Historical Perspectives in Volatility Forecasting Methods with Machine Learning

Zhiang Qiu[1], Clemens Kownatzki[2], Fabien Scalzo[1], and Eun Sang Cha[1]

[1]*Pepperdine Seaver College*

[2]*Pepperdine Graziadio Business School*

Abstract

Volatility forecasting in the financial market plays a pivotal role across a spectrum of disciplines, such as risk management, option pricing, and market making. However, volatility forecasting is challenging because volatility can only be estimated, and different factors influence volatility, ranging from macroeconomic indicators to investor sentiments. While recent works suggest advances in machine learning and artificial intelligence for volatility forecasting, a comprehensive benchmark of current statistical and learning-based methods for such purposes is lacking. Thus, this paper aims to provide a comprehensive survey of the historical evolution of volatility forecasting with a comparative benchmark of key landmark models. We open-source our benchmark code to further research in learning-based methods for volatility forecasting.

Keywords: volatility forecasting, risk management, deep learning, time series analysis, GARCH, LSTM, Transformer

I. INTRODUCTION

Financial institutions are required by the government and driven by their performance target to manage risk. An integral aspect of this risk stems from the movement of the equity market, primarily the market's volatility. Volatility can be used to determine the risk exposure of a portfolio (R. Engle, 2004), the anticipated fluctuations throughout the duration of an option (Black & Scholes, 1973), and the bid-ask spread of options as well as their underlying asset (Bollerslev & Melvin, 1994). A model that helps financial institutions forecast the volatility of their holdings would provide a clearer picture of their risk and facilitate the process of risk management and decision-making.

Volatility cannot be observed; therefore, it has to be estimated, which is why we embark on this journey to survey different volatility models. The advantages and feasibility of monitoring volatility have spurred researchers to employ a wide range of models, ranging from traditional Generalized Autoregressive Conditional Heteroskedasticity (GARCH) (Bollerslev, 1986) to Neural Network frameworks such as Long Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) and Transformers (Vaswani et al., 2023). Subsequent sections will explore the key characteristics of volatility and the current challenges in the field, starting with the stylized facts.

First, volatility is dynamic and displays temporal clustering (Mandelbrot, 1997). Observing significant returns today, regardless of direction, often presages returns of similar magnitude in the ensuing days (D. Kim & Shin, 2023). Moreover, volatility's past fluctuations can exert lasting influences on its future path, signifying that volatility possesses a long memory (Poon & Granger, 2003). Another observation is that the probability of extreme market events exceeds what the normal distribution would predict, indicating that return distributions exhibit heavy tails (Cont, 2001).

Additionally, a negative correlation exists between prices and volatility: as prices drop, volatility intensifies, and as prices rise, volatility diminishes, though to a lesser extent (McAleer & Medeiros, 2008). This relationship is commonly termed either the leverage effect or the Asymmetric Volatility Phenomenon (AVP) (Aıt-Sahalia, 2017) (R. F. Engle & Ng, 1993). Due to the AVP, options prices exhibit a skew, with strike prices below the current market price typically having higher implied volatility than higher strikes. This skewness can be attributed to several factors. Firstly, there's the behavioral finance principle of loss aversion, where investors tend to prioritize avoiding losses over achieving equivalent gains (Tversky & Kahneman, 1991). Secondly, when a stock's value decreases, its financial leverage rises as the percentage of debt in its capital structure increases, making the stock riskier and boosting its volatility (Christie, 1982). Lastly, adverse events increase conditional covariances substantially, whereas positive shocks have a mixed impact on conditional covariances (Bekaert & Wu, 2000).

Volatility also exhibits mean reversion. Unlike stocks that have a positive drift, implied volatility tends to gradually increase before earnings and major events such as the Federal Open Market Committee (FOMC) meetings. It can also spike when encountering unexpected events. However, in either case, volatility tends to revert to the mean after the event happens (Goudarzi, 2013). These stylized attributes make volatility forecasting possible.

Nevertheless, predicting volatility remains a challenging endeavor. Volatility is influenced by an array of elements, from macroeconomic phenomena, corporate earnings reports, interest rates, global commodity price trends, and psychology (Shiller, 1999). The interactions among these factors can be complex. Volatility is also easily mistaken with uncertainty (Knight, 1921). Uncertainty refers to exogenous shocks, such as geopolitical tensions, natural calamities, or abrupt regulatory shifts, which can yield immediate and pronounced volatility spikes and are inherently challenging to anticipate. While we can estimate volatility through various risk measures, uncertainty still evades our current capabilities of being quantified and measured.

The benefits and challenges of forecasting volatility have garnered attention from many, and ongoing revisions have been undertaken to include the latest progressions in this domain (T. Andersen et al., 2005). There are numerous studies surveying volatility forecasting methods, (Poon & Granger, 2003)(Ge et al., 2023)(Sezer et al., 2020), but we are not aware of a recent comprehensive review that both provides a clear explanation and compares the foundational and cutting-edge volatility models side by side. Within this paper, we survey the evolution of volatility forecasting models and evaluate their performance using a representative dataset from the Standard and Poor's 500 index (S&P 500). The contributions of this paper are:

1. We *survey* the evolution of volatility forecasting models, transitioning from traditional AR and implied volatility models to contemporary variations of the Transformer models, which represent the current state-of-the-art.
2. We select a representative model from each category and conduct a *comparative benchmark* to show their respective performances, paving the way for subsequent model developments.
3. We *open-source our benchmarks* and comprehensively analyze the advantages and disadvantages inherent to each model type.[1]

II. LITERATURE REVIEW

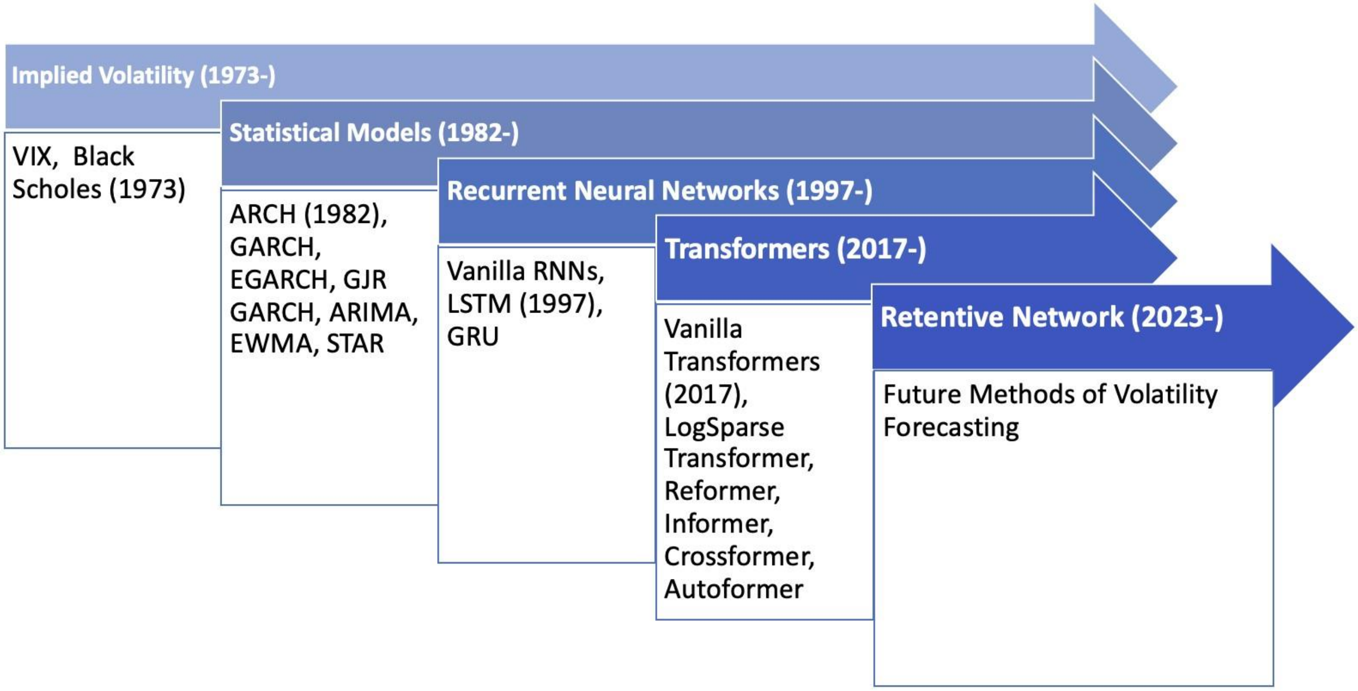

Implied Volatility (1973-)
VIX, Black Scholes (1973)
Statistical Models (1982-)
ARCH (1982), GARCH, EGARCH, GJR GARCH, ARIMA, EWMA, STAR
Recurrent Neural Networks (1997-)
Vanilla RNNs, LSTM (1997), GRU
Transformers (2017-)
Vanilla Transformers (2017), LogSparse Transformer, Reformer, Informer, Crossformer, Autoformer
Retentive Network (2023-)
Future Methods of Volatility Forecasting

