COMPARISON BETWEEN LONG SHORT TERM MEMORY AND DEEP NEURAL NETWORK

Submitted in partial fulfillment of the
requirements of the degree of

Master of Science
In
Mathematics

Submitted by:

Harshita Gogia

Roll No: 2022PMA6908

Under the supervision of

Dr. Amita Sharma

(Department of Mathematics)

NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY
(Formerly Netaji Subhas Institute of Technology)
AZAD HIND FAUZ MARG, SECTOR-3, DWARKA, NEW DELHI-110078

May, 2024

(Department of Mathematics)

NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

(Formerly Netaji Subhas Institute of Technology)

AZAD HIND FAUJ MARG, SECTOR-3, DWARKA, NEW DELHI-110078

CANDIDATES'S DECLARATION

I, HARSHITA GOGIA, Roll No 2022PMA6908, student of M.Sc. Mathematics, hereby declare that the project Dissertation titled "COMPARISON BETWEEN LONG SHORT TERM MEMORY AND DEEP NEURAL NETWORK" which is submitted by me to the Department of Mathematics, Netaji Subhas University of Technology, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Science is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship, or other similar title or recognition.

Place: Delhi HARSHITA GOGIA

Date: 8 May 2024

(Department of Mathematics)

NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

(Formerly Netaji Subhas Institute of Technology)

AZAD HIND FAUJ MARG, SECTOR-3, DWARKA, NEW DELHI-110078

CERTIFICATE

I hereby certify that the Project Dissertation titled "COMPARISON BETWEEN LONG SHORT TERM MEMORY AND DEEP NEURAL NETWORK" which is submitted by HARSHITA GOGIA, Roll No 2022PMA6908, Department of Mathematics, Netaji Subhas University of Technology, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Science, is a record of the project work carried out by the students under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi | DR.AMITA SHARMA

Date: 8 May 2024 | SUPERVISOR

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, Dr. Amita Sharma, for their invaluable guidance, unwavering support, and insightful feedback throughout the duration of this project. Their expertise, encouragement, and mentorship have been instrumental in shaping the direction of my research and enhancing the quality of this work.

I am also deeply thankful to the Mathematics Department at Netaji Subhas University of Technology for providing me with the resources, facilities, and conducive environment necessary for carrying out this research. The opportunity to pursue this project has been a privilege, and I am grateful for the department's continuous encouragement and support.

Furthermore, I extend my appreciation to my project group members for their collaboration, teamwork, and dedication. Their contributions and collective effort have enriched the discussions, problem-solving processes, and overall experience of working on this project together.

I acknowledge the valuable insights and knowledge gained from the numerous research papers and publications referenced in this report. The work of researchers and scholars in the field has served as a foundation for my understanding and has inspired me to explore new avenues of inquiry.

Last but not least, I would like to express my heartfelt gratitude to my family and friends for their love, encouragement, and understanding throughout this journey. Their unwavering support and encouragement have been a source of strength and motivation.

I am deeply grateful to all who have contributed in any way to the completion of this project.

Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol/Abbreviation	Description
&	And
θ	theta
α	Alpha
γ	Gamma
δ	Delta
ϵ	Epsilon
$\sum$	Summation Symbol
$\bigcirc$	Nodes
θ_0	Intercept
θ_k	Weight Paramter
ML	Machine Learning
AI	Artificial Intelligence
NN	Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
DNN	Deep Neural Network
LSTM	Long Short-Term Memory
S&P500	Standard & Poor's Index
MSE	Mean Squared Error
w.r.t	with respect to

ABSTRACT

In the realm of financial markets, accurate prediction of stock returns holds significant implications for investors, traders, and financial analysts alike. Leveraging advancements in machine learning, this project delves into the comparative analysis between Long Short-Term Memory (LSTM) networks and Deep Neural Networks (DNNs) for the prediction of stock returns, with a focus on the constituents of the S&P500 index.

The report begins with an exploration of machine learning's role in finance, highlighting its applications and challenges in the domain. It proceeds to elucidate the fundamentals of neural networks, encompassing both traditional architectures and deep learning paradigms. Special emphasis is placed on LSTM networks, renowned for their ability to model temporal dependencies and capture long-range dependencies in sequential data, making them particularly suitable for time series prediction tasks.

Employing a comprehensive dataset of historical stock prices from the S&P500 index, the project meticulously examines the predictive capabilities of LSTM networks and DNNs. The experimental setup encompasses data preprocessing, model selection, hyperparameter tuning, and rigorous evaluation using appropriate performance metrics.

Results obtained from the comparative analysis shed light on the efficacy and suitability of LSTM networks and DNNs in predicting stock returns. Insights gleaned from the findings are discussed in detail, elucidating the strengths, weaknesses, and practical implications of each approach.

This study contributes to the burgeoning field of financial machine learning by providing empirical evidence and insights into the comparative performance of LSTM networks and DNNs in stock return prediction. The findings serve as a valuable resource for practitioners, researchers, and stakeholders seeking to leverage machine learning techniques for enhanced decision-making in financial markets.

Chapter 1

INTRODUCTION

1.1 MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Machine Learning (ML):
ML serves as a subset within the broader domain of AI, dedicated to crafting algorithms and models that empower computers to learn from data autonomously, without explicit programming for each task. These algorithms adeptly discern patterns and structures within datasets, allowing systems to refine their performance as they encounter more information. Supervised, unsupervised, and reinforcement learning represent diverse ML approaches, each tailored for distinct tasks and datasets.

Artificial Intelligence (AI):
AI stands as a comprehensive discipline focused on designing systems or machines capable of executing tasks traditionally associated with human intellect [9]. This expansive field strives to replicate cognitive functions such as reasoning, problem-solving, perception, and decision-making. Beyond ML, AI encompasses a spectrum of techniques spanning natural language processing (NLP), computer vision, expert systems, and robotics, all aimed at fostering systems that evolve and enhance

their capabilities through experience and feedback.

Table 1.1: Comparison of Machine Learning and Artificial Intelligence

Machine Learning (ML)	**Artificial Intelligence (AI)**
ML focuses on creating algorithms to enable computers to learn from data and make predictions or decisions autonomously.	AI encompasses a broader range of techniques aimed at developing systems capable of tasks requiring human-like intelligence.
ML includes supervised, unsupervised, and reinforcement learning algorithms, each tailored to different task requirements and dataset characteristics.	AI involves techniques such as robotics, expert systems, and autonomous agents, with the goal of endowing machines with human-like intelligence.

1.1.1 MACHINE LEARNING IN FINANCE

In the world of finance, especially when dealing with financial time series data, accurately predicting future outcomes poses a significant challenge. This difficulty arises due to several factors, including the *inherent noise in financial data* and the widely accepted concept of *market efficiency* in its semi-strong form, as proposed by Fama in 1970.

However, despite the notion of market efficiency, there exists a multitude of anomalies in capital markets that seem to defy this principle. For instance, studies conducted by Jacobs in 2015 and Green, Hand, and Zhang in 2013 have identified over 100 such anomalies, which rely on predictive signals to outperform the market. These anomalies underscore the potential for predictive modeling in finance.

