mathbb{E}_p[\log(p(y))] - \mathbb{E}_p[\log(f(y))] &\geq 0 \\
\mathbb{E}_p[\log(p(y))] &\geq \mathbb{E}_p[\log(f(y))] \\
\mathbb{E}_p[S_{\log}(p,y)] &\geq \mathbb{E}_p[S_{\log}(f,y)]
\end{aligned}$$

Since $f \in \mathscr{C}$ was arbitrary, this shows that the log score rule is proper.

Next, we show that the scoring rule is strictly proper. If $p = f$ a.e. then

$$\begin{aligned}
\mathbb{E}_p[\log(p(y))] &= \mathbb{E}_p[\log(f(y))] \\
\mathbb{E}_p[S_{\log}(p,y)] &= \mathbb{E}_p[S_{\log}(f,y)]
\end{aligned}$$

Conversely, assuming $\mathbb{E}_p[S_{\log}(p,y)] = \mathbb{E}_p[S_{\log}(f,y)]$, define pointwise $R := \frac{f}{p}$ Now,

$$\begin{aligned}
\mathbb{E}_p[R] &= \int_{\mathbb{R}} p(x)\frac{f(x)}{p(x)}dx \\
&= \int_{\mathbb{R}} f(x)dx \\
&= 1
\end{aligned}$$

$x \mapsto \log x$ is concave and $R \in \mathscr{L}^1(\Omega, \mathscr{F}, \mathbb{P})$. Therefore, by Jensen's inequality,

$$\begin{aligned}
\mathbb{E}_p[\log R] &\leq \log(\mathbb{E}_p[R]) \\
&= \log(1) \\
&= 0
\end{aligned}$$

This is equivalent to

$$\mathbb{E}_p[\log(p(y)] \geq \mathbb{E}_p[\log(f(y))]$$

Jensen's inequality holds strictly iff $R = \mathbb{E}_p[R]$ a.e., i.e. if for almost every $y \in \mathbb{R}$

$$\begin{aligned}
R &= \mathbb{E}_p[R] \\
\frac{f(y)}{p(y)} &= \mathbb{E}_p[R] \\
\frac{f(y)}{p(y)} &= 1 \\
f(y) &= p(y)
\end{aligned}$$

Thus, the log score rule is strictly proper.

B.2.2 Spherical Score

Definition 4. (Spherical Score) The spherical score is a scoring function $SphS : \mathscr{C} \times \mathbb{R} \to \mathbb{R}$ given by (for $g \in \mathscr{C}$)

$$SphS(g, y) = \frac{g(y)}{||g||_2}$$

Theorem 2. *The spherical score rule is a strictly proper scoring rule.*

Consider,

$$\begin{aligned}
\mathbb{E}_p[SphS(p, y) - SphS(f, y)] &= \int_{\mathbb{R}} (SphS(p, y) - SphS(f, y)) dF_p(y) \\
&= \int_{\mathbb{R}} (SphS(p, y) - SphS(f, y)) p(y) dy \\
&= \int_{\mathbb{R}} \left(\frac{p(y)}{||p||_2} - \frac{f(y)}{||f||_2} \right) p(y) dy \\
&= \frac{1}{||p||_2} \int_{\mathbb{R}} p^2(y) dy - \frac{1}{||f||_2} \int_{\mathbb{R}} f(y) p(y) dy \\
&= ||p||_2 - \frac{(f, p)}{||f||_2} \\
&= \frac{||p||_2 ||f||_2 - (f, p)}{||f||_2} \\
&\geq 0
\end{aligned}$$

Where the final inequality is Cauchy-Schwarz. So, $\mathbb{E}_p[SphS(p, y)] \geq \mathbb{E}_p[SphS(f, y)]$ with equality if and only if f is a scalar multiple of p by the Cauchy-Schwarz inequality. As all probability density functions have L_1 norm equal to 1, $\mathbb{E}_p[SphS(p, y)] = \mathbb{E}_p[SphS(f, y)]$ if and only if $f = p$ a.e..