Fig. 1: Timeline

A. Overview

In volatility forecasting, numerous models have been employed, and many of them have been applied in conjunction with others. For organizational purposes, we have categorized these models into four primary classifications: Statistical models (Bollerslev, 1986), Implied Volatility, Recurrent Neural Networks (RNNs) (Connor et al., 1994) (Hochreiter & Schmidhuber, 1997) (Chung et al., 2014), and Transformers (Vaswani et al., 2023). In subsequent sections, we will discuss the specifics of the models within each category, highlighting the landmark contributions.

B. AR and MA models

To address the stylized facts, Robert Engle's Autoregressive Conditional Heteroskedasticity (ARCH) model (R. F. Engle, 1982) was initially adopted for volatility forecasting. This was followed by the introduction of the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model (Bollerslev, 1986) by his student Tim Bollerslev, with the common goal of leveraging the occurrence of volatility clustering (T. G. Andersen, 2018). Since its inception, numerous scholars have employed and adapted the GARCH model. For example, the asymmetric GARCH (AGARCH), Exponential GARCH (EGARCH), and Glosten

[1] The data and Python implementation are available at https://github.com/WithAnOrchid0513/VolData.

Jagannathan Runkle GARCH (GJR GARCH) observed that negative shocks have a more substantial impact on the variance than positive shocks. According to the survey paper Bollerslev wrote in 2007 (Bollerslev, 2007), there were already numerous variants of the original ARCH model, and this number is continuously growing. Other than GARCH models, other statistical models include simple moving average (SMA) (Johnston et al., 1999), exponentially weighted moving average (EWMA) (Holt, 2004), Smooth Transition Exponential Smoothing (Taylor, 2004), autoregressive integrated moving average (ARIMA) (Box & Pierce, 1970), and smooth transition autoregressive (STAR) (Bildirici & Ersin, 2015). These models accommodate factors such as seasonality, long-term trends, autoregressive, and moving averages. Among them, SMA creates a smooth curve from past data over a defined period, while EWMA emphasizes recent data more by giving more weight to recent observations. The Smooth Transition Exponential Smoothing model uses a logistic function based on a selected variable for its smoothing effect. ARIMA combines autoregressive techniques and moving averages, and STAR focuses on detecting non-linear trends in data. A unifying trait among AR models is their reliance on historical values and stationarity for predictions. For a time series to be stationary, its statistical properties such as mean, variance and covariance must not be a function of time. Stock prices themselves are clearly not stationary, as they tend to increase with a drift term; however, their log returns are usually stationary. To test whether a time series is stationary, the ADF (Augmented Dickey-Fuller) test is typically implemented. The stationarity requirement made prepossessing the data into log returns necessary, which is a significant limitation for AR models when implemented in real-time.

C. Implied Volatility

In addition to AR models, both the Chicago Board Options Exchange's CBOE Volatility Index (VIX) and implied volatility (IV) serve as predictors for future volatility. Often referred to as the "fear index," the VIX mirrors the market's 30-day anticipated volatility (Whaley, 2009). Unlike its original derivation, which was based on a narrow set of strike prices to determine implied volatility, today, the VIX is based on the methodology of a volatility swap (Derman, 1999). However, rather than being an actual swap, a volatility swap is a forward contract on the realized variance (Diamond, 2012). The computation of VIX uses two months of the latest option data while interpolating between the nearest and second nearest expiration month to create a consistent 30-day window of expected volatility. Specific option strikes are then chosen for the VIX calculation, as elaborated in the VIX white paper (CBOE, 2019). IV, which underpins the VIX, is viewed as a reliable volatility predictor since it draws from option prices, capturing real investor expectations about future events (Poon & Granger, 2003). However, IV has its limitations. It can only be estimated using the option's price by calculating the volatility implied in that price through an iterative procedure.

D. Early ML models

With its ability to learn from and make predictions based on data, machine learning has been applied to more and more different fields, and finance is no exception. Unlike econometric models that aim to be parsimonious by limiting the number of parameters, machine learning embraces the use of a vast number of parameters (Kelly & Xiu, 2023). This approach has led to the adoption of many new techniques for volatility forecasting, either as entirely new methods or as extensions of existing ones (Ge et al., 2023). The following paragraphs will start with traditional machine learning (ML) models and extend to Neural Networks (NN), both used extensively for volatility analysis.

Decision trees (DTs) (Loh, 2011), random forests (RF) (Breiman, 2001), and XGBoost (Chen & Guestrin, 2016) are among the foundational applications for machine learning. Decision trees consist of a supervised learning algorithm that ascertains the value of a target variable by deducing straightforward decision rules from the data's features (Loh, 2011). The Random Forest (RF) algorithm operates as an ensemble of these decision trees, chosen through stochastic processes (Breiman, 2001). XGBoost is an optimized distributed gradient boosting library rooted in decision tree algorithms (C. Zhang et al., 2023). Different from random forest, XGBoost uses a boosting method to combine trees together so that each tree corrects the error of the previous one.

Furthermore, another well-used method of machine learning is Principal Component Analysis (PCA). PCA reduces the dimension of the dataset while ensuring the principal components are still consistent estimators of the true factors (Stock & Watson, 2002). For example, Ludvigson and Ng found a volatility factor and a risk premium factor that contain significant information about future returns (Ludvigson & Ng, 2007). Such analysis is further improved by assigning weights to predictors that reflect their relative forecasting strength (D. Huang et al., 2022).

E. Neural Networks

While these traditional ML methods have reasonable short-term forecasting capabilities and flexibility, they also have several limitations. Their ability to predict long-term and complex volatility is limited, and missing values, which are not uncommon in practice, can cause significant issues for these traditional models. In order to address these problems, Neural Networks (NNs) were introduced (Pranav, 2021). NNs were inspired by the biological neural networks in human brains and can detect complex patterns in nonlinear form (Hornik et al., 1989). Structurally, a neural network (NN) is composed of multiple layers. Each layer features neurons that are interconnected through weighted links, which are then adjusted during the training process. This adjustment is done by using backpropagation to compute the gradient of the loss function for each weight using the chain rule.

NNs were traditionally used for tasks like image and speech recognition (Abdel-Hamid et al., 2014), targeted marketing (Venugopal & Baets, 1994), and autonomous vehicles (Pomerleau, 1988). When applied to time series, the NNs can recognize the relationship between past and current inputs and use them to predict future outputs. While various NNs have been applied (Chow & Leung, 1996) (Marcek, 2018), Recurrent Neural Networks (RNNs) are particularly prominent for time series-related tasks (Connor et al., 1994). Their appeal lies in their time-sensitive activation functions; RNNs retain information specific to a particular timestep, which is sequentially updated, making them ideal for forecasting endeavors. In the RNN process, data is initially fed to produce a preliminary result. This result is then contrasted with the actual outcome via a loss function, triggering a backpropagation to fine-tune the gradient for each neuron in the network. This iterative process optimizes each neuron's weight. However, despite their potential, RNNs have limitations, especially the vanishing gradient problem ("Gradient Flow in Recurrent Nets," 2009). In backpropagation, derivatives are calculated layer by layer from the end to the beginning. As per the chain rule, when these derivatives are successively multiplied, they can diminish exponentially, causing them to vanish. Similarly, the gradient could also explode if the opposite happens. These lead to the RNN's failure to learn long-term dependencies. RNNs are also computationally expensive and cannot be parallelized, which makes training an RNN difficult.