Traditional financial models often struggle to capture the intricate, *non-linear dependencies* [3] present in financial data. They typically rely on transparent relationships between predictive signals for returns (features) and future returns themselves (targets). However, these models may falter in accounting for the complexities inherent in financial markets.

Enter machine learning (ML). ML techniques offer a promising solution to the challenges of predictive modeling in finance. By harnessing sophisticated algorithms and computational capabilities, ML models can analyze extensive financial datasets, uncover patterns, and extract valuable insights that traditional models may overlook.

ML algorithms excel at identifying non-linear relationships and capturing complex *interactions within financial data*. They possess the ability to adapt and learn from experience, continually enhancing their predictive prowess. This adaptability and capacity to handle complexity make ML a valuable tool for financial analysts, traders, and investors seeking to gain a competitive edge in the markets.

In summary, machine learning stands poised to revolutionize predictive modeling in finance. By leveraging ML algorithms, financial professionals can unlock valuable insights, detect market anomalies, and make more informed decisions in the ever-evolving and intricate landscape of finance.

1.2 MODELS OF MACHINE LEARNING

Some of the most popular machine learning model are written below:

1. **Linear Regression**: A statistical method for modeling the relationship between a dependent variable and one or more independent variables by fitting a linear equation. Applications include predicting house prices based on features like size and location, and forecasting sales based on advertising spend.

2. **Logistic Regression**: A regression analysis used for predicting the probability of a binary outcome based on one or more predictor variables. It's applied in various fields, such as healthcare for predicting the likelihood of disease occurrence based on risk factors, and in marketing for predicting customer churn.

3. **Decision Trees**: A predictive modeling approach that recursively partitions the input space into regions and makes decisions based on simple rules inferred from data features. It's utilized in classification tasks, such as determining whether an email is spam or not, and in regression tasks, such as predicting a customer's income level.

4. **Random Forest** [11]: An ensemble learning method that constructs multiple decision trees during training and outputs the mode or mean prediction of the individual trees. It's employed in both classification and regression tasks, including credit scoring, fraud detection, and predicting stock prices.

5. **Support Vector Machines (SVM)**: A supervised learning algorithm that analyzes data for classification and regression analysis. It works by finding the hy-

perplane that best separates data points of different classes. Applications include text categorization, image recognition, and bioinformatics.

6. **Neural Networks (Deep Learning)**: A computational model inspired by the structure and function of the human brain, comprising interconnected nodes (neurons) organized in layers. Deep learning models have achieved state-of-the-art performance in various tasks, including image and speech recognition, natural language processing, and autonomous driving.

1.3 OBJECTIVE OF THE PROJECT

The objective of the project is to compare three different models based on their performance in predicting stock prices. The focus will be on evaluating the models' effectiveness by measuring their respective losses. Specifically, the project aims to assess the predictive capabilities of Long Short-Term Memory (LSTM), Deep Neural Networks (DNN), and a combined LSTM + DNN model. The comparison will be conducted by calculating the values of loss using Mean Squared Error (MSE) over a sample period of S&P 500 data. The project seeks to provide insights into which model performs better in terms of minimizing prediction errors and thus aiding in better decision-making regarding stock market investments.

Chapter 2

DEEP NEURAL NETWORK AND LONG SHORT TERM MEMORY

DNN represents an advancement over traditional Neural Networks, achieved by increasing the number of hidden layers. LSTM, on the other hand, stands as a refined version of RNN, a specific type of Neural Network. Let's delve into these concepts in more detail below. Before understanding the concept of deep neural network we need a prerequisite of Neural Network which is explained below:

2.1 NEURAL NETWORK

Neural networks play a pivotal role in the landscape of artificial intelligence and machine learning, drawing inspiration from the intricate architecture and functionality of the human brain's neural networks. These computational frameworks consist of interconnected nodes, referred to as neurons, meticulously organized into layers. Each neuron processes input signals through an activation function, channeling the outcomes to subsequent layers in a seamless flow of information.

At the heart of neural networks lies the feedforward neural network, orchestrating the unidirectional propagation of information—from input, traversing concealed layers, to eventual output. Deep neural networks (DNNs) build upon this foundation by incorporating multiple hidden layers, thereby unlocking the potential for intricate computations and the abstraction of complex representations.

Training neural networks involves an iterative optimization process, often guided by techniques like gradient descent, aimed at refining the weights and biases of neuron connections. This iterative optimization endeavor seeks to minimize the discrepancy between predicted and actual outputs when supplied with labeled input data. Known as supervised learning, this process continues until the network's parameters converge to a desired level of precision.

Neural networks have demonstrated remarkable success across a multitude of applications, ranging from image recognition to natural language processing and speech recognition. Their innate ability to discern intricate patterns and derive meaningful insights has solidified their position as indispensable tools across various domains such as healthcare, finance, robotics, and beyond, where addressing complex challenges necessitates sophisticated data analysis techniques. In the realm of finance, neural networks are spearheading a transformative revolution in predictive analytics and decision-making paradigms. Leveraging intricate analyses of extensive financial datasets encompassing historical market dynamics, economic indicators, and investor sentiments, neural networks excel in accurately prognosticating stock market trends. This predictive prowess not only equips traders, investors, and financial institutions with invaluable insights to capitalize on market opportunities but also enables them to proactively mitigate risks. Moreover, neural networks serve as indispensable tools in risk management, adept at identifying anomalies and potential fraudulent activities within transactions, thereby fortifying the integrity of financial institutions and safeguarding the interests of their clientele. Additionally, in portfolio management, neural networks contribute significantly to optimizing asset allocation strategies and crafting bespoke investment portfolios tailored to individual risk appetites and investment objectives. This integration of neural networks into financial applications epitomizes a paradigm shift, fostering more astute decision-making processes, heightened operational efficiency, and enhanced resilience in navigating the complexities of financial markets.

Table 2.1: Difference between Biological Neural Networks & Artificial Neural Networks [2]

Biological Neural Networks	**Artificial Neural Networks**
Biological neural networks are composed of interconnected neurons, which are specialized cells in the nervous system responsible for processing and transmitting information.	Artificial neural networks are computational models inspired by biological neural networks, designed to mimic the behavior of neurons and synapses.
The structure of biological neural networks is highly complex and adaptive, capable of learning and plasticity. Neurons in the brain communicate through electrochemical signals transmitted across synapses.	Artificial neural networks consist of interconnected nodes, or artificial neurons, organized into layers. Each neuron receives input signals, processes them, and produces an output signal.
Biological neural networks exhibit parallel processing and massive parallelism, enabling the brain to perform complex computations and tasks simultaneously.	Artificial neural networks can be trained using supervised, unsupervised, or reinforcement learning algorithms. They learn from labeled or unlabeled data, adjusting their parameters to minimize the difference between predicted and actual outputs.
Biological neural networks are highly fault-tolerant and robust, capable of adapting to changes and disturbances in the environment.	Artificial neural networks have shown remarkable success in various applications, including pattern recognition, machine translation, autonomous vehicles, and medical diagnosis.
The functioning of biological neural networks is still not fully understood, and there are many aspects of brain function that remain mysterious.	Artificial neural networks are more predictable and easier to analyze, as their behavior is governed by mathematical equations and algorithms.

