*Corresponding Author’s Email: admi.souhail@yahoo.com
Proceedings of the International Conference on Conference on Business, Management and Finance
Vol. 2, Issue. 1, 2025, pp. 45-52
DOI: https://doi.org/10.33422/icbmf.v2i1.1048

ISSN: 3030-0959 online

Uncovering Nonlinear Patterns in Stock Returns Using Gated Recurrent Unit (GRU) Model

Admi Souhail
Chouaib Doukkali University, National School of Commerce and Management, Laboratory for Studies and Research in Economic and Management Sciences, Corner of Ahmed Chaouki Avenue and Fes Street, P.O. Box 122 - 24000 El Jadida, Morocco

Abstract

The escalating complexity of financial markets has driven the adoption of advanced machine learning techniques, such as Gated Recurrent Unit (GRU) models, to uncover nonlinear patterns in stock returns that traditional linear methods fail to capture. While the EMH posits that stock prices follow a random walk, empirical evidence increasingly challenges this assumption, revealing nonlinear dependencies. The GRU achieves an R^2 of 0.0131, explaining only 0.13% of the variance in log returns, which underscores the inherent difficulty of forecasting noisy financial data. The observed pattern suggests marginal but exploitable predictability in returns, valuable in high-frequency trading. GRU models can uncover subtle nonlinear dependencies, enhancing both risk management and dynamic asset allocation. This highlights their practical utility in modern quantitative finance for capturing small but impactful market inefficiencies.

Keywords: Neural Networks, Gated Recurrent Unit, Efficient Market Hypothesis, Nonlinearity, Volatility Clustering

1. Introduction

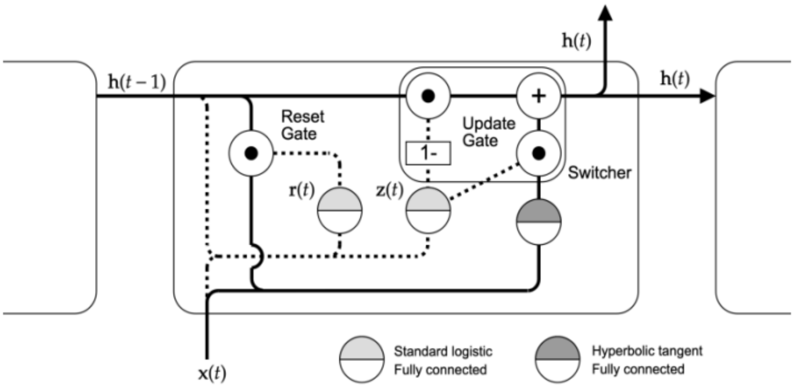
The Efficient Market Hypothesis (EMH) (Fama, 1970) asserts that asset prices fully reflect all available information and follow a random walk (Bachelier, 1900), implying that historical data holds no predictive value and rendering it impossible to outperform the market consistently. This theory has long dominated financial theory. However, a growing body of empirical evidence challenges its assumptions, particularly regarding the nonlinearity and non-Gaussianity of stock returns. Traditional linear models often fail to capture these complexities (Fama, 1965). (Mandelbrot, 1963) observed that financial time series frequently exhibit heavy-tailed distributions, deviating significantly from the Gaussian assumptions underlying the EMH. Similarly, (MacKinlay, 1988) identified persistent patterns in small-cap stock returns, contradicting the notion of purely random price movements.

Few academics today fully endorse the EMH (Brown, 2020), given mounting evidence of market inefficiencies. External factors, such as rumors, geopolitical events, corporate announcements, market sentiment, and government policies, can significantly influence stock prices (Pattanayak, 2024). These factors often lead to non-random price movements that contain identifiable patterns, suggesting that stock market behavior is at least partially predictable. However, when these patterns are nonlinear, traditional linear models prove inadequate. In such cases, advanced computational approaches, such as deep learning, are necessary to capture complex dependencies. The Gated Recurrent Unit (GRU) model (Kyunghyun Cho, 2014) offers a promising solution by effectively modeling temporal and nonlinear relationships in financial time series.