To address these problems, algorithmic methods such as Long Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), and Gated Recurrent Unit (GRU) (Chung et al., 2014), have been introduced (Sezer et al., 2020) (Pranav, 2021). Both LSTM and GRU extend the capabilities of RNNs. These models employ gates to update or remove information to the hidden states to address the long-term dependencies.

The GRU is a simplified version of the LSTM by combining gates and reducing parameters. While less powerful, it is relatively faster compared with LSTM. Both LSTM and GRU have been widely used in time series forecasting, either on their own or in conjunction with other models (H. Y. Kim & Won, 2018) (Ozdemir et al., 2022) (Gu et al., 2020). Yet, the inherent sequential operations in LSTMs and GRUs make them intrinsically time-intensive. As such, a modern approach, Transformers, has been proposed.

F. Transformers

The main advantage of transformers is their ability to do parallel computing, which takes advantage of GPUs and significantly improves the speed for longer data sequences. The original use of Transformer was in translating languages (Vaswani et al., 2023) (T. Andersen et al., 2005), but soon it has seen extensive use in different tasks like audio processing (C.-Z. A. Huang et al., 2018), computer visions (Liu et al., 2021), and time series (Ahmed et al., 2022) (Zerveas et al., 2021). Transformers abandon recurrence entirely and use attention mechanisms instead. Transformers have an encoder-decoder structure. The encoder maps the input sequence and produces a continuous representation, and the decoder chooses what and how much previously encoded information to access. The encoder's attention mechanism derives attention scores from input vectors of queries, keys, and values. These scores determine the weight of each piece of information in predicting every time step. A dot product was used for simplicity in the original Transformer (Vaswani et al., 2023), while many different approaches were introduced in later works (Wen et al., 2023). To further refine and simplify the process, an array of attention mechanisms emerged.

Notable examples include LogSparse Transformer (S. Li et al., 2019), Reformer (Kitaev et al., 2020), Informer (Zhou et al., 2021), Crossformer (Y. Zhang & Yan, 2023), and Autoformer (Wu et al., 2021). As research in this area is ongoing, these mechanisms constantly advance the state of the art. These innovative mechanisms, in their distinct ways, cut time and memory requirements compared to the original transformer.

LogSparse Transformer (S. Li et al., 2019) introduces convolutional self-attention. It generates queries and keys using causal convolution, prioritizing more recent information for immediate step forecasting. Reformer (Kitaev et al., 2020) uses locality-sensitive hashing to replace simple dot products and reversible residual layers to replace standard residuals. Informer (Zhou et al., 2021) shares similarities with LogSparse by utilizing the sparsity found in the self-attention probability distribution. However, Informer identifies a long-tail distribution within the attention distribution. Therefore, it leverages the fact that a small number of dot products produce the majority of attention to selectively choose only the top queries and replaces vanilla self-attention with ProbSparse self-attention. Crossformer (Y. Zhang & Yan, 2023) identifies a gap in cross-dimensional dependency modeling. To remedy this, it introduces a Two-Stage Attention (TSA) layer to bridge this deficiency. Autoformer (Wu et al., 2021) model utilizes a decomposition layer to separate the time series into long-term trends, seasonality, and random components. Then, it replaces the self-attention with the autocorrelation mechanism, which extracts frequency-based dependencies from queries and keys instead of the vanilla dot product.

Despite transformers being regarded as a state-of-the-art (SOTA) way of forecasting volatility, Zeng et al. (Zeng et al., 2023) challenged this notion by introducing a straightforward single-layer linear model that surpassed all current transformer models across nine datasets. They argued that transformer architectures, despite their success in NLP, may not be suitable for time series forecasting. This is because the self-attention mechanism is inherently anti-order. Zeng argued that while this might not significantly impact sentences, as they retain most of their meaning even if the sequence of the words is changed, it is problematic for time series where the continuous sequence order is vital.

This perspective quickly gained attention. For instance, Nie et al. introduced PatchTST (Nie et al., 2023), addressing the anti-order issue by segmenting time steps into subseries-level patches. Although Cirstea et al. (Cirstea et al., 2022) first introduced the concept of patches for simplifying complexities, PatchTST (Nie et al., 2023) was the first to utilize them as input units. Moreover, to emphasize locality, they incorporated a channel-independence technique, previously validated in (Zheng et al., 2014). This ensures the input token is derived from a single channel, in contrast to earlier transformers that adopted channel-mixing methods. Their results indicated a marked improvement over both standard transformers and the linear model proposed by Zeng et al (Zeng et al., 2023).

While constructing the paper, Microsoft and Tsinghua University introduced a novel architecture called the Retentive Network (Sun et al., 2023). This architecture is presented as an enhancement to the Transformer model, retaining its benefits but reducing inference costs and long-sequence memory complexity. Although its primary intention is for use in language models, similar to the evolution of Transformers, it is possible that this architecture will find broader applications, like time series.

III. METHODS

A. Overview

We will discuss four milestone models: Generalized AutoRegressive Conditional Heteroskedasticity, Implied Volatility, Long Short Term Memory, and Transformer.

To compute for errors, we used both root mean square error (RMSE) and root mean square percentage error (RMSPE):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(A_i - F_i)^2}$$

$$\text{RMSPE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{A_i - F_i}{A_i}\right)^2}$$

where:

- N: Total number of observations.
- A_i: Actual value for the i^{th} observation.
- F_i: Predicted value for the i^{th} observation.

B. Data Processing

We used thirty years of daily S&P 500 data from Yahoo Finance, October 1, 1993, to October 1, 2023. We used the first twenty-seven years for training and the last three years for testing. The actual training data (November 16, 1993, to September 28, 2020) is slightly shorter than twenty-seven years because of data loss when processing data and calculating rolling returns. We calculated the realized volatility as the annualized standard deviation of 22 rolling trading days (as an approximation for one month in time)'s log return. The logarithmic return for a given day *t* is represented as:

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right)$$

where:

- r_t: Log return on day *t*.
- P_t: Price on day *t*.
- P_{t-1}: Price on the previous day, day $t - 1$.

The average log return over 22 trading days is:

$$\bar{r} = \frac{1}{22}\sum_{i=0}^{21} r_{t-i}$$

The realized volatility over 22 trading days is given by:

$$\sigma = \sqrt{\frac{1}{21}\sum_{i=0}^{21}(r_{t-i} - \bar{r})^2}$$

C. GARCH

The ARCH model was introduced prior to the GARCH model for forecasting volatility. The name, Auto Regressive Conditional Heteroskedasticity, means that volatility depends on time series value in previous periods and some error term. GARCH is a variant of the ARCH model that addresses the problem of predictions being bursty, which means the prediction can vary by a huge amount day by day. This enhancement is achieved by incorporating the previous day's volatility into the current day's calculation, alongside the ARCH model's time series value and error term. Therefore, the resulting predictions tend to be more stable, given that today's volatility is likely to mirror the previous day due to its inclusion in the equation. The equation for GARCH(p, q) is:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$$

where:

- $\sigma_t{}^2$: Conditional volatility at time t.
- α_0: positive empirical parameters.
- α_i: Non-negative empirical parameters.
- ε_{t-i}^2: the squared residual at time $t - i$.
- β_j: Non-negative empirical parameters.
- σ_{t-j}^2: Variance of the return series at time $t - j$.

While there are many different versions of GARCH models, we will use a simple GARCH (1,1) model as this model is representative, simple, and powerful (Hansen & Lunde, 2005). GARCH (1,1) considers only one lag of the squared return and one lag of the conditional variance. The equation for GARCH (1,1) model is:

$$\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$$

We used rolling forecast techniques, and the model will have the training data as well as the past test data.