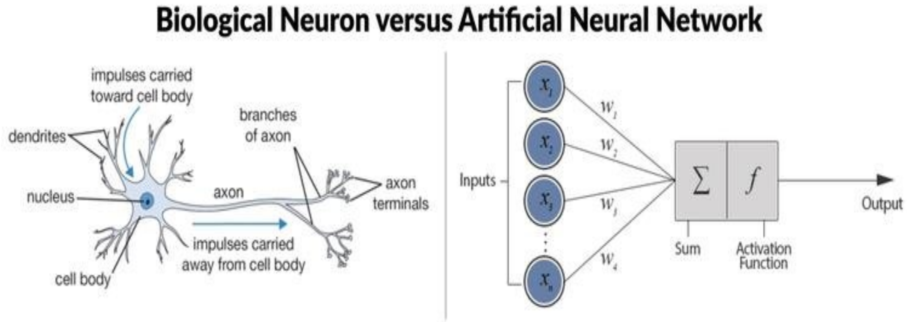

Fig 6: Biological Neutral Network Versus Artifical Neural Network

2.1.1 MODELS ON NEURAL NETWORK

Table 2.2: Neural Network Models and Their Applications in Finance

Neural Network Model	**Applications in Finance**
Deep Multilayer Perceptrons (DMLP)	Utilized for forecasting stock prices, assessing risks, and detecting fraudulent activities, leveraging their capacity to analyze intricate financial data and detect hidden patterns.
Long Short-Term Memory (LSTM)	Applied to predict time series data like stock prices, currency exchange rates, and market trends, thanks to their ability to capture prolonged dependencies within financial time series data.
Convolutional Neural Networks (CNN)	Employed in financial tasks such as document analysis, handwriting recognition, and signature verification, as well as in fraud detection systems for analyzing transactional data.
Random Forest (RF)	Used for credit scoring, loan approval, risk management, and portfolio optimization due to their flexibility in handling large datasets and understanding complex relationships among variables.
Feedforward Neural Networks (FNN)	Used in credit scoring, fraud detection, and algorithmic trading due to their ability to learn intricate patterns from historical financial data and make precise predictions.
Gated Recurrent Units (GRU)	Applied in tasks like sentiment analysis, risk assessment in loan applications, and predicting market trends based on historical data, capturing long-term dependencies in sequential financial data.
Autoencoders	Utilized for anomaly detection in financial transactions, reducing dimensions in portfolio optimization, and extracting features from high-dimensional market data.
Generative Adversarial Networks (GANs)	Employed for generating synthetic financial data, detecting financial fraud through data augmentation, and simulating market scenarios for risk analysis and stress testing.

2.1.2 ACTIVATION FUNCTIONS IN NEURAL NETWORK

Activation functions are vital components in neural networks, facilitating the network's ability to capture intricate patterns and relationships within data by introducing non-linearity. Devoid of activation functions, neural networks would be constrained to linear transformations, severely limiting their capacity to model real-world phenomena.

These functions play a pivotal role during network training, allowing for the approximation of complex functions and non-linear transformations on input data. By incorporating non-linearity, neural networks can comprehend and represent intricate relationships between inputs and outputs.

Across neural network architectures, a spectrum of activation functions exists, each tailored to specific tasks and exhibiting unique characteristics. Notable among these are [14]:

- The Sigmoid Function: Characterized by its smooth, S-shaped curve, it compresses input values between 0 and 1, often employed in binary classification tasks' output layers.

- The Rectified Linear Unit (ReLU): A simple and widely utilized activation function, producing the input value if positive and zero otherwise. Its efficiency mitigates issues like the vanishing gradient problem.

- The Leaky ReLU: Similar to ReLU, but featuring a slight slope for negative values, addressing the concern of "dying ReLU" where neurons may become inactive during training.

- The Hyperbolic Tangent (Tanh) Function: Resembling the sigmoid function but with outputs ranging from -1 to 1, frequently utilized within neural network hidden layers.

- The Softmax Function: Applied in multi-class classification tasks' output layers, it transforms raw scores into probabilities, ensuring a summation of output values equal to 1.

These functions serve as the backbone of neural network architecture, enabling them to tackle diverse tasks effectively and efficiently.

Table 2.3: Types of Activation Functions in Neural Networks

Activation Function	**Mathematical Formula**	**Domain**	**Range**
Sigmoid	$\sigma(x) = \frac{1}{1+e^{-x}}$	$(-\infty, \infty)$	$(0, 1)$
ReLU	$f(x) = \max(0, x)$	$(-\infty, \infty)$	$[0, \infty)$
Leaky ReLU	$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{otherwise} \end{cases}$	$(-\infty, \infty)$	$(-\infty, \infty)$
Tanh	$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$(-\infty, \infty)$	$(-1, 1)$
Softmax	$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$	$(-\infty, \infty)$	$(0, 1)$

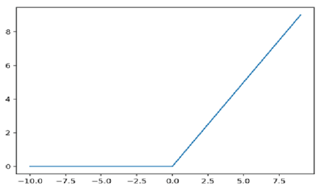

Fig 1: ReLU function

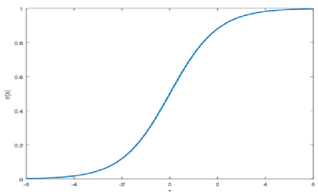

Fig 2: Sigmoid function

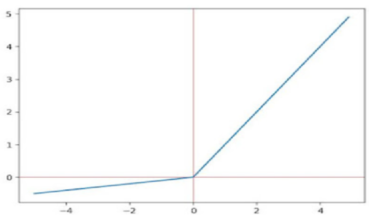

Fig 3: Leaky ReLU function

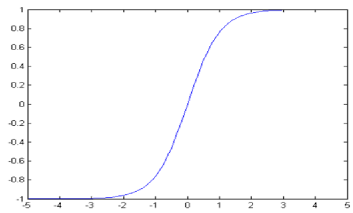

Fig 4: Tanh function

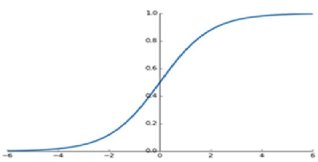

Fig 5: SoftMax function

2.1.3 COMPONENTS OF NEURAL NETWORK

The table have seven integral components of the neural network, each accompanied by a detailed description.

Table 2.4: Components of a Neural Network

Component	**Description**
Input Layer	The initial segment of the neural network where data is fed into the system. Each node within this layer signifies a distinct input feature.
Hidden Layers	Situated between the input and output layers, these layers facilitate the actual computation process. Each neuron within these layers conducts a weighted summation of inputs, applies an activation function, and transfers the outcome to the subsequent layer.
Output Layer	The conclusive layer responsible for generating the network's output, determined by the computations performed within the hidden layers.
Activation Functions	These are non-linear functions applied to the output of each neuron. They introduce non-linearity into the network, empowering it to comprehend intricate relationships within the data.
Weights and Biases	These are pivotal parameters fine-tuned by the network during training to minimize error. They govern the intensity of connections between neurons.
Loss Function	Serving as a metric to gauge the variance between predicted and actual outputs, the loss function guides the network during training by quantifying the magnitude of error.
Optimization Algorithms	These encompass a range of techniques employed to update weights and biases during the training phase, including popular methods such as gradient descent, Adam, RMSProp, among others.