B.2.3 Quadratic Score

Definition 5. (Quadratic Score) The quadratic score is a scoring function $QS : \mathscr{C} \times \mathbb{R} \to \mathbb{R}$ given by (for $g \in \mathscr{C}$)

$$QS(g, y) = 2g(y) - ||g||_2^2$$

Theorem 3. *The quadratic score rule is a strictly proper scoring rule.*

Consider,

$$\begin{aligned}
\mathbb{E}_p[QS(p, y) - QS(f, y)] &= \int_{\mathbb{R}} [QS(p, y) - QS(f, y)] dF_p(y) \\
&= \int_{\mathbb{R}} [QS(p, y) - QS(f, y)] p(y) dy \\
&= \int_{\mathbb{R}} [2p(y) - ||p||_2^2 - 2f(y) + ||f||_2^2] p(y) dy \\
&= ||p||_2^2 - 2(f, p) + ||f||_2^2 \\
&= ||p - f||_2^2 \\
&\geq 0
\end{aligned}$$

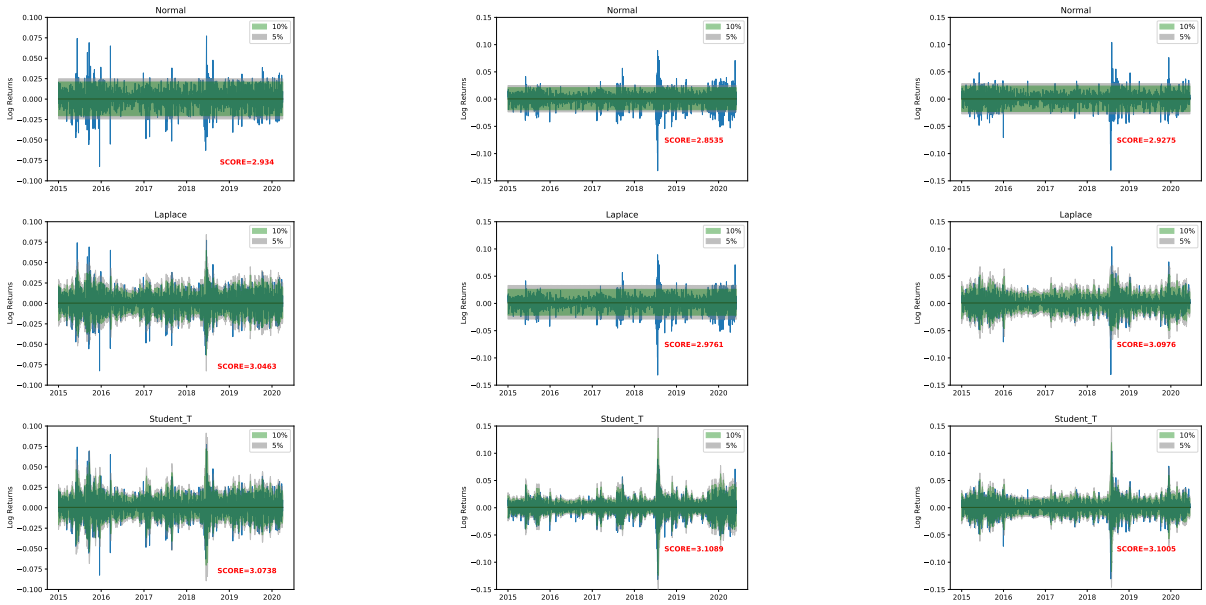
So, $\mathbb{E}_p[QS(p, y)] \geq \mathbb{E}_p[QS(f, y)]$ with equality if and only if $f = p$ a.e..