D. Implied Volatility

What distinguishes IV from other models is that it is forward-looking. Implied volatility captures the market's expectation of the volatility for the next 22 days, calculated backward from the option's price using the Black Scholes Merton Formula:

$$c = S_0 N(d_1) - K e^{-rT} N(d_2)$$

$$p = K e^{-rT} N(-d_2) - \ S_0 N(-d_1)$$

where:

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r - q + \frac{\sigma^2}{2}\right) T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

Explanation of terms:

- c: Price of the European call option.
- p: Price of the European put option.
- S_0: Current stock price.
- K: Strike price of the option.
- T: Time to maturity (in years).
- r: Risk-free rate.
- q: Continuous dividend yield.
- N(.): the probability that a variable with a standard normal distribution will be less than x.
- σ: Volatility of the stock's return.

While it is impossible to invert the function to calculate implied volatility directly as a function of other variables, an iterative approach can be used to search for implied volatility (Hull, 2018). In this paper, we used the corresponding daily close of the VIX index as the implied volatility and compared it with the realized volatility of the S&P 500. The VIX data is available from the Chicago Board Options Exchange (CBOE).

E. LSTM

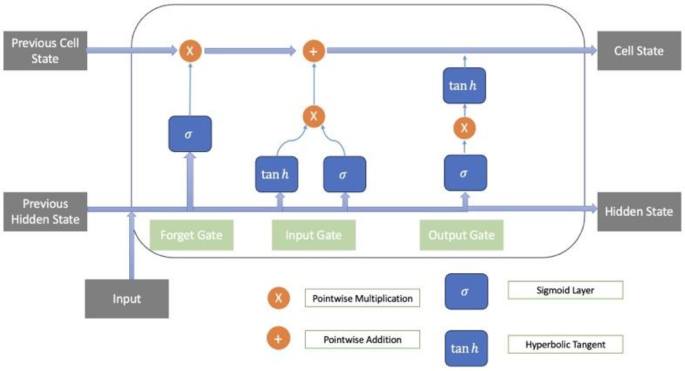


Fig. 2: Long Short Term Memory

LSTM specifically utilizes sigmoid and tanh activation functions. The sigmoid function confines any input value within a range of 0 to 1, whereas the tanh function limits it between -1 and 1. With the current and previous information, these activation functions determine the amount of previous information to keep or discard in the forget gate. If the forget gate outputs 0, it forgets everything; if it outputs 1, it remembers everything. Then, these functions are used to decide the input and output gates. In the output gate, potential short-term memory and its corresponding retention percentage are considered to calculate the ensuing short-term memory. This then becomes the output for the current LSTM cell and the input for the subsequent one. The long-term memory receives updates by initially processing the forgotten state and then assimilating it with the input state. This iterative updating of short-term and long-term memory persists till the model concludes its operation. The equations for LSTM are:

The cell state (C):

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C_t}$$

The forget gate (f):

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

The input gate (i):

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

The output gate (o):

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

The hidden state h_t:

$$h_t = o_t \times \tanh(C_t)$$

The candidate for cell state at timestamp t ($\widetilde{C_t}$):

$$\widetilde{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

where:

- σ: sigmoid activation function
- W_f, W_i, W_C, and W_o: weight matrices for the forget gate, input gate, candidate values, and output gate respectively

- b_f, b_i, b_C, and b_o: biases corresponding to each gate.
- h_{t-1}: last hidden state
- x_t: current input
- tanh is the hyperbolic tangent activation function.

Our model used a 22-day windowed dataset, a batch size of 64 and 200 epochs to train the two-layered bidirectional LSTM model. We used bidirectional LSTM because it retains sequence information both forward and backward, which helps the model better understand the context—a crucial aspect in forecasting volatility. We use Adam optimizer to optimize our model and only keep the best model and employ an early stopping with patience of 20.

F. Transformer

We built a vanilla transformer that is modified to work with time series tasks. We processed our data in batches of 128 over 200 epochs. During preprocessing, we windowed our dataset into 22-day segments and reshaped it into three dimensions.

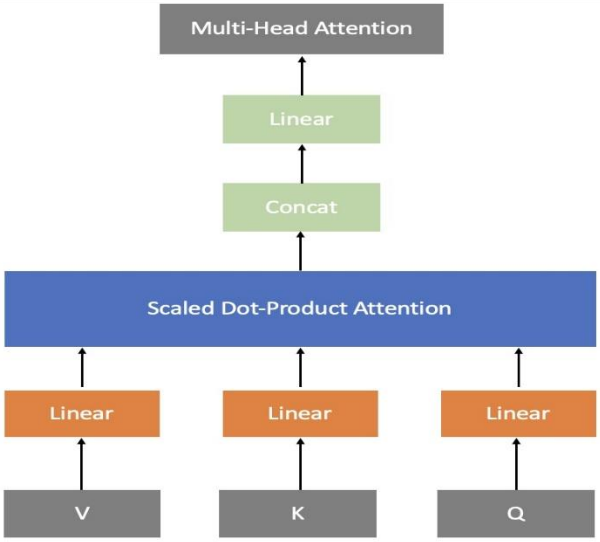


Fig. 3: Multi-Head Attention (Transformer)

We used 4 layers for our encoder. Each layer contains Layer Normalization, Multi-Head Attention, Dropout, Residual Connection, and a Feed Forward Network. The process begins with Layer Normalization, which normalizes the input data to have zero mean and unit variance. Subsequently, the Multi-Head Attention calculates a weighted sum of the input based on its relationships with other parts of the input. Dropout is then applied to regularize the network. It achieves this by randomly setting a fraction of the input units to 0 at each update during training, which helps prevent overfitting. The equation for Multi-Head Attention is:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W_O$$

$$\text{head}_i = \text{Attention}\left(QW_{Q_i}, KW_{K_i}, VW_{V_i}\right)$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The Residual Connection assists in counteracting the vanishing gradient problem encountered in deep networks. Finally, the Feed Forward Network comprises 1-D convolutional filters with a RELU activation function (Nair & Hinton, 2005) that replaces the feed forward layer in the original transformer.

IV. RESULTS

A. Limitations

To ensure universality and fairness in comparison, we constructed models for each category that are general and representative instead of fine-tuned for specific tasks. While these models provide an adequate representation for the purpose of this benchmark, further optimization could reveal the full potential of each model's performance.

Furthermore, the 30-year data is small for the Transformer model, which was originally used for large language processing tasks. This is part of why Transformer does not perform better than older models like LSTM. In addition, there were numerous changes in the stock market throughout the 30-year period; this evolving nature of the stock market made the forecasting especially complex. Although the non-machine learning models perform worse than the ML models, they do not require a large dataset, making them advantageous in settings where data is limited.

B. Experimental Evaluation

We used historical VIX close value, which makes the computation faster than what would happen intraday in real-time, which is updated every 15 seconds. In the thirty years of S&P 500 data we used to train and test our models, two-layered LSTM performs the best, followed by Transformer. The results are shown in Table I and Figure 5.

V. Performance During Crisis Periods

A. Overview

To test the models' performance during extreme scenarios where volatility spikes, we selected two recent crisis periods: the 2008 Financial Crisis and the 2020 COVID-19 Pandemic. We kept the other parameters in the models unchanged except for using different periods of training and testing data for each case.

B. 2008 Financial Crisis

We used a training period from 1997-01-16 to 2007-01-23 and a testing period from 2007-01-24 to 2010-01-22. The result is shown in Fig.6.