2.1.4 WORKING OF NEURAL NETWORK

The functioning of a neural network revolves around two key processes: feedforward and backward propagation.

1. **Feedforward Propagation:[7]**

   - Neural network with no hidden layers:
     The left fig. shows the simple possible neural network with no hidden layer.In this input layer consist of 4 neurons, viz $\{z_1, z_2, z_3, z_4\}$.It includes intercept (θ), one w_i associated to each z_i.The output layer aggregates the weighted signal into the forecast $\theta_0 + \sum_{k=1}^{4} z_k\theta_k.$; that is;the simplest neural network is a linear regression model.

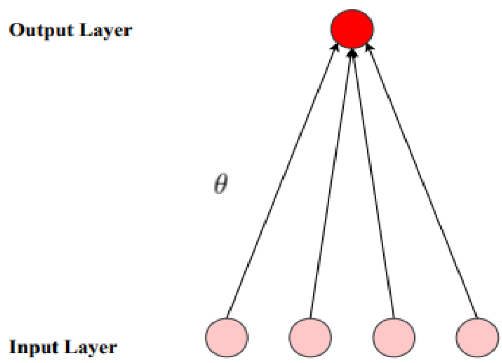

Fig 7: NN with no Hidden Layer

   - Neural Network with hidden layer:

     The model enhances predictive capabilities by introducing hidden layers between inputs and outputs. Illustrated here are four neurons in the input layer, one hidden layer containing five neurons, and one neuron in the output layer. Each neuron gathers information linearly from all input units, similar to the simpler network depicted on the left. Subsequently, an activation function f, a non-linear function, is applied to the aggregated signal before forwarding the output to the subsequent layer. For instance, the second neuron in the hidden layer transforms inputs into an output $x_2^{(1)} = f(\theta_{2,0}^{(0)} + \sum_{j=1}^{4} \theta_{2,j}^{(0)})$. Finally, the outputs from each neuron are linearly combined to produce the ultimate output forecast: $g(z;\theta) = \theta_0^{(1)} + \sum_{j=1}^{5} x_j^{(1)} \theta_{j=1}^{(1)}$.

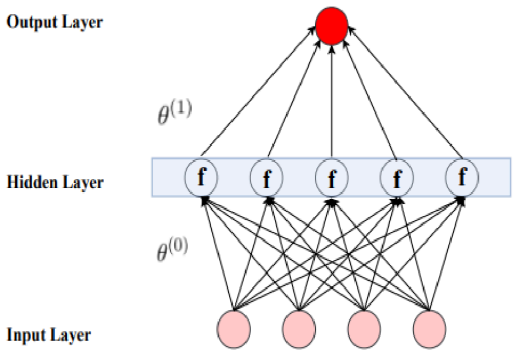

Fig 8: NN with Hidden Layer

2. **Backward Propagation:**

   Backward Propagation means propogating the error backward to adjust the weights assosiated to each neurons which results in reducing error between the actual output and predictive output.There are total 5 steps involved during the back propogation.

   - Forward Pass:

     In this step,Input data is passed through the input layer and then ,input data is multipied with their associated weights and adds a bias.The corresponding result give the output.

   - Calculating the Error:

     In this step the error is calculated by the loss function(MSE, cross-entropy for classification, etc).

     Mathematically

     $$L = \frac{1}{2}(actual\ output\ -\ predicted\ output)$$

   - Backward Pass: It involves propagating the error backward inorder to update the weights and biases.It also computes the gradiend of loss function w.r.t weights and biases using the chain rule. Let's consider a simple neural network with one hidden layer.

     Let us assume

$x =$ input

$W_h =$ weight corresponding to hidden layer

$b_h =$ bias of the hidden layer

$W_o =$ weight corresponding to the output layer

$b_o =$ bias corresponding to output layer

The forward pass of NN is represented by follows:

$$a = f(W_h \cdot x + b_h)$$
$$b = g(W_o \cdot h + b_o)$$

where f and g are the activation functions in the hidden and output layers.

During the backward pass, we need to find the gradients of the loss function L with respect to the parameters of the NN using the chain rule. The gradients are as follows:

$$\frac{\partial L}{\partial W_h}, \quad \frac{\partial L}{\partial b_h}, \quad \frac{\partial L}{\partial W_o}, \quad \frac{\partial L}{\partial b_o}$$

To find the value of above gradients, we'll use the chain rule of calculus.

(a) To find the gradient of the loss w.r.t the predicted output:

$$\frac{\partial L}{\partial b}$$

(b) To find the gradients of the loss w.r.t the parameters of the output layer using the chain rule:

$$\frac{\partial L}{\partial W_o} = \frac{\partial L}{\partial b} \cdot \frac{\partial b}{\partial W_o}$$

$$\frac{\partial L}{\partial b_o} = \frac{\partial L}{\partial b} \cdot \frac{\partial b}{\partial b_o}$$

(c) To find the gradients of the loss w.r.t the hidden layer activations:

$$\frac{\partial L}{\partial a} = \frac{\partial L}{\partial b} \cdot \frac{\partial b}{\partial a}$$

(d) To find the gradients of the loss w.r.t the parameters of the hidden layer by the chain rule:

$$\frac{\partial L}{\partial W_h} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial W_h}$$

$$\frac{\partial L}{\partial b_h} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial b_h}$$

- Updating the weights:
  After finding these gradients ,we update the weights using a variant (SGD,Adam) with φ ,the learning rate:

$$W_{1\ new} = W_{1\ old} - \varphi . \frac{\partial L}{\partial W_1}$$

$$W_{2\ new} = W_{2\ old} - \varphi . \frac{\partial L}{\partial W_2}$$

- Iterative Learning:
  In this step, the NN learn continuously by updating weights and bias and reduces the error.

2.2 DEEP (MULTILAYERED) NEURAL NETWORK(DNN)

Multilayered NN means adding multiple layers in the neural network.An advanced computational model called a deep neural network (DNN) [11] is modeled after the architecture and operation of biological neural networks. It is made up of linked layers of artificial neurons, each layer processing and transforming data in a hierarchical fashion to generate the intended output. Examining a DNN's essential components and training procedure is necessary to comprehend how it functions.

2.2.1 WORKING OF DEEP NEURAL NETWORK

1. **Input Layer:**

   When new features or raw data are added to the network, the process starts here. In this layer, every neuron represents a unique attribute of the input data. By forwarding the processed data to later layers, this layer acts as the neural network's entry point for information.

2. **Hidden Layer:**

   The hidden layers, which house the central computation, are positioned between the input and output layers. In these layers, neurons take in inputs from the layer above, add a bias term, calculate the weighted sum of these inputs, and then use an activation function to produce an output. The network is able to discover complex patterns and relationships in the data thanks to the activation function's introduction of non-linearity. The network can learn more abstract features by stacking multiple hidden layers, which enables sophisticated pattern recognition.

