Investigating the use of three recurrent neural networks (RNN) – Vanilla RNN, LSTM and GRU to predict Amazon’s (AMZN) stock price for the next 30-days based on Year-to-Date (YTD) stock information.

a1772276

# *Abstract*

*This report outlines the investigation undertaken to view the impact of architectural differences in Recurrent Neural Networks, specifically, Vanilla RNNS, GRU and LSTM on their efficiency in predicting stock based on past data. It was found that LSTM performed the best, as hypothesised. Since a vanilla RNN loses information overtime due to its vanishing gradient problem, this is catered to by an LSTM. Therefore, using Year-to-Date data from December 2024 for Amazon’s stock information, the LSTM, with minor mean-squared error was able to predict the stock’s price for the next give days.*

1. Introduction

Recurrent Neural Networks (RNNs) are a specialized type of artificial neural network designed to process sequential data, making them highly effective for tasks involving time series, natural language, and other ordered inputs [1][2]. Conventional neural networks have independent inputs and outputs. However, in an RNN, its unique architecture involving a specialized memory mechanism enables it to consider past information when generating predictions [3]. For every iteration, the output of the previous iteration is used as input to consider it as important information for the next calculation steps, thus allowing the neural network to consider all previous information, unlike a generic neural network where each output is distinct.

This feature of RNNs is described as hidden states, which function as memory units [6]. This state is continuously updated as the network processes each element in a sequence, allowing it to capture and utilize information from previous inputs [3]. This architecture makes RNNs particularly well-suited for tasks where context is crucial, such as language modelling or sentiment analysis [5]. To support this mechanism, the same set of weights is applied across all elements in a sequence [4][7]. This parameter sharing enables RNNs to handle sequences of varying lengths while keeping the number of parameters lower compared to traditional feedforward networks [3][6].

This special RNN architecture makes them suitable for Natural Language Processing (NLP), Speech recognition, Machine translation, Time series analysis etc.

This paper will introduce traditional RNNs and compare the architectures of three-different types of RNNs, including a Vanilla RNN, Long-Short Term Memory (LSTM) and Gates Recurrent Unit (GRU) architectures.

# Method Description

* 1. Dataset

To conduct this investigation, a dataset of time-series nature was used. The daily historical stock data from December 2023-December 2024 for Amazon (AMZN) was used as the primary dataset for this investigation. It included the features – Date, Close/Last, Open, High, Low and Volume.

This dataset was cleaned to remove the ‘$’ symbols for all price describing columns. All features were then normalised using Sci-kit-Learn’s MinMax Scaler.

This dataset was used for training three different types of architectures – Vanilla RNN, LSTM and GRU.

* 1. Vanilla Recurrent Neural Network (Vanilla- RNN)

Vanilla RNN are the simplest form of an RNN. RNNs involve a hidden state that store and provide the output of th previous iteration as an input to the next, allowing the neural network to take-in all information sequentially and synthesise a pattern between the inputs overtime. However, with increasing number of iterations, an RNN may eventually drop or forget the older input provided to it. This occurs due to a phenomenon called the “vanishing gradient problem”. This indicates that the gradient or weight term for the older inputs *vanishes* as it gets closer to 0, with increasing iterations. Naturally, this hampers an RNN’s ability to learn long-term dependencies effectively [8].

* 1. Long-Short term Memory (LSTM)

LSTM networks are a specialized type of RNN that address the limitations of standard RNNs – i.e., the loss of memory due to the vanishing gradient. They include three elements in addition to the hidden state – namely, the input gate, forget gate, and output gate. These gates orchestrate the flow of information, allowing LSTMs to retain relevant information over longer sequences while mitigating the vanishing gradient problem. This architecture enables LSTMs to excel in tasks requiring long-term memory, such as language modelling and time series prediction [9].

* 1. Gates Recurrent Unit (GRU)

Gated Recurrent Units (GRUs) are another variant of RNNs that simplify the LSTM architecture by combining the forget and input gates into a single update gate. GRUs also utilize a reset gate to determine how balance the information in memory, i.e., they help in determining which information to forget. This streamlined design reduces computational complexity while still effectively capturing dependencies in sequential data. GRUs have been shown to perform comparably to LSTMs on various tasks, particularly in scenarios with less complex data [10].

* 1. Comparison of the Architectures

While RNNs serve as foundational models for processing sequential data, they often struggle with long-term dependencies due to their simple architecture. In contrast, both LSTMs and GRUs are designed to overcome these limitations by incorporating gating mechanisms that control the flow of information. LSTMs excel in handling high-complexity sequences due to their more intricate structure but may require more computational resources. GRUs offer a simpler alternative that can outperform LSTMs on low-complexity sequences while being computationally efficient. Ultimately, the choice between these architectures depends on the specific characteristics of the task at hand and the complexity of the data being processed [9][10].

# Method Implementation

# Experiment and Analysis

# Reflection

References

1. GeeksforGeeks (2024) 'Introduction to Recurrent Neural Networks', GeeksforGeeks, 15 November. Available at: <https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/> (Accessed: 5 December 2024).
2. CAIS++ (no date) 'Recurrent Neural Networks', CAIS++. Available at: <https://caisplusplus.usc.edu/curriculum/neural-network-flavors/recurrent-neural-networks> (Accessed: 4 December 2024).
3. AlmaBetter (no date) 'Recurrent Neural Network', AlmaBetter. Available at: <https://www.almabetter.com/bytes/articles/recurrent-neural-network> (Accessed: 5 December 2024).
4. Datafloq (no date) 'Enterprise Applications of Recurrent Neural Networks', Datafloq. Available at: <https://datafloq.com/read/enterprise-applications-recurrent-neural-networks/> (Accessed: 5 December 2024).
5. Yao, J., et al. (2017) 'Visual Analysis of Recurrent Neural Networks', Hong Kong University of Science and Technology. Available at: <https://cse.hkust.edu.hk/~huamin/vast_yao_rnnvis_2017.pdf> (Accessed: 5 December 2024).
6. AWS (no date) 'What is a Recurrent Neural Network?', Amazon Web Services. Available at: <https://aws.amazon.com/what-is/recurrent-neural-network/> (Accessed: 4 December 2024).
7. Analytics Steps (no date) 'Recurrent Neural Network (RNN): Types and Applications', Analytics Steps. Available at: <https://www.analyticssteps.com/blogs/recurrent-neural-network-rnn-types-and-applications> (Accessed: 5 December 2024).
8. Asquith, N., 2020. Understanding RNNs, LSTMs and GRUs. Towards Data Science. Available at: <https://towardsdatascience.com/understanding-rnns-lstms-and-grus-ed62eb584d90?gi=0196e8b0c27d> [Accessed 4 December 2024].
9. Chen, X., Zhang, H. and Liu, Y., 2023. A comparison of LSTM and GRU networks for learning symbolic sequences. *Intelligent Computing*, pp.771-785 1.
10. Raza, M.R., Hussain, W. and Merigó, J.M., 2021. Cloud Sentiment Accuracy Comparison using RNN, LSTM and GRU. In: Innovations in Intelligent Systems and Applications Conference (ASYU). IEEE.