This study employs the GRU model to investigate whether stock returns contain exploitable patterns, thereby testing the validity of the EMH. Identifying such patterns could empower algorithmic traders, enhance investment strategies, and improve forecasting accuracy, leading to better risk management and portfolio optimization. For investment managers, implementing sophisticated advanced techniques like the GRU model can provide a strategic advantage in an increasingly data-driven industry. This paper aims to equip quantitative traders with a tool to detect short-term nonlinear predictability by capturing short volatility clustering, enabling dynamic hedging strategies.

2. Methods

This study employs daily adjusted closing prices of NVIDIA stock over the period from August 2021 to August 2024. To prepare the data for modeling, returns are calculated using log-differences, a common technique that transforms the data into a stationary series suitable for time series analysis. Initially, an autoregressive (AR) model is applied to examine the presence of linear dependencies in stock returns, serving as a benchmark for evaluating market efficiency under the framework of the EMH. However, given the known limitations of linear models in capturing complex patterns (Zaiyong Tang, 1993) in financial data, the study proposes the use of a Gated Recurrent Unit (GRU) neural network to identify potential nonlinear structures. The dataset is partitioned into training and testing sets, comprising 75% and 25% of the total observations, respectively, and all data is scaled using min-max normalization to ensure that input values fall within a consistent range, thereby facilitating stable model convergence. Next, input-output pairs are generated for supervised learning. Then, the data is reshaped into three-dimensional tensors to meet the input format required by PyTorch for training the GRU. The architecture of the latter is employed because of its ability to capture dependencies in sequential data through the use of update and reset gates, which regulate the flow of information across time steps. This is particularly relevant for financial applications where latent effects can drive nonlinear patterns in asset return dynamics.



Gated Recurrent Unit (GRU)

Source : Troiano, Bhandari, And Villa 2020

The update (or input) gate ($z(t)=\sigma[W_z x(t)+U_z h(t\text{-}1)+b_z]$) controls how much the current input $x(t)$, and the previous output $h(t\text{-}1)$ must be passed to the next cell. The activation function used in this equation is the sigmoid function (σ). On the other hand, the reset gate ($r(t)=\sigma[W_r x(t)+U_r h(t\text{-}1)+b_r]$), which also uses the sigmoid activation function, decides how much of the past information to forget. The current memory content ensures that only the relevant information will pass to the next iteration, which is regulated by the weight W, with b as the bias. The output of the reset gate is used by the new candidate's hidden state $\tilde{h}(t)$ as it is described as: $\tilde{h}(t)=\tanh\{W_h x(t)+U_h[r(t)\odot h(t\text{-}1)]+b_h$, where the tanh activation function is adopted here, while $\odot$ represents element-wise multiplication, and the final hidden state equation $h(t)$ is shown as: $h(t)=z(t)\odot h(t\text{-}1)+[1\text{-}z(t)]\odot \tilde{h}(t)$ where $z(t)\odot h(t\text{-}1)$ is the contribution of the previous hidden state scaled by the update gate and $[1\text{-}z(t)]\odot \tilde{h}(t)$ Is the contribution of the candidate hidden state scaled by the complement of the update gate. The GRU does not directly apply an activation function to its final output, as it is primarily designed for sequence learning and state updating. Therefore, after processing the input sequence, the output is typically passed to a final output layer where the activation function is applied. A linear activation function is applied at the output layer to generate continuous-valued forecasts suitable for financial time series.

The next step involves determining the number of epochs, which represent a complete cycle through the training set using subsets of the data known as mini-batches. The number of epochs during each training process varied between 30 and 300 (Wang, 2024) (Risheng Qiao, 2022).