3. **Output Layer:**

   The output layer, which is the last layer in a neural network, is responsible for producing the network's predictions or outputs based on the data that has been processed from the hidden layers. Depending on the task, the output layer has a different number of neurons, each of which represents a distinct class or prediction. The output layer's activation function changes based on the task at hand, using linear activation for regression and softmax for classification.

4. **Weights and Biases:**

   Weights and biases are learned during training and govern the connections between neurons in neighbouring layers. These parameters control how each neuron affects the output of the network and how strong the connections are. Using optimisation techniques like backpropagation, weights and biases are iteratively changed during training to reduce the difference between expected and actual outputs.

5. **Training:**

   In order to update weights and biases, backpropagation is used in deep neural network training to propagate errors between predicted and actual outputs backward through the network. Iterative training proceeds until the network reaches a level of performance that is deemed acceptable based on metrics like accuracy and loss function values.

In summary, using interconnected layers of neurons, a deep neural network performs hierarchical computation to identify intricate patterns and relationships in data. DNNs can effectively and efficiently perform tasks like speech recognition, image recognition, and natural language processing with training and optimisation.

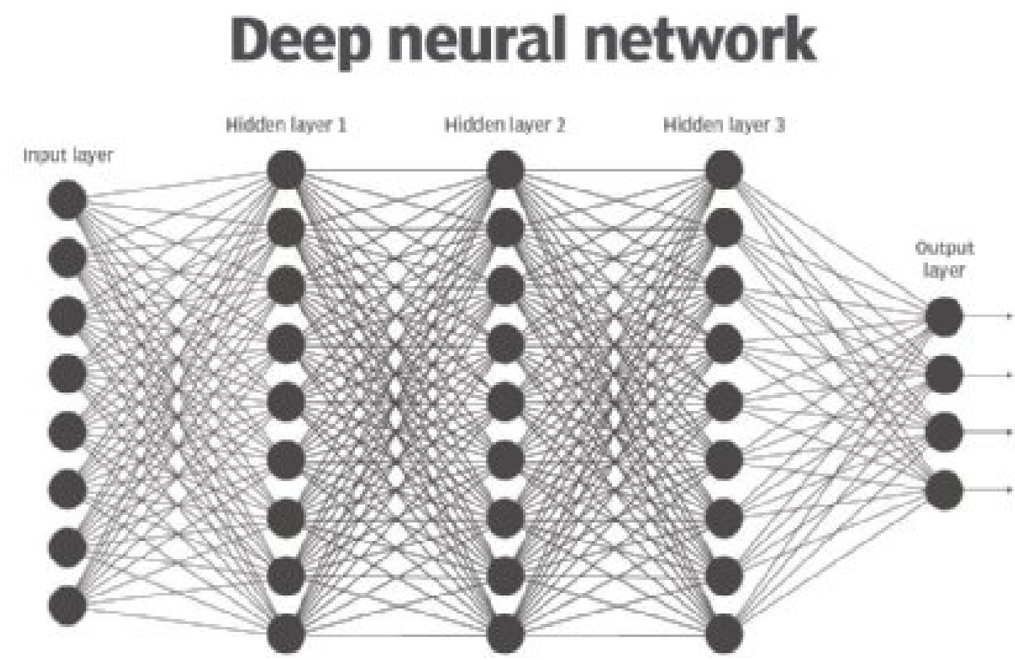

Fig 9: Deep Neural Network

2.2.2 APPLICATIONS OF DEEP NEURAL NETWORK

1. **Image Classification:**
   One of the most well-known applications of DNNs is image classification, where the network is trained to classify images into predefined categories. [10]

2. **Natural lanaguage Processing:**
   DNNs have been widely used in NLP tasks such as sentiment analysis, text classification, machine translation, and language generation. [1]

3. **Speech Recognition:**

DNNs have revolutionized speech recognition, enabling the development of systems that can transcribe spoken language accurately. [8]

4. **Healthcare:**
   DNNs have been applied in healthcare for tasks such as medical image analysis, disease diagnosis, and patient outcome prediction. [12]

2.3 RECURRENT NEURAL NETWORK(RNN)

Since LSTM stands as refined version of RNN, therefore before understanding LSTM, we need to understand RNN which is short form for *Recurrent Neural Network*

2.3.1 RECURRENT NEURAL NETWORK

A class of artificial neural networks called recurrent neural networks (RNNs) is made to process sequential data by holding onto information from previous inputs. RNNs possess an internal memory that allows them to maintain context and capture temporal dependencies within sequential data, in contrast to traditional feedforward neural networks that process each input independently.

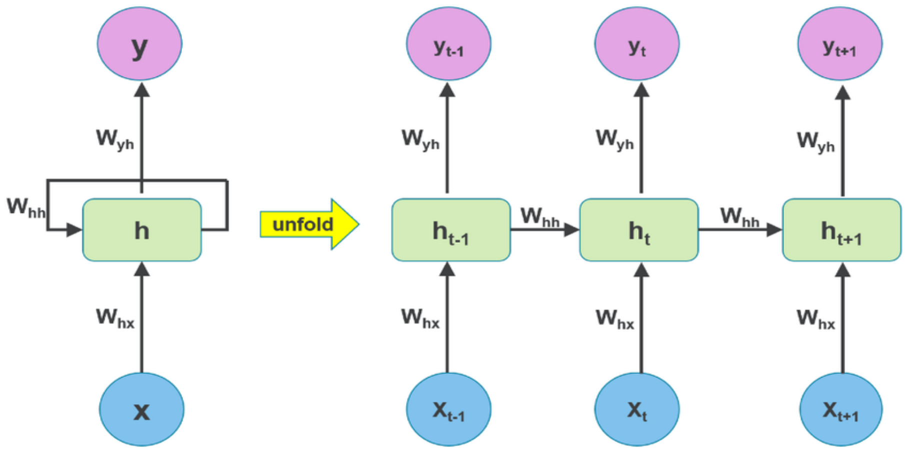

Fig 10: Unfolding of RNN

In order to update an internal hidden state that contains information from prior inputs, RNNs must iteratively process input sequences, one element at a time. Every time gradually an output is generated by combining the current input with the previous hidden state. This output is then utilised to update the hidden state for the subsequent time step. Because of this recurrent connection, *RNNs can use data from earlier inputs to inform their predictions or classifications, which are then based on the context and present input*.

2.3.2 LIMITATIONS OF RNN

RNNs are flexible enough to handle sequential data, but they are not without *limitations*. The vanishing or exploding *gradient problem*, in which the gradients used to update the network's parameters grow either very large or very small during training, is one major obstacle. This problem makes it more difficult for RNNs to identify long-range dependencies in sequential data.

Additionally, traditional RNNs struggle to retain information over long sequences due to their *fixed-size internal memory*.Let us visualize an example.Assume(in hypothetical way) that RNN has memory limit of 3 words.We have to identify a gender on the basis of sentence ”My name is Ram and I am a boy” (output which is to be identified).