C Tables and Figures

C.1 Graphs of Forecasted Densities

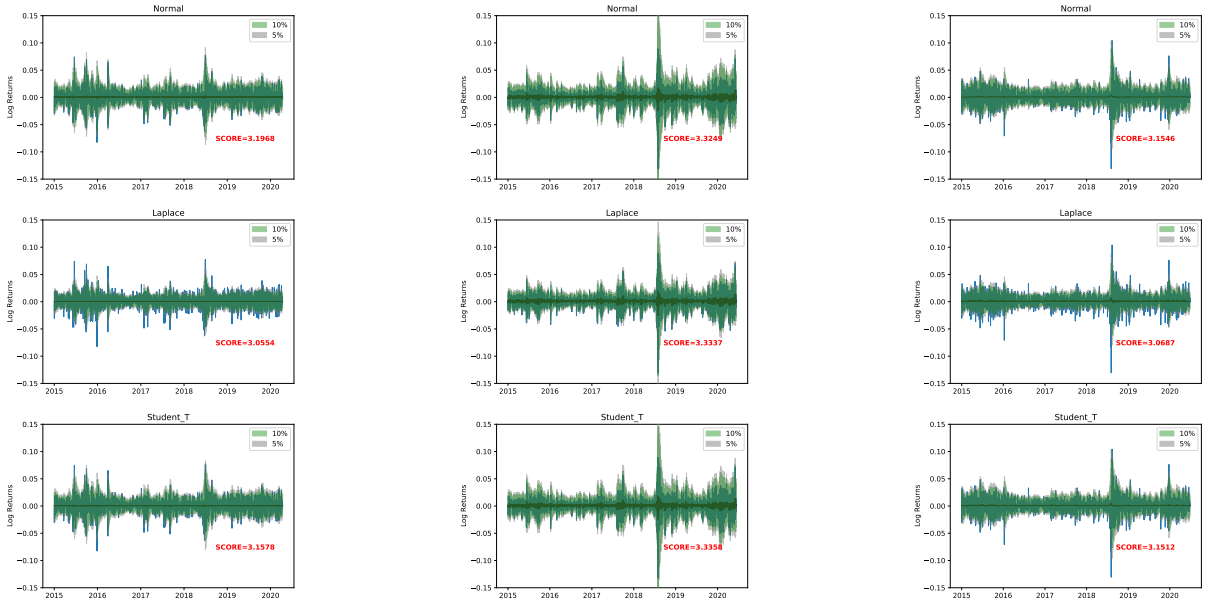
10% and 5% VaR bands are shown. Score represents the average logarithmic score over the validation set.

Figure 5: Neural Network Forecasts Over Validation Set



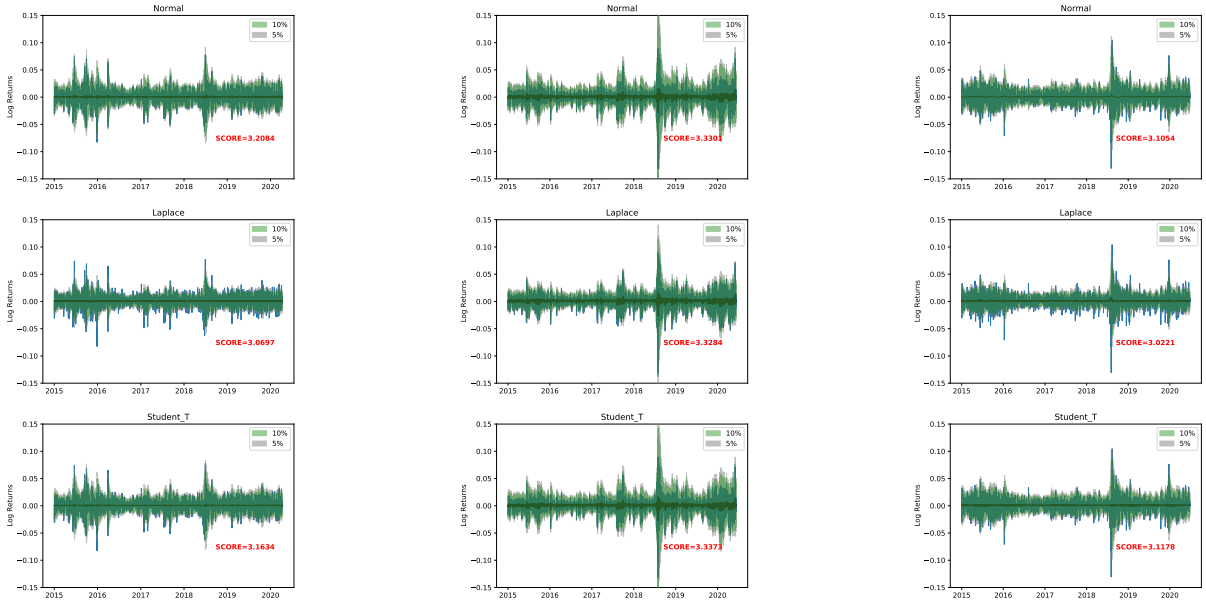
(a) Nikkei 225 **(b)** NASDAQ **(c)** DAX