C. 2020 COVID-19 Pandemic

We used two different data periods for the training part of the COVID-19 case to compare the performance of machine learning models when given different lengths of training data. Specifically, we selected two periods: one that does not include the 2008 Financial Crisis, and another that does. The first period uses data from 2010-03-01 to 2020-03-03 for training and from 2020-03-04 to 2023-03-03 for testing. The second period uses data from 2005-02-25 to 2020-03-03 for training and from 2020-03-04 to 2023-03-03 for testing. As shown in the results section, the Transformer model trained with the longer dataset that includes the Financial Crisis period (Fig. 8) performed significantly better than the one without it (Fig. 7) in predicting the spike in volatility at the onset of the COVID-19 pandemic. The LSTM model's performance is similar in both cases.

Model Name	RMSPE (Base)	RMSE (Base)	RMSPE (2008)	RMSE (2008)	RMSPE (COVID1)	RMSE (COVID1)	RMSPE (COVID2)	RMSE (COVID2)
GARCH (1,1)	0.1581	0.0251	0.1158	0.0320	0.2019	0.0507	0.2019	0.0507
Implied Volatility	0.6014	0.0701	0.4035	0.0902	0.6736	0.1201	0.6736	0.1201
2-layered LSTM	0.0400	0.0070	0.0605	0.0133	0.0548	0.0121	0.0492	0.0110
Transformer	0.0408	0.0058	0.0856	0.0402	0.0848	0.0577	0.0576	0.0129

TABLE I: Comparison of Different Models

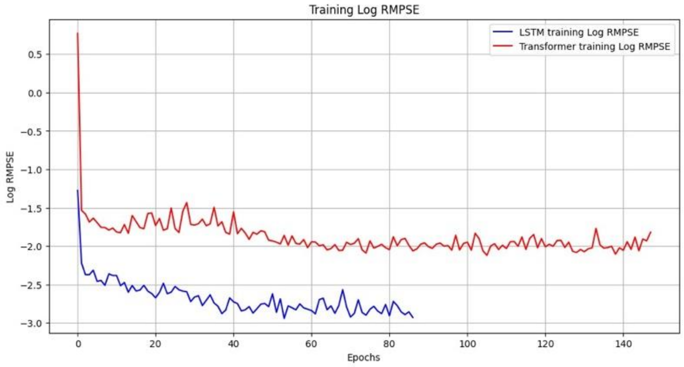


Fig. 4: The Training Error of LSTM and Transformer

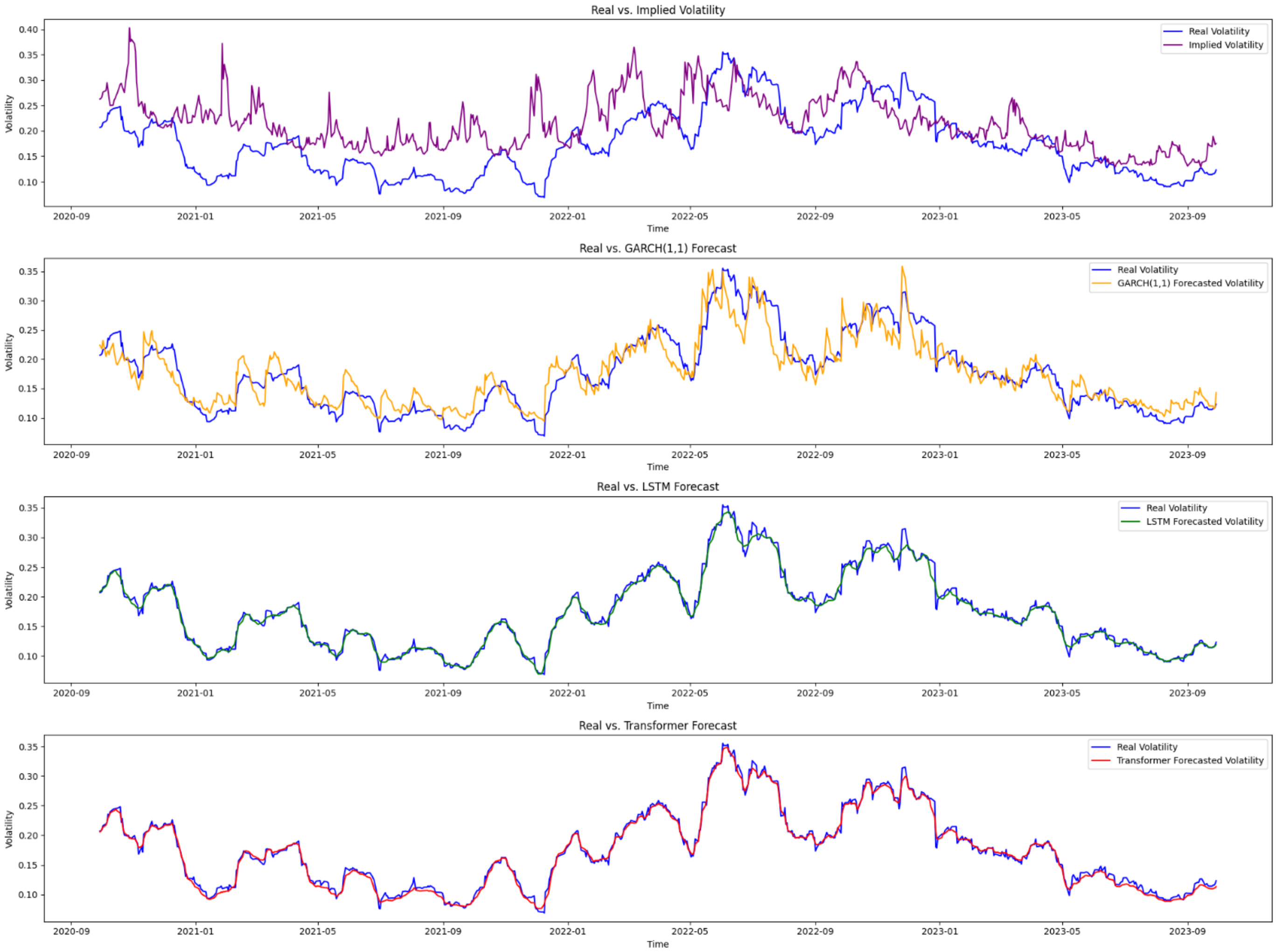


Fig. 5: Comparison of Performance (Base)