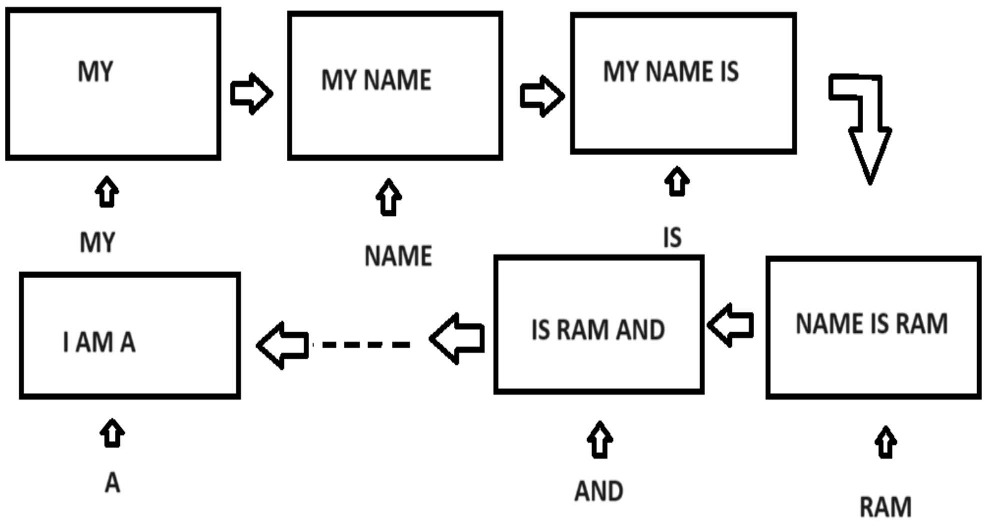

Fig 11: Fixed size internal memory

So ,at the last step,RNN have only 3 words left in its hidden state ”I am a” by which RNN won't be able to determine the gender .

As a result, they may have difficulty capturing context or dependencies that span a large number of time steps. To address these limitations, more advanced RNN architectures, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), have been developed. These variants incorporate specialized gating mechanisms that allow them to selectively update and retain information in the hidden state, making them more effective at capturing long-range dependencies and mitigating the vanishing gradient problem.

2.4 LONG SHORT TERM MEMORY(LSTM)

The recurrent neural network (RNN) architecture known as Long Short-Term Memory (LSTM) is a specialised kind that was created to overcome the shortcomings of conventional RNNs in terms of identifying long-range dependencies and preserving data across lengthy sequences. Natural language processing, speech recognition, time series prediction, and other fields have seen a rise in the use of LSTM networks.
The vanishing gradient problem, which affects traditional RNNs, causes gradients to exponentially decrease as they travel backward in time, making it more difficult for the network to learn long-term dependencies. In order to solve this difficulty, specialised memory cells and gating mechanisms were added to LSTM networks, which enable them to update and keep data selectively over a number of time steps.

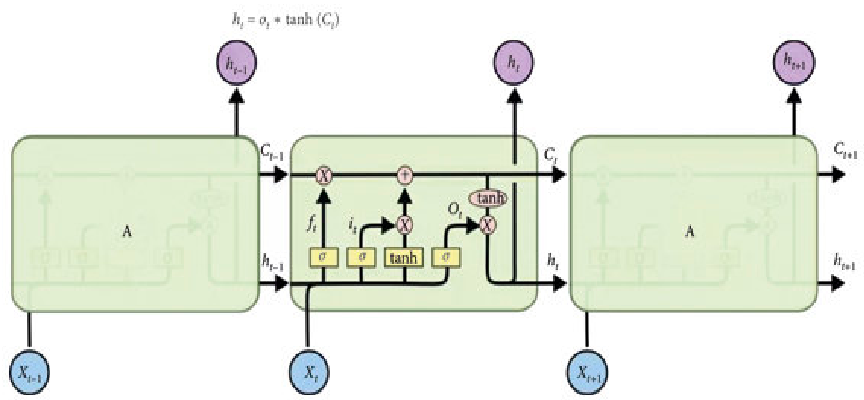

Fig 12: Long Short term Memory[4]

Working and components of the LSTM are explained below:

2.4.1 COMPONENT AND WORKING OF LSTM

Information travels across a number of gates in an LSTM network's operation, which control the information's entry and exit from the memory cell.

- *Forgot Gate*: Based on the input at hand and the previously hidden state, the forget gate decides which data from the cell state should be ignored or forgotten. Mathetmatically,

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

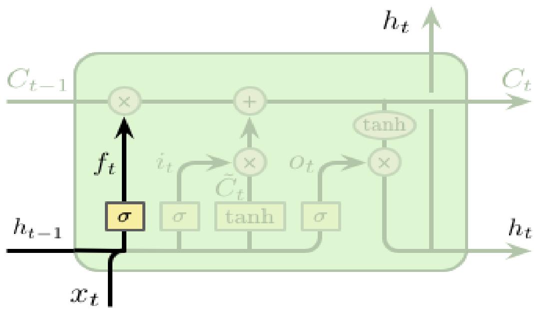

Fig 13: Forget Gate

- *Input Gate*: Using the current input and the hidden state as a basis, the input gate determines what new data needs to be added to the cell state. Mathematically[3],

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

$$\tilde{C}_t = \text{tanh}(W_C \cdot [h_{t-1}, x_t] + b_C)$$

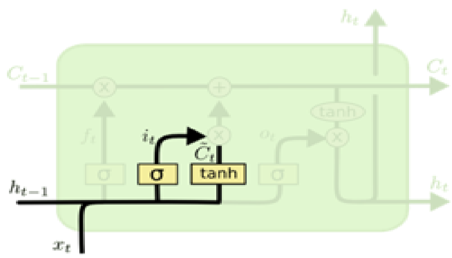

Fig 14: Input Gate

- *Cell State*: Information from the input gate (which determines what to add) and forget gate (which determines what to discard) is combined to change the cell state.

Mathematically[3],

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

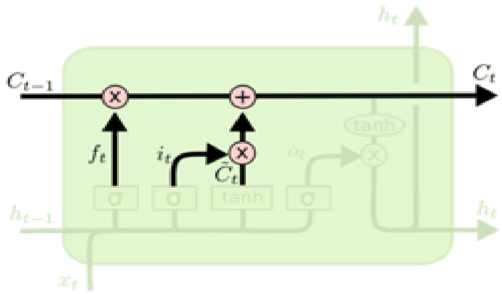

Fig 15: Cell State

- *Output Gate*: Selects which data from the current cell state should be made available to the following hidden state and output through the output gate. Mathematically[3],

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

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

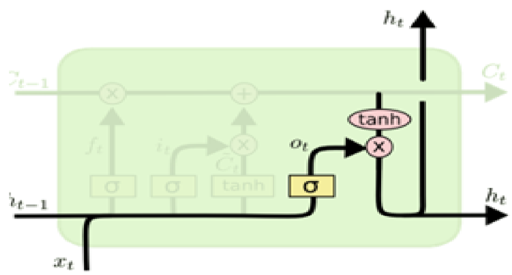

Fig 16: Output Gate

- *Hidden State*:Through a process managed by the output gate, the cell state is transformed into the hidden state, which contains the data required for predictions.
  Thanks to this technique, LSTMs are useful for tasks involving sequential data processing because they can retain and use essential information selectively over lengthy sequences.