Figure 6: ARMA(1,1)-GARCH(1,1) Forecasts Over Validation Set



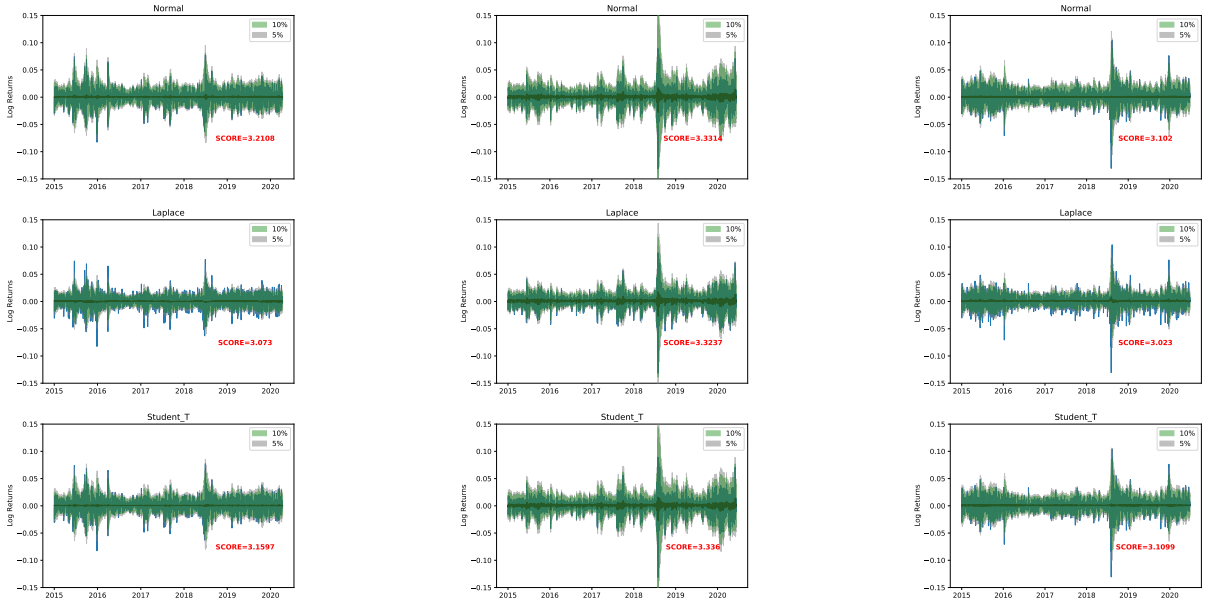
(a) Nikkei 225 **(b)** NASDAQ **(c)** DAX

Figure 7: ARMA(2,2)-GARCH(2,2) Forecasts Over Validation Set



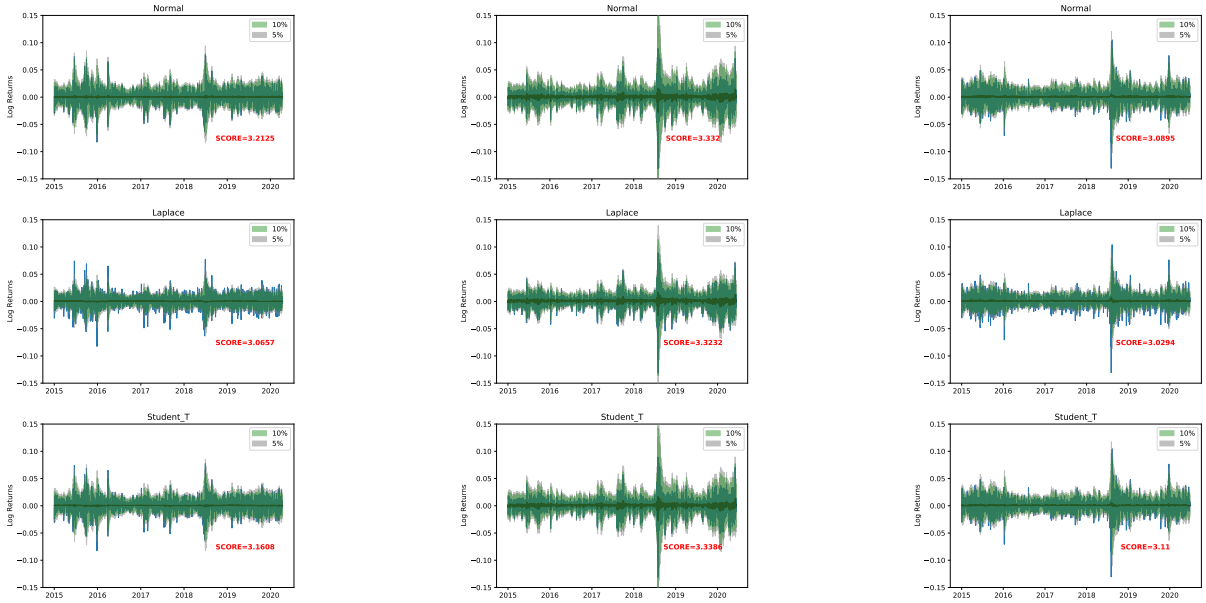
(a) Nikkei 225 **(b)** NASDAQ **(c)** DAX