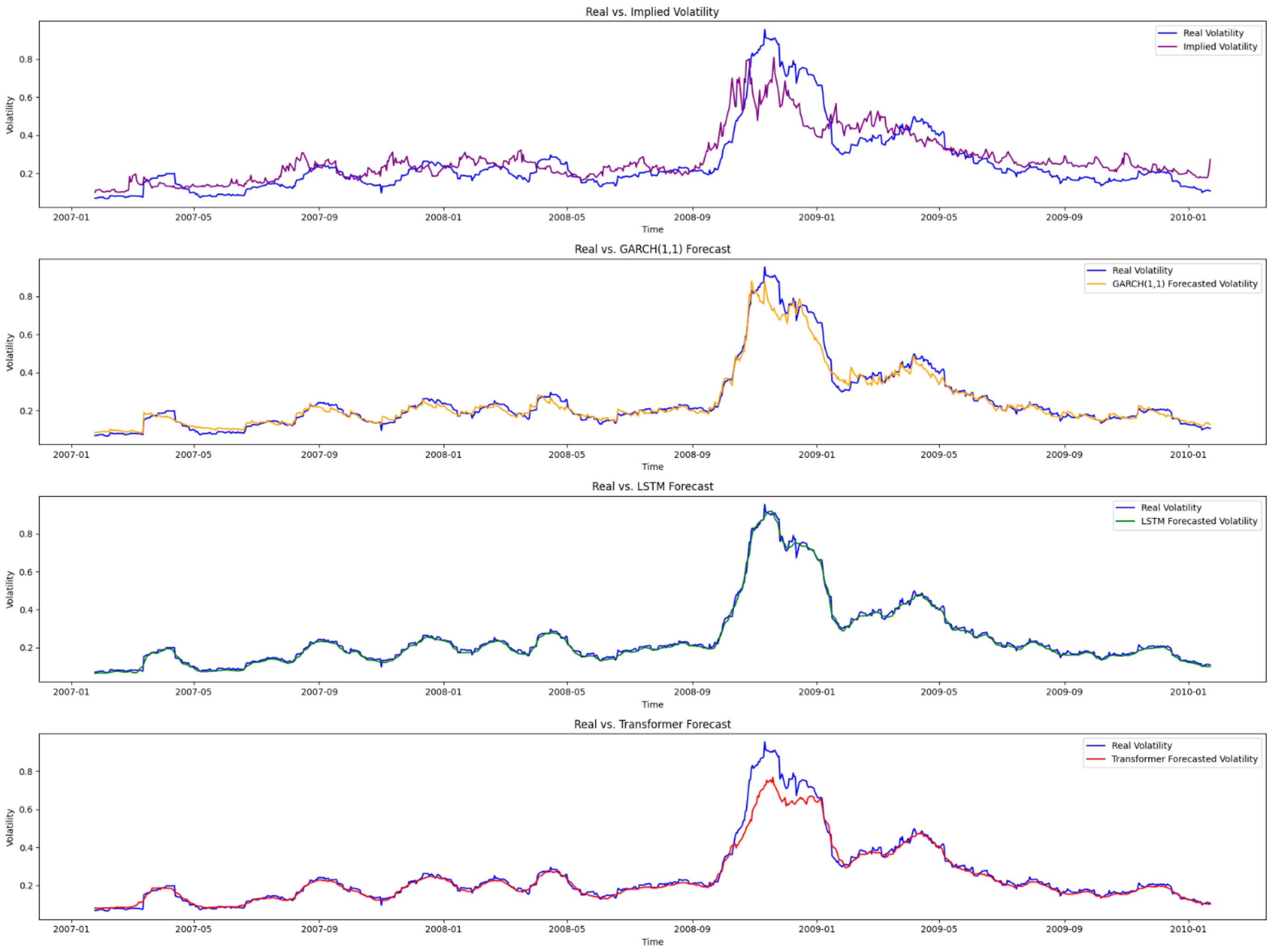
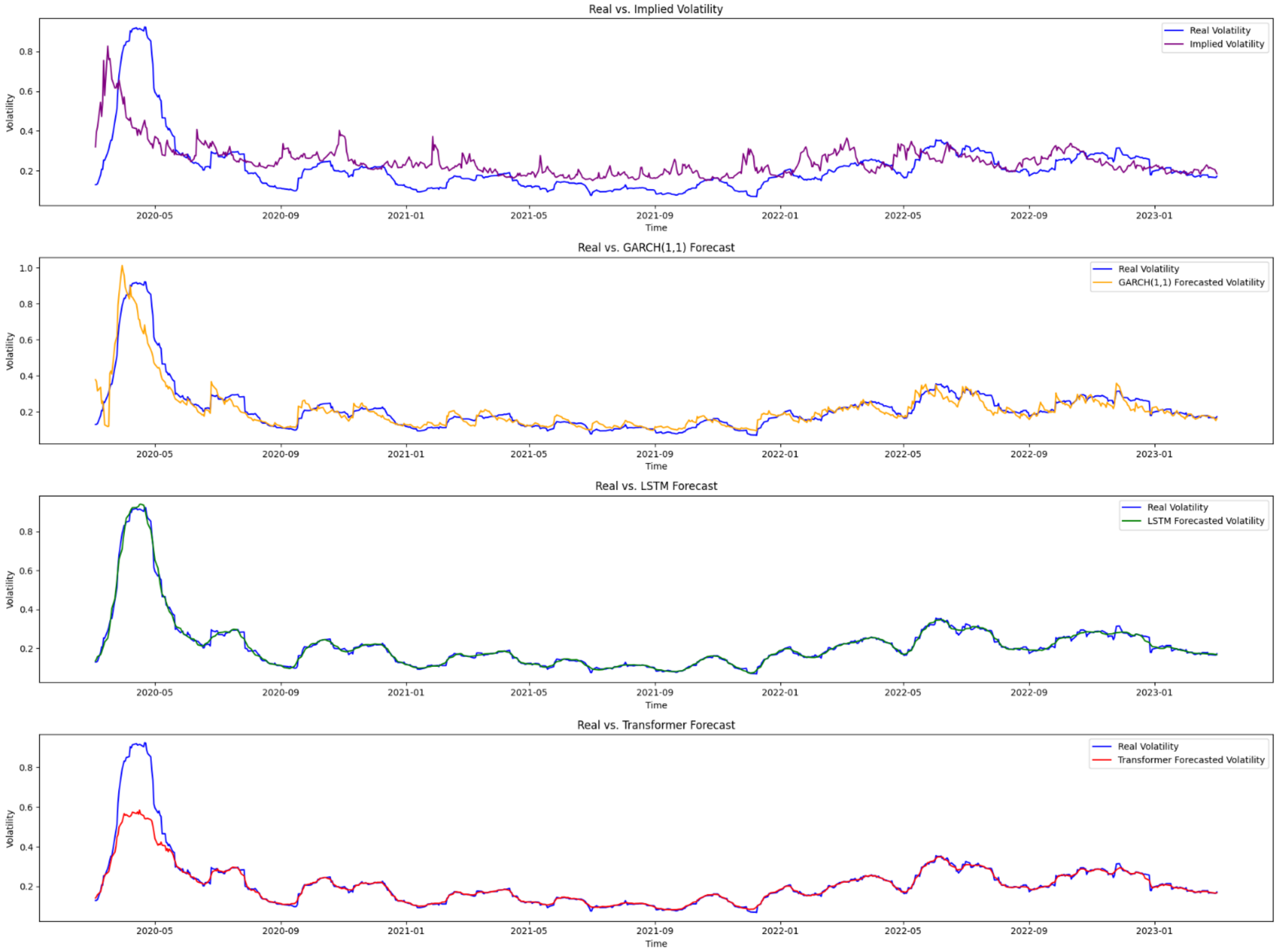


Fig. 6: Comparison of Performance (Financial Crisis Period)


Real vs. Implied Volatility

Real vs. GARCH(1,1) Forecast

Real vs. LSTM Forecast

Real vs. Transformer Forecast

Fig. 7: Comparison of Performance (COVID with 10 years Training)