In our case, it is fixed to 100 with a 32 batch size since the best performance has been consistently obtained for mini-batch sizes between m=2 and m=32 (Latha, 2023) (Dominic Masters, 2018). Additionally, identifying the optimal number of hidden nodes is critical. However, there is no specific method available for calculating the exact number of hidden units needed. In practice, a simpler network structure with fewer hidden nodes often yields good results in out-of-sample forecasting (Zhang, 2003). Best practice leads to choosing 64 (Greff et al., 2017). Additionally, determining the optimal number of hidden layers is crucial for an accurate architecture. Currently, no formal methods are in place to establish the optimal architecture. networks with two hidden layers have been shown to outperform those with only one (Alan J. Thomas, 2017).

Concerning the optimization process, the Adam algorithm (Diederik P. Kingma, 2014), is performed as it is known for its computational efficiency and adaptive learning rate, which is fixed in this paper at 0.001 (Smith, 2018). The model's training involves forward backpropagation through time (BPTT), with performance assessed through the mean squared error (MSE) as a loss function. At each epoch, both training and testing losses are recorded to monitor learning progress and to detect potential overfitting or underfitting after applying a dropout with a value of 0.2, consistent with best practices for regularization in similar architectures.

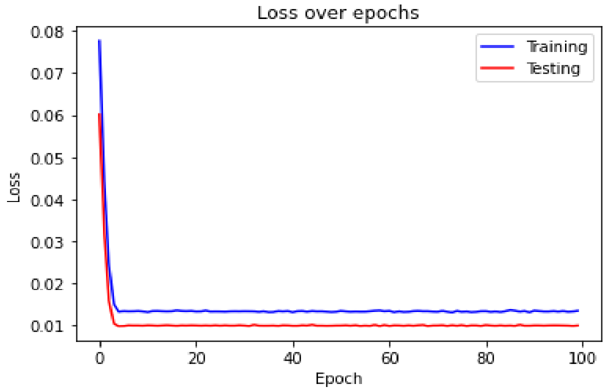
Final evaluation metrics include the R^2 on the test set, providing a comprehensive measure of predictive accuracy and explanatory power. In the current study, this metric is used to evaluate the degree of market efficiency applied to the test set, where values close to zero indicate consistency with EMH, and values significantly greater than zero suggest nonlinear dependencies and potential inefficiency in the market (White, 1988). This methodological framework aims to assess whether nonlinear deep learning models like GRU can outperform the benchmarks in forecasting stock returns and thus provide empirical evidence regarding the (in)efficiency of financial markets.

The following section evaluates whether these design choices succeeded in capturing statistically significant dependencies and uncovering subtle nonlinearities in efficient markets by presenting training and testing results, comparing predicted and actual returns, and analyzing whether the GRU model's performance offers meaningful insights into the predictability of financial time series.

3. Results and discussion

Consistent with the GRU's design described in the previous section, Figure 1 shows closely aligned training and testing loss curves, confirming the model's generalization capability. The plot, depicting the train-test split, shows that the test set performance remains under the training set, indicating no clear signs of overfitting. This suggests the GRU model generalizes adequately to unseen data, avoiding memorization of noise, as the curves remain closely aligned with no divergence.

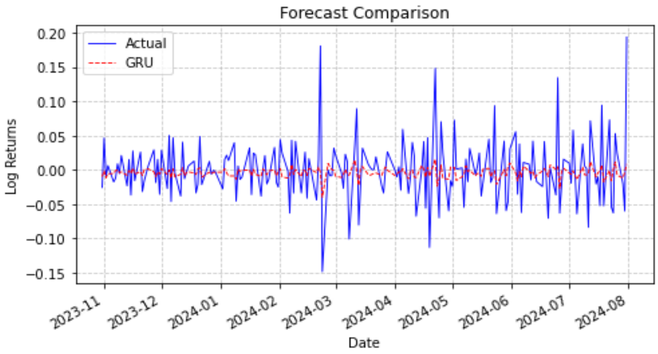
Figure 1: Training and Validation Loss Curves of the GRU Model



Source: Author's analysis

Regarding the second plot, it compares the actual log returns with those predicted by the GRU model. Although certain short-term trends and localized patterns exhibit partial alignment, the widespread dispersion between the two series underscores the model's limited capacity to capture the intricate dynamics of financial time series, a finding consistent with the low R² value of 0.0131. It confirms that the model explains only 0.13% of the variance in log returns, underscoring the challenges of forecasting noisy financial data.