2.4.2 APPLICATIONS OF LSTM

1. **Language Modelling and Text Generation**:

LSTM networks are widely used for language modeling and text generation tasks, enabling the generation of coherent and contextually relevant text. [6]

2. **Speech Recognition**:
   LSTM networks, particularly in conjunction with the Connectionist Temporal Classification (CTC) method, have proven effective for speech recognition tasks, facilitating accurate transcription of spoken language. [5]

3. **Machine Translation**:
   LSTM networks play a crucial role in sequence-to-sequence (Seq2Seq) models for machine translation, allowing for the translation of text between different languages with high accuracy. [15]

4. **Time Series Prediction**:
   LSTM networks are effective in modeling and predicting time series data, making them valuable in applications such as stock market forecasting, weather prediction, and energy consumption forecasting. [13]

2.5 UNDERSTANDING THE CONTRAST: DNN & LSTM

Deep Neural Network and Long Short Term Memory have some advantages as well as some disadvantages.Few of them are discussed in Table 2.5

Table 2.5: Advantages and Disadvantages of LSTM and DNN

Aspect	**LSTM**	**DNN**
Advantages	- Captures long-range dependencies - Suitable for sequential data - Effective in time series prediction	- Suitable for various tasks - Capable of processing large datasets - Can capture complex patterns
Disadvantages	- Longer training time - Requires more memory - Complexity in implementation	- Limited ability to capture long-term dependencies - Less interpretable - Vulnerable to overfitting

Following table is the basic difference between LSTM and DNN ,one should know before comparison.

Table 2.6: prerequisite difference between LSTM and DNN

Aspect	**LSTM**	**DNN**
Architecture	Recurrent Neural Network (RNN) with memory	Feedforward Neural Network
Usage	Time series prediction, sequential data	General purpose,like image processing
Long-Term Memory	Captures long-range dependencies	Limited ability of long-term dependencies
Training Time	Longer due to sequential processing	Shorter as it processes data in parallel
Interpretability	More interpretable with memory cells	Less interpretable due to hidden layers
Suitability	Suitable for sequential data	Suitable for various tasks

Table 2.7: Working Mechanisms: LSTM vs DNN

Aspect	**LSTM**	**DNN**
Processing	- Processes sequential input - Operates sequentially with memory cells - Retains memory of previous inputs	- Processes fixed-size input - Operates layer-by-layer - No memory retention
Memory	- Retains memory of past inputs - Each cell retains state information - Suitable for sequential data	- Does not retain memory - Each layer processes input independently - Suitable for static data
Usage	- Particularly effective for sequential data - Widely used in natural language processing - Commonly employed in time series prediction	- Versatile for various tasks - Effective for image, text, and numerical data - Commonly used in deep learning

Table 2.7 shows the difference between LSTM and DNN on the basis of working mechanism.

Chapter 3

ANALYSIS

3.1 UNDERLYING STUDY

This research endeavors to undertake a comprehensive comparative study focusing on three distinct neural network models: Long Short-Term Memory (LSTM), Deep Neural Networks (DNN), and a hybrid LSTM plus DNN architecture. The primary objective is to rigorously evaluate and juxtapose these models' predictive performances in the intricate domain of stock price forecasting. Through meticulous examination of their respective losses, quantified using Mean Squared Error (MSE) metrics across a designated timeframe of S&P 500 historical data, this study aims to discern nuances in predictive accuracy, shedding light on the nuanced efficacy of different neural network configurations. By offering a nuanced understanding of their relative strengths and weaknesses, this investigation endeavors to empower investors and financial analysts with invaluable insights into selecting the most adept neural network model for robust and precise stock price predictions, thereby facilitating informed decision-making in the dynamic landscape of financial markets.

3.2 METHODOLOGY

3.2.1 OVERVIEW

One of the most closely watched stock market indices globally, the Standard & Poor's 500, or S&P 500, serves as a benchmark for the general performance of the American stock market. The S&P 500 is explained in detail below:

The S&P 500, or Standard & Poor's 500, is a leading benchmark for the United States stock market, consisting of 500 of the country's largest publicly traded firms. This index represents a diverse range of industries, including technology, healthcare, finance, consumer discretionary, and industrial, providing a comprehensive picture of the US economy. The market capitalization weighting system ensures that companies with higher market values have a greater impact on the index's performance. Since its creation in 1957, the S&P 500 has become a cornerstone for investors and financial professionals, serving as a critical instrument for assessing portfolio performance and market trends.

The following graph is the S&P 500 graph from the year 1950 to 2016.

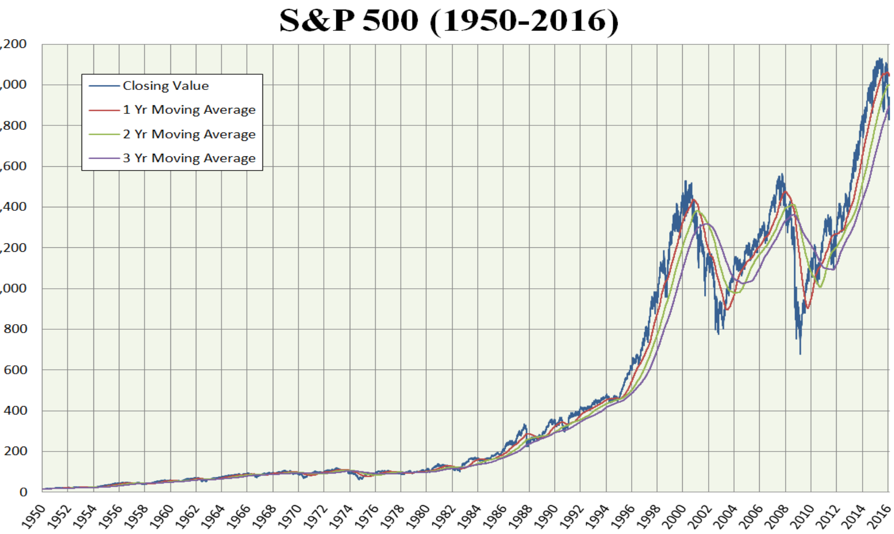

Fig 17: S&P 500 (1950-2016)

This research compares LSTM and DNN models using real-world data, specifically S&P 500 stock prices from January 2, 1973 to July 10, 2015, for companies designated Amazon, ABBOTT, AES, International Business Machine, Adobe System, Airgas, Air products & Chemical, EIDU Pont, Boston Properties. Initially, the collection had 11,094 days of data. After removing null values, the dataset was adjusted to include 4,713 days of clean data for each item. The data was then divided into two sets: training and testing, with 80% used for training and 20% for testing.

3.2.2 FILTERATION

Following the initial dataset curation, null values were found and eliminated, leaving a clean dataset ready for analysis. This filtration method assured that the subsequent study was based on accurate and comprehensive data, which included 4,713 days of observations for all assets.