Figure 8: ARMA(3,3)-GARCH(3,3) Forecasts Over Validation Set



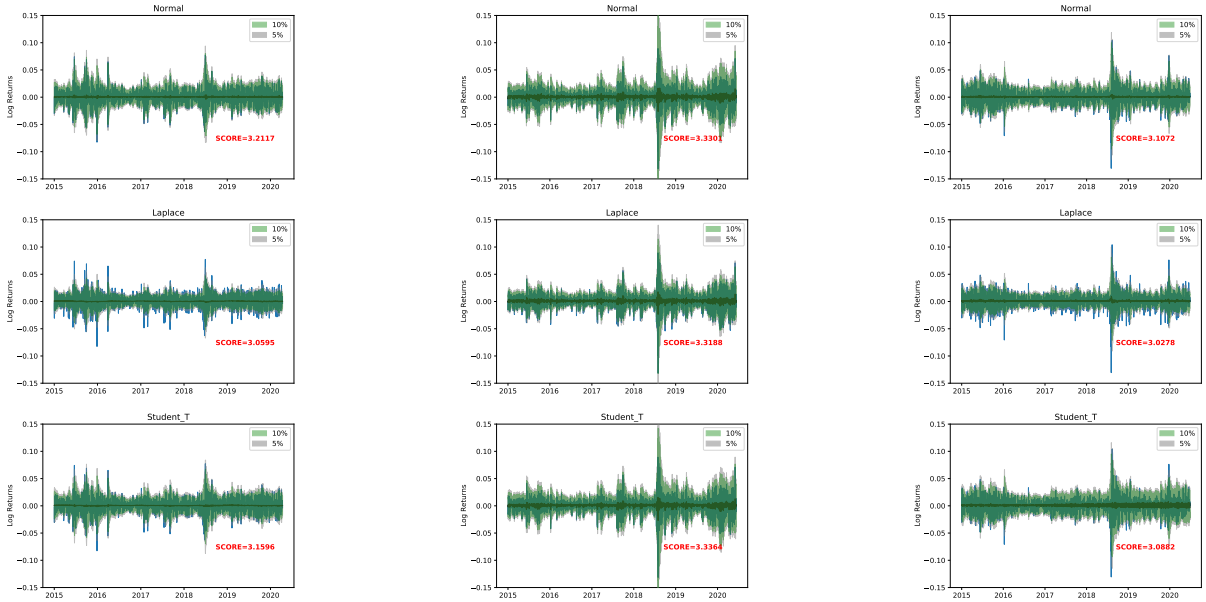
(a) Nikkei 225 **(b)** NASDAQ **(c)** DAX

Figure 9: ARMA(4,4)-GARCH(4,4) Forecasts Over Validation Set



(a) Nikkei 225 **(b)** NASDAQ **(c)** DAX

Figure 10: ARMA(5,5)-GARCH(5,5) Forecasts Over Validation Set



(a) Nikkei 225 **(b)** NASDAQ **(c)** DAX

C.2 Test Function Robustness Checks

Table 3: MCS Compositions at $\alpha = 0.01, 0.05$, and 0.1 for $h_t = (1, y_t, \Delta L_t, \Delta L_{t-1})'$