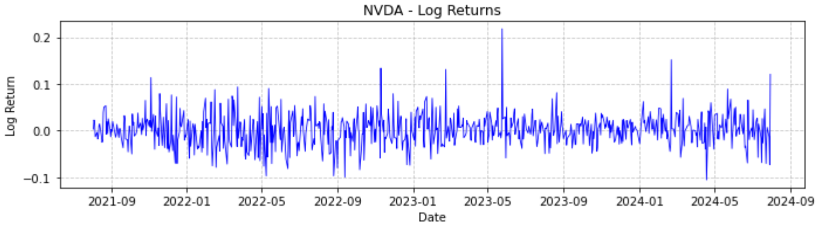
Figure 2: Actual vs. GRU-Predicted Log Returns



Source: Author's analysis

However, it provides limited evidence against the strict EMH, and it does not strongly support the persistence of nonlinear patterns in stock returns. Indeed, frequent upward and downward spikes indicate high volatility in the log returns, as shown in the plot below.

Figure 3: Nvidia Log Returns



Source: Author's analysis

The plot also reveals some short-term volatility clustering, which the GRU model attempts to capture. This pattern suggests that there are brief periods where past information provides insight into future returns, even if the predictability is only marginal. In high-frequency or high-volume trading environments, such marginal predictability can have substantial economic value. Traders and investment firms can leverage these subtle nonlinear dependencies to construct short-term algorithmic trading strategies. By systematically exploiting these inefficiencies, such a strategy may capture arbitrage opportunities that would otherwise go unnoticed in traditional linear models as long as they remain economically viable after accounting for real-world trading costs. For instance, short-term predictive signals can be integrated into algorithmic strategies that execute only when forecasted returns exceed a threshold adjusted for expected transaction costs. Although the model's explanatory power is low, such signals, when applied with other indicators, can enhance the precision of entry and exit timing in high-frequency trading environments. Moreover, the ability of GRU models to identify latent patterns in time series data can enhance risk management by improving forecasts of market volatility, enabling more responsive and adaptive hedging techniques. Additionally, incorporating such predictive signals into portfolio optimization frameworks allows for dynamic asset allocation that better reflects the evolving structure of financial markets. This underscores the practical relevance of using advanced deep learning models like GRU in modern quantitative finance, particularly in markets where even the smallest informational edge can translate into significant profits when scaled across large volumes of trades.

4. Conclusion

This study demonstrates that while GRU models can capture minimal nonlinear dependencies in stock returns, their predictive accuracy remains constrained in efficient markets, as reflected by the low R^2. These findings challenge the strict EMH but do not suggest exploitable nonlinear patterns in the long term. Nevertheless, it could be beneficial in the very short term unless it becomes economically useless after accounting for frictions.

These findings highlight the importance of integrating advanced machine learning techniques, such as GRU, into the analytical processes of investment managers. By utilizing GRU models, they can achieve a more accurate understanding of market dynamics, leading to improved forecasting accuracy and better-informed investment decisions. This approach enables practitioners to identify opportunities and risks that may not be apparent through conventional methods, ultimately enhancing portfolio performance.

To further enhance the accuracy of our predictive capabilities, future studies will investigate the development of a hybrid model that combines GRU with other machine learning techniques, as well as the exploration of the AdaMax algorithm, an extension of the Adam algorithm.