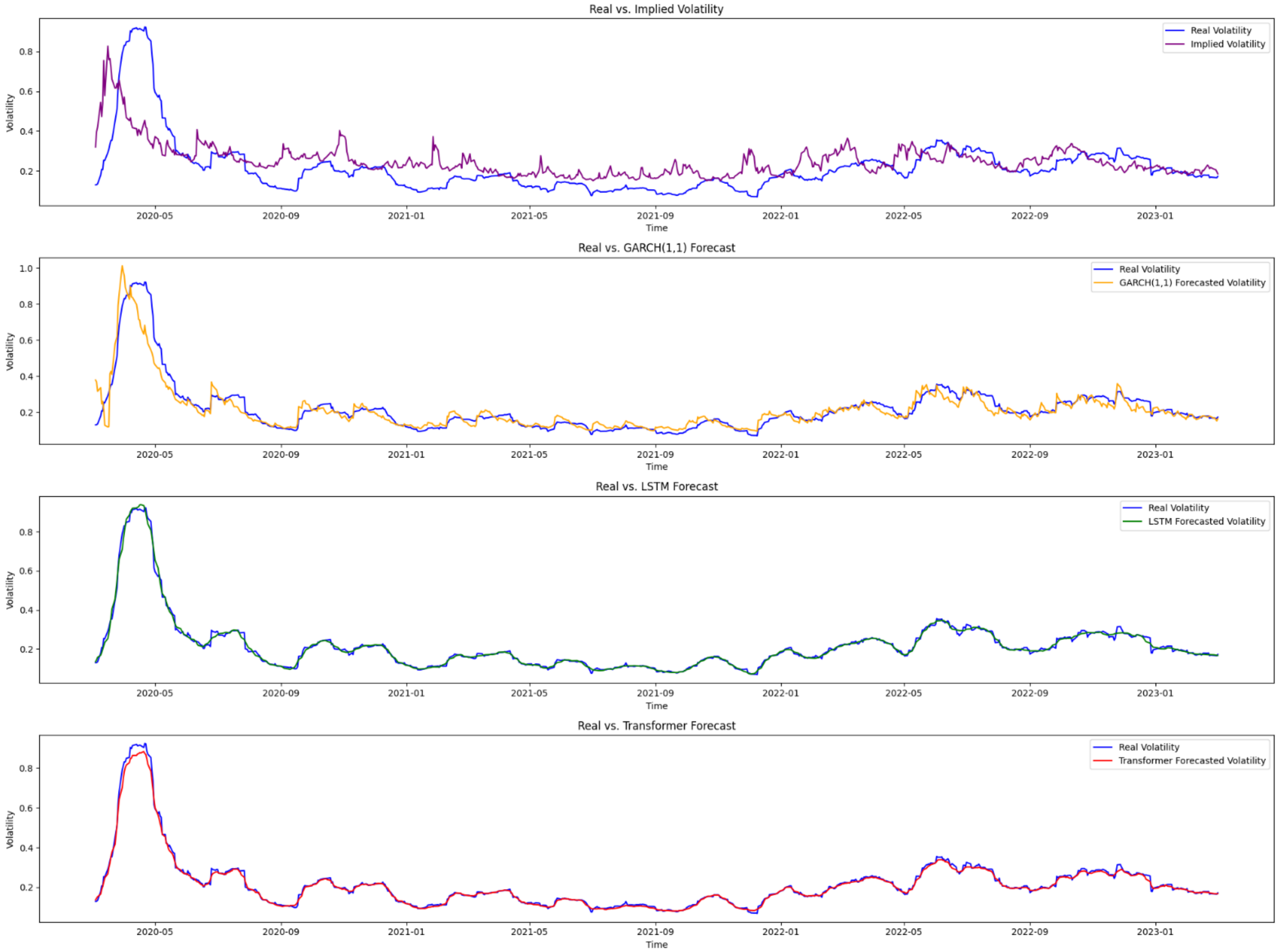


Fig. 8: Comparison of Performance (COVID with 15 years Training)

VI. CONCLUSION

Volatility contributes as an essential part of financial institutions' risk exposure, which makes the forecasting of it an important task. A benchmark comparing the efficacy and performance of different forecasting techniques can be beneficial. While some existing literature reviews focus on specific models, like GARCH, there remains a gap for a holistic and up-to-date benchmark. Our study summarizes the key attributes of market volatility, such as its dynamic nature, clustering behavior, long memory, heavy tails, and the asymmetric relationship between prices and volatility.

The study also offers a comprehensive review of volatility forecasting methods, ranging from traditional models to the current state-of-the-art. Traditional models like GARCH have historically performed well, but with recent advancements in machine learning, algorithms such as LSTMs and Transformers have enhanced forecasting accuracy. Specifically, in the thirty-year dataset we use, the two-layered LSTM model produces an RMSPE of only 0.04, the Transformer model produces an RMSPE of 0.0408, while the GARCH model produces an RMSPE of 0.1581. Similar results have been shown in testing with other time intervals, as detailed in the results section. This overperformance is a trend that's expected to advance as even more sophisticated algorithms emerge.

Nonetheless, machine learning models are not perfect. Financial data is small compared to other applications of machine learning and can only be generated through passage of time. As demonstrated by the extreme case study of the COVID-19 period, a lack of sufficient historical data can lead to a model's inability to predict sudden spikes in volatility. Additionally, these models demand significant computational resources and extended training times. Their results can also be challenging to interpret intuitively.

VII. FUTURE WORKS

In this survey paper, we used baseline models to ensure universality, while more sophisticated models and hyperparameter tuning can further improve accuracy for the models we mentioned. Despite the inherent challenges in predicting volatility due to its sensitivity to a multitude of factors, including economic, corporate, psychological, and unforeseeable exogenous shocks, our research has shown that forecasting volatility is possible and that its accuracy is expected to increase as we employ more sophisticated models that can help capture more intricate dataset. It has been demonstrated that a combination of existing models can improve forecasting accuracy. We also believe that state-of-the-art models, such as Retentive Networks, hold potential for future applications in volatility prediction.

ACKNOWLEDGEMENTS

This work was supported in part by the Keck Institute Undergraduate Research Grant at Pepperdine University.

REFERENCES

Abdel-Hamid, O., Mohamed, A., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *22*(10), 1533–1545. https://doi.org/10.1109/TASLP.2014.2339736

Ahmed, S., Nielsen, I. E., Tripathi, A., Siddiqui, S., Rasool, G., & Ramachandran, R. P. (2022). *Transformers in Time-series Analysis: A Tutorial*. https://doi.org/10.48550/ARXIV.2205.01138

Aıt-Sahalia, Y. (2017). *Estimation of the Continuous and Discontinuous Leverage Effects*.

Andersen, T., Bollerslev, T., Christoffersen, P., & Diebold, F. (2005). *Volatility Forecasting* (w11188; p. w11188). National Bureau of Economic Research. https://doi.org/10.3386/w11188

Andersen, T. G. (Ed.). (2018). *Volatility*. Edward Elgar Publishing, Inc.

Bekaert, G., & Wu, G. (2000). Asymmetric Volatility and Risk in Equity Markets. *Review of Financial Studies*, *13*(1), 1–42. https://doi.org/10.1093/rfs/13.1.1

Bildirici, M., & Ersin, Ö. (2015). Forecasting volatility in oil prices with a class of nonlinear volatility models: Smooth transition RBF and MLP neural networks augmented GARCH approach. *Petroleum Science*, *12*(3), 534–552. https://doi.org/10.1007/s12182-015-0035-8

Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *The Journal of Political Economy*, *81*(3,), 637–654.

Bollerslev, T. (1986). *GENERALIZED AUTOREGRESSIVE CONDITIONAL HETEROSKEDASTICITY*.

Bollerslev, T. (2007). *Glossary to ARCH (GARCH)*.

Bollerslev, T., & Melvin, M. (1994). Bid—Ask spreads and volatility in the foreign exchange market. *Journal of International Economics*, *36*(3–4), 355–372. https://doi.org/10.1016/0022-1996(94)90008-6

Box, G. E. P., & Pierce, D. A. (1970). Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *Journal of the American Statistical Association*, *65*(332), 1509–1526. https://doi.org/10.1080/01621459.1970.10481180

Breiman, L. (2001). Random Forest. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

CBOE. (2019). *Volatility Index Methodology: Cboe Volatility Index*.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chow, T. W. S., & Leung, C. T. (1996). Neural network based short-term load forecasting using weather compensation. *IEEE Transactions on Power Systems*, *11*(4), 1736–1742. https://doi.org/10.1109/59.544636

Christie, A. (1982). The stochastic behavior of common stock variances Value, leverage and interest rate effects. *Journal of Financial Economics*, *10*(4), 407–432. https://doi.org/10.1016/0304-405X(82)90018-6

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. https://doi.org/10.48550/ARXIV.1412.3555

Cirstea, R.-G., Guo, C., Yang, B., Kieu, T., Dong, X., & Pan, S. (2022). Triformer: Triangular, Variable-Specific Attentions for Long Sequence Multivariate Time Series Forecasting. *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 1994–2001. https://doi.org/10.24963/ijcai.2022/277

Connor, J. T., Martin, R. D., & Atlas, L. E. (1994). Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, *5*(2), 240–254. https://doi.org/10.1109/72.279188

Cont, R. (2001). Empirical properties of asset returns: Stylized facts and statistical issues. *QUANTITATIVE FINANCE*.

Derman, E. (1999). *More Than You Ever Wanted to Know About Volatility Swaps*.