	MCS Size	NN	ARMA-GARCH	NN Models	ARMA-GARCH Models
$\alpha = 0.01$					
Nikkei					
Log Score	1	NO	YES	-	(1,1)(1,1)(L)
Quadratic Score	1	NO	YES	-	(1,1)(1,1)(L)
Spherical Score	1	NO	YES	-	(1,1)(1,1)(L)
NASDAQ					
Log Score	1	YES	NO	N	-
Quadratic Score	1	NO	YES	-	(2,2)(2,2)(N)
Spherical Score	1	NO	YES	-	(2,2)(2,2)(N)
DAX					
Log Score	1	NO	YES	-	(1,1)(1,1)(T)
Quadratic Score	1	NO	YES	-	(1,1)(1,1)(N)
Spherical Score	1	NO	YES	-	(1,1)(1,1)(T)
$\alpha = 0.05$					
Nikkei					
Log Score	1	NO	YES	-	(1,1)(1,1)(L)
Quadratic Score	1	NO	YES	-	(1,1)(1,1)(L)
Spherical Score	1	NO	YES	-	(1,1)(1,1)(L)
NASDAQ					
Log Score	1	YES	NO	N	-
Quadratic Score	1	NO	YES	-	(2,2)(2,2)(N)
Spherical Score	1	NO	YES	-	(2,2)(2,2)(N)
DAX					
Log Score	1	NO	YES	-	(1,1)(1,1)(T)
Quadratic Score	1	NO	YES	-	(1,1)(1,1)(N)
Spherical Score	1	NO	YES	-	(1,1)(1,1)(T)
$\alpha = 0.1$					
Nikkei					
Log Score	1	NO	YES	-	(1,1)(1,1)(L)
Quadratic Score	1	NO	YES	-	(1,1)(1,1)(L)
Spherical Score	1	NO	YES	-	(1,1)(1,1)(L)
NASDAQ					
Log Score	1	YES	NO	N	-
Quadratic Score	1	NO	YES	-	(2,2)(2,2)(N)
Spherical Score	1	NO	YES	-	(2,2)(2,2)(N)
DAX					
Log Score	1	NO	YES	-	(1,1)(1,1)(T)
Quadratic Score	1	NO	YES	-	(1,1)(1,1)(N)
Spherical Score	1	NO	YES	-	(1,1)(1,1)(T)

A YES or NO in the NN and ARMA-GARCH columns indicates at least one model in that class is in the MCS. A T,L, or N in the NN Models column represents which NN model is in the final MCS (T for assumed T-distribution, L for Laplace, and N for normal). A (p,q)(m,s)(X) in the ARMA-GARCH Models column represents an ARMA(p,q)-GARCH(m,s) model with X=T,L,N assumed distributed innovations is in the MCS.

References

Amisano, G. and Giacomini, R. (2007). Comparing density forecasts via weighted likelihood ratio tests. *Journal of Business & Economic Statistics*, 25(2):177–190.

Bao, Y., Lee, T.-H., and Saltoğlu, B. (2007). Comparing density forecast models. *Journal of Forecasting*, 26(3):203–225.

Berkowitz, J. (2001). Testing density forecasts, with applications to risk management. *Journal of Business & Economic Statistics*, 19(4):465–474.

Boero, G. and Marrocu, E. (2004). The performance of setar models: a regime conditional evaluation of point, interval and density forecasts. *International Journal of Forecasting*, 20(2):305–320. Forecasting Economic and Financial Time Series Using Nonlinear Methods.

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

Bollerslev, T. (1987). A conditionally heteroskedastic time series model for speculative prices and rates of return. *The Review of Economics and Statistics*, 69(3):542–547.

Bollerslev, T. (1990). Modelling the coherence in short-run nominal exchange rates: A multivariate generalized arch model. *The Review of Economics and Statistics*, 72(3):498–505.

Borup, D., Eriksen, J. N., Kjaer, M. M., and Thyrsgaard, M. (2022). Predicting bond return predictability. *Management Science*, page Accepted for Publication.

Clements, M. P., Franses, P. H., and Swanson, N. R. (2004). Forecasting economic and financial time-series with non-linear models. *International Journal of Forecasting*, 20(2):169–183. Forecasting Economic and Financial Time Series Using Nonlinear Methods.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Dahl, C. M. and Hylleberg, S. (2004). Flexible regression models and relative forecast performance. *International Journal of Forecasting*, 20(2):201–217. Forecasting Economic and Financial Time Series Using Nonlinear Methods.

De Gooijer, J. G. and Kumar, K. (1992). Some recent developments in non-linear time series modelling, testing, and forecasting. *International Journal of Forecasting*, 8(2):135–156.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. (2018). Being robust (in high dimensions) can be practical.

Diebold, F. X., Gunther, T. A., and Tay, A. S. (1998). Evaluating density forecasts with applications to financial risk management. *International Economic Review*, 39(4):863–883.