3.2.3 STANDARDIZED RETURN

The standardized return of data was calculated using a predefined formula. This approach standardized the data, allowing for consistent and comparable analysis across different assets and time periods.[3]

3.2.4 TRAINING AND TESTING DATA

The dataset was split into training and testing sets, with 80% of the data used to train the algorithms and the other 20% used to test their performance. This split allowed the evaluation of model performance on before unseen data, which assisted in assessing generalization capabilities.

3.2.5 LOSS FUNCTION, ACTIVATION FUNCTION, OPTIMIZER

The loss function used in both LSTM and DNN models was mean squared error. Rectified Linear Unit (ReLU) was chosen as the activation function due to its simplicity and effectiveness in capturing nonlinear relationships in the data. Adam was used as the optimizer for model training because of its flexible learning rate capabilities and efficient convergence qualities. The model architectures differed slightly, with the LSTM having three layers and the DNN having three dense layers with different neuron units.

3.3 RESULTS

The graphs below represent the performance of three models viz. LSTM,DNN and LSTM+DNN, applied to nine different companies. Each graph illustrates the comparison between the models. However, to draw a conclusive inference on which model performed best for each specific company, we need to examine the total loss incurred by each model.

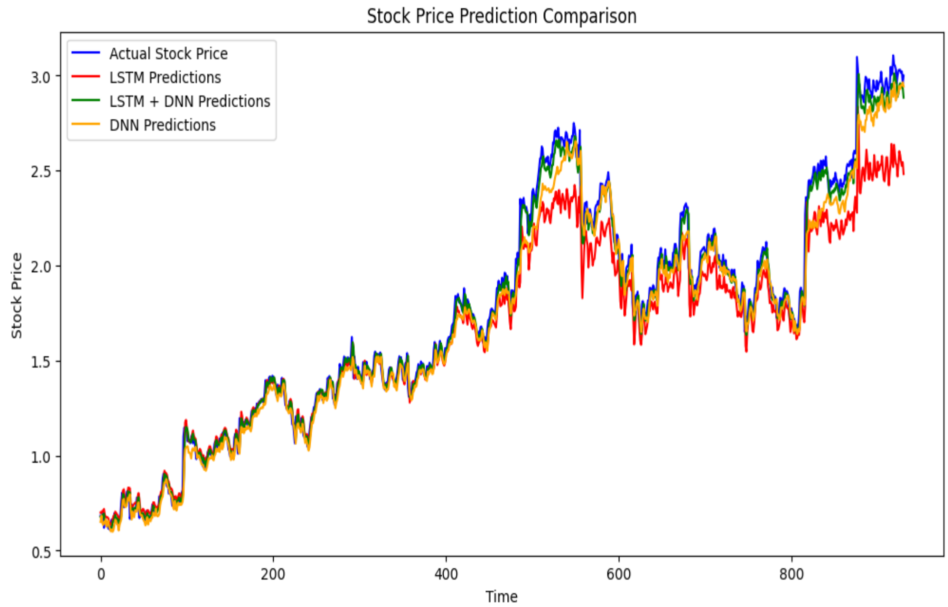

Fig 18: PREDICTION COMPARISON OF DIFFERENT MODEL ON AMAZON

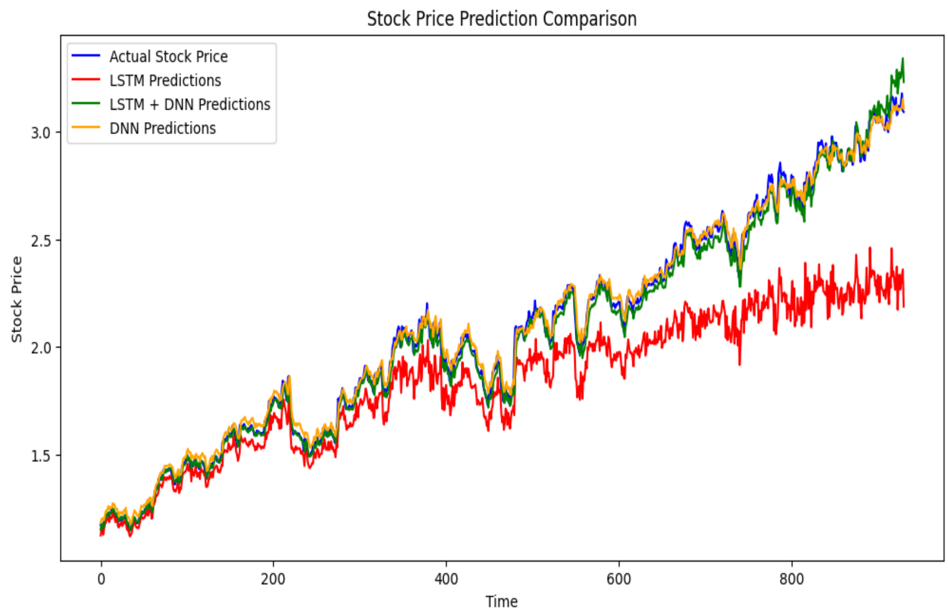

Fig 19: PREDICTION COMPARISON OF DIFFERENT MODEL ON ABBOTT

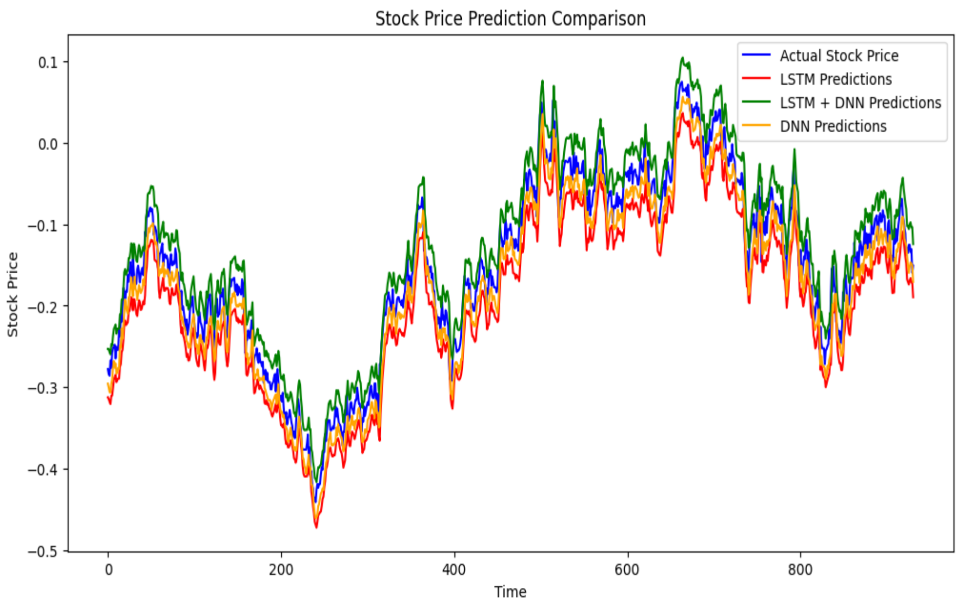

Fig 20: PREDICTION COMPARISON OF DIFFERENT MODEL ON AES