References

Bachelier, L. (1900). Théorie de la spéculation. *Annales scientifiques de l'École normale supérieure*, *17*, 21–86. https://doi.org/10.24033/asens.476

Brown, S. J. (2020). The Efficient Market Hypothesis, the Financial Analysts Journal, and the Professional Status of Investment Management. *Financial Analysts Journal*, *76*(2), 5–14. https://doi.org/10.1080/0015198X.2020.1734375

Cho, K., Merrienboer, B. van, Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation* (No. arXiv:1406.1078; Version 3). arXiv. https://doi.org/10.48550/arXiv.1406.1078

Economic prediction using neural networks: The case of IBM daily stock returns | IEEE Conference Publication | IEEE Xplore. (n.d.). https://ieeexplore.ieee.org/document/23959

Efficient Capital Markets: A Review of Theory and Empirical Work on JSTOR. (n.d.). Retrieved May 29, 2025, from https://www-jstor-org.eressources.imist.ma/stable/2325486

Fama, E. F. (1965). The Behavior of Stock-Market Prices. *The Journal of Business*, *38*(1), 34–105. https://www.jstor.org/stable/2350752

Feedforward Neural Nets as Models for Time Series Forecasting | ORSA Journal on Computing. (n.d.). Retrieved May 29, 2025, from https://pubsonline.informs.org/doi/epdf/10.1287/ijoc.5.4.374

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924

John, A., & and Latha, T. (2023). Stock market prediction based on deep hybrid RNN model and sentiment analysis. *Automatika*, *64*(4), 981–995. https://doi.org/10.1080/00051144.2023.2217602

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *CoRR*. https://www.semanticscholar.org/paper/a6cb366736791bcccc5c8639de5a8f9636bf87e8

Lo, A. W., & MacKinlay, A. C. (1988). Stock Market Prices do not Follow Random Walks: Evidence from a Simple Specification Test. *The Review of Financial Studies*, *1*(1), 41–66. https://www.jstor.org/stable/2962126

Mandelbrot, B. (1963). The Variation of Certain Speculative Prices. *The Journal of Business*, *36*(4), 394–419. https://www.jstor.org/stable/2350970

Masters, D., & Luschi, C. (2018). *Revisiting Small Batch Training for Deep Neural Networks* (No. arXiv:1804.07612). arXiv. https://doi.org/10.48550/arXiv.1804.07612

Pattanayak, A. M., Swetapadma ,Aleena, & and Sahoo, B. (2024). Exploring Different Dynamics of Recurrent Neural Network Methods for Stock Market Prediction—A Comparative Study. *Applied Artificial Intelligence*, *38*(1), 2371706. https://doi.org/10.1080/08839514.2024.2371706

Qiao, R., Chen ,Weike, & and Qiao, Y. (2022). Prediction of stock return by LSTM neural network. *Applied Artificial Intelligence*, *36*(1), 2151159. https://doi.org/10.1080/08839514.2022.2151159

Smith, L. N. (2018). *A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay* (No. arXiv:1803.09820). arXiv. https://doi.org/10.48550/arXiv.1803.09820

Thomas, A. J., Petridis, M., Walters, S. D., Gheytassi, S. M., & Morgan, R. E. (2017). Two Hidden Layers are Usually Better than One. In G. Boracchi, L. Iliadis, C. Jayne, & A. Likas (Eds.), *Engineering Applications of Neural Networks* (pp. 279–290). Springer International Publishing. https://doi.org/10.1007/978-3-319-65172-9_24

Troiano, L., Kriplani, P., & Villa, E. M. (2020). *Hands-On Deep Learning for Finance: Implement Deep Learning Techniques and Algorithms to Create Powerful Trading Strategies*. Packt Publishing, Limited.

Wang, C. (2024). Stock return prediction with multiple measures using neural network models. *Financial Innovation*, *10*(1), 72. https://doi.org/10.1186/s40854-023-00608-w

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, 159–175. https://doi.org/10.1016/S0925-2312(01)00702-0