Diebold, F. X. and Mariano, R. S. (2002). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 20(1):134–144.

Diebold, F. X. and Nason, J. A. (1990). Nonparametric exchange rate prediction? *Journal of International Economics*, 28(3):315–332.

Diks, C., Panchenko, V., and van Dijk, D. (2011). Likelihood-based scoring rules for comparing density forecasts in tails. *Journal of Econometrics*, 163(2):215–230.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007.

Engle, R. F. (2015). Discussion. *The Review of Financial Studies*, 3(1):103–106.

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

Fama, E. F., Fisher, L., Jensen, M. C., and Roll, R. (1969). The adjustment of stock prices to new information. *International Economic Review*, 10(1):1–21.

Giacomini, R. and White, H. (2006). Tests of conditional predictive ability. *Econometrica*, 74(6):1545–1578.

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

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Goulet, Coulombe, P., Leroux, M., Stevanovic, D., and Surprenant, S. (2022). How is machine learning useful for macroeconomic forecasting? *Journal of Applied Econometrics*, 37(5):920–964.

Granger, C. W. J. and Newbold, P. (1977). *Forecasting economic time series*. Academic Press, London;New York (etc.);.

Guo, Z., Zhou, K., Zhang, X., and Yang, S. (2018). A deep learning model for short-term power load and probability density forecasting. *Energy*, 160:1186–1200.

Hansen, P. R. and Lunde, A. (2005). A forecast comparison of volatility models: does anything beat a garch(1,1)? *Journal of Applied Econometrics*, 20(7):873–889.

Hansen, P. R., Lunde, A., and Nason, J. M. (2011). The model confidence set. *Econometrica*, 79(2):453–497.

Harvey, D., Leybourne, S., and Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, 13(2):281–291.

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

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization.

Lee, J., Sohl-dickstein, J., Pennington, J., Novak, R., Schoenholz, S., and Bahri, Y. (2018). Deep neural networks as gaussian processes. In *International Conference on Learning Representations*.

Leitch, G. and Tanner, J. E. (1991). Economic forecast evaluation: Profits versus the conventional error measures. *American Economic Review*, 81(3):580–90.

Ling, S. and McAleer, M. (2003). Asymptotic theory for a vector arma-garch model. *Econometric Theory*, 19(2):280–310.

Mitchell, J. and Hall, S. (2005). Evaluating, comparing and combining density forecasts using the klic with an application to the bank of england and niesr 'fan' charts of inflation*. *Oxford Bulletin of Economics and Statistics*, 67(s1):995–1033.

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

Park, J.-H., Yoo, S.-J., Kim, K.-J., Gu, Y.-H., Lee, K.-H., and Son, U.-H. (2017). Pm10 density forecast model using long short term memory. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 576–581.

Savage, L. J. (1971). Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801.

Taylor, J. W. and Buizza, R. (2004). A comparison of temperature density forecasts from garch and atmospheric models. *Journal of Forecasting*, 23(5):337–355.

Tong, H., Chan, K. S., Cox, D. R., Cutler, C. D., Guégan, D., Jensen, J. L., Johansen, S., Lawrance, A. J., Lebaron, B., Ozaki, T., Nychka, D. W., Ellner, S., Bailey, B. A., Gallant, A. R., Smith, L. R., Smith, R. L., and Wolff, R. C. L. (1995). A personal overview of non-linear time series analysis from a chaos perspective [with discussion and rejoinder]. *Scandinavian Journal of Statistics*, 22(4):399–445.

Wei, J., Xu, Q., and He, C. (2022). Deep learning of predicting closing price through historical adjustment closing price. *Procedia Computer Science*, 202:379–384. International Conference on Identification, Information and Knowledge in the internet of Things, 2021.

West, K., Edison, H., and Cho, D. (1993). A utility-based comparison of some models of exchange rate volatility. *Journal of International Economics*, 35(1-2):23–45.

Yeo, K., Melnyk, I., Nguyen, N., and Lee, E. K. (2018). De-rnn: Forecasting the probability density function of nonlinear time series. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 697–706.

Zhang, L., Lu, S., Ding, Y., Duan, D., Wang, Y., Wang, P., Yang, L., Fan, H., and Cheng, Y. (2022). Probability prediction of short-term user-level load based on random forest and kernel density estimation. *Energy Reports*, 8:1130–1138. ICPE 2021 - The 2nd International Conference on Power Engineering.