Diamond, R. V. (2012). VIX as a Variance Swap. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2030292

Engle, R. (2004). Risk and Volatility: Econometric Models and Financial Practice. *American Economic Review*, *94*(3), 405–420. https://doi.org/10.1257/0002828041464597

Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, *50*(4), 987. https://doi.org/10.2307/1912773

Engle, R. F., & Ng, V. K. (1993). Measuring and Testing the Impact of News on Volatility. *The Journal of Finance*, *48*(5), 1749–1778. https://doi.org/10.1111/j.1540-6261.1993.tb05127.x

Ge, W., Lalbakhsh, P., Isai, L., Lenskiy, A., & Suominen, H. (2023). Neural Network–Based Financial Volatility Forecasting: A Systematic Review. *ACM Computing Surveys*, *55*(1), 1–30. https://doi.org/10.1145/3483596

Goudarzi, H. (2013). VOLATILITY MEAN REVERSION AND STOCK MARKET EFFICIENCY. *Asian Economic and Financial Review*.

Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies. (2009). In J. F. Kolen & S. C. Kremer, *A Field Guide to Dynamical Recurrent Networks*. IEEE. https://doi.org/10.1109/9780470544037.ch14

Gu, W., Zheng, S., Wang, R., Dong, C., & School of Statistics and Mathematics, Zhejiang Gongshang University 18 Xuezheng Street, Xiasha Education Park, Hangzhou, Zhejiang 310018, China. (2020). Forecasting Realized Volatility Based on Sentiment Index and GRU Model. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, *24*(3), 299–306. https://doi.org/10.20965/jaciii.2020.p0299

Hansen, P. R., & Lunde, A. (2005). A forecast comparison of volatility models: Does anything beat a GARCH(1,1)? *Journal of Applied Econometrics*, *20*(7), 873–889. https://doi.org/10.1002/jae.800

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, *20*(1), 5–10. https://doi.org/10.1016/j.ijforecast.2003.09.015

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A. M., Hoffman, M. D., Dinculescu, M., & Eck, D. (2018). *Music Transformer* (arXiv:1809.04281). arXiv. http://arxiv.org/abs/1809.04281

Huang, D., Jiang, F., Li, K., Tong, G., & Zhou, G. (2022). Scaled PCA: A New Approach to Dimension Reduction. *Management Science*, *68*(3), 1678–1695. https://doi.org/10.1287/mnsc.2021.4020

Hull, J. (2018). *Options, futures, and other derivatives* (Tenth Edition). Pearson.

Johnston, F. R., Boyland, J. E., Meadows, M., & Shale, E. (1999). Some properties of a simple moving average when applied to forecasting a time series. *Journal of the Operational Research Society*, *50*(12), 1267–1271. https://doi.org/10.1057/palgrave.jors.2600823

Kelly, B. T., & Xiu, D. (2023). Financial Machine Learning. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4501707

Kim, D., & Shin, M. (2023). Volatility models for stylized facts of high-frequency financial data. *Journal of Time Series Analysis*, *44*(3), 262–279. https://doi.org/10.1111/jtsa.12666

Kim, H. Y., & Won, C. H. (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. https://doi.org/10.1016/j.eswa.2018.03.002

Kitaev, N., Kaiser, Ł., & Levskaya, A. (2020). *REFORMER: THE EFFICIENT TRANSFORMER*.

Knight, F. (1921). *RISK, UNCERTAINTY AND PROFIT*. Houghton Mifflin Company.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., & Yan, X. (2019). *Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows* (arXiv:2103.14030). arXiv. http://arxiv.org/abs/2103.14030

Loh, W. (2011). Classification and regression trees. *WIREs Data Mining and Knowledge Discovery*, *1*(1), 14–23. https://doi.org/10.1002/widm.8

Ludvigson, S. C., & Ng, S. (2007). The empirical risk–return relation: A factor analysis approach☆. *Journal of Financial Economics*, *83*(1), 171–222. https://doi.org/10.1016/j.jfineco.2005.12.002

Mandelbrot, B. B. (1997). The variation of certain speculative prices. In B. B. Mandelbrot, *Fractals and Scaling in Finance* (pp. 371–418). Springer New York. https://doi.org/10.1007/978-1-4757-2763-0_14

Marcek, D. (2018). Forecasting of financial data: A novel fuzzy logic neural network based on error-correction concept and statistics. *Complex & Intelligent Systems*, *4*(2), 95–104. https://doi.org/10.1007/s40747-017-0056-6

McAleer, M., & Medeiros, M. C. (2008). Realized Volatility: A Review. *Econometric Reviews*, *27*(1–3), 10–45. https://doi.org/10.1080/07474930701853509

Nair, V., & Hinton, G. E. (2005). *Rectified Linear Units Improve Restricted Boltzmann Machines*.

Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). *A Time Series is Worth 64 Words: Long-term Forecasting with Transformers* (arXiv:2211.14730). arXiv. http://arxiv.org/abs/2211.14730

Ozdemir, A. C., Buluş, K., & Zor, K. (2022). Medium- to long-term nickel price forecasting using LSTM and GRU networks. *Resources Policy*, *78*, 102906. https://doi.org/10.1016/j.resourpol.2022.102906

Pomerleau, D. A. (1988). *ALVINN, an autonomous land vehicle in a neural network*.

Poon, S.-H., & Granger, C. (2003). *Forecasting Volatility in Financial Markets: A Review*.

Pranav, B. (2021). *Volatility Forecasting Techniques using Neural Networks: A Review*.

Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, *90*, 106181. https://doi.org/10.1016/j.asoc.2020.106181

Shiller, R. J. (1999). *Market volatility* (1. paperback ed., [Nachdr.]). MIT Press.

Stock, J. H., & Watson, M. W. (2002). Forecasting Using Principal Components From a Large Number of Predictors. *Journal of the American Statistical Association*, *97*(460), 1167–1179. https://doi.org/10.1198/016214502388618960

Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., & Wei, F. (2023). *Retentive Network: A Successor to Transformer for Large Language Models* (arXiv:2307.08621). arXiv. http://arxiv.org/abs/2307.08621

Taylor, J. W. (2004). Smooth transition exponential smoothing. *Journal of Forecasting*, *23*(6), 385–404. https://doi.org/10.1002/for.918

Tversky, A., & Kahneman, D. (1991). Loss Aversion in Riskless Choice: A Reference-Dependent Model. *The Quarterly Journal of Economics*, *106*(4), 1039–1061. https://doi.org/10.2307/2937956

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (arXiv:1706.03762). arXiv. http://arxiv.org/abs/1706.03762

Venugopal, V., & Baets, W. (1994). Neural Networks and Statistical Techniques in Marketing Research: A Conceptual Comparison. *Marketing Intelligence & Planning*, *12*(7), 30–38. https://doi.org/10.1108/02634509410065555

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., & Sun, L. (2023). Transformers in Time Series: A Survey. *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 6778–6786. https://doi.org/10.24963/ijcai.2023/759

Whaley, R. E. (2009). Understanding the VIX. *The Journal of Portfolio Management*, *35*(3), 98–105. https://doi.org/10.3905/JPM.2009.35.3.098

Wu, H., Xu, J., Wang, J., & Long, M. (2021). *Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting*.

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are Transformers Effective for Time Series Forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*(9), 11121–11128. https://doi.org/10.1609/aaai.v37i9.26317

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., & Eickhoff, C. (2021). A Transformer-based Framework for Multivariate Time Series Representation Learning. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2114–2124. https://doi.org/10.1145/3447548.3467401

Zhang, C., Zhang, Y., Cucuringu, M., & Qian, Z. (2023). *Volatility forecasting with machine learning and intraday commonality* (arXiv:2202.08962). arXiv. http://arxiv.org/abs/2202.08962

Zhang, Y., & Yan, J. (2023). *CROSSFORMER: TRANSFORMER UTILIZING CROSS- DIMENSION DEPENDENCY FOR MULTIVARIATE TIME SERIES FORECASTING*.

Zheng, Y., Liu, Q., Chen, E., Ge, Y., & Zhao, J. L. (2014). Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks. In F. Li, G. Li, S. Hwang, B. Yao, & Z. Zhang (Eds.), *Web-Age Information Management* (Vol. 8485, pp. 298–310). Springer International Publishing. https://doi.org/10.1007/978-3-319-08010-9_33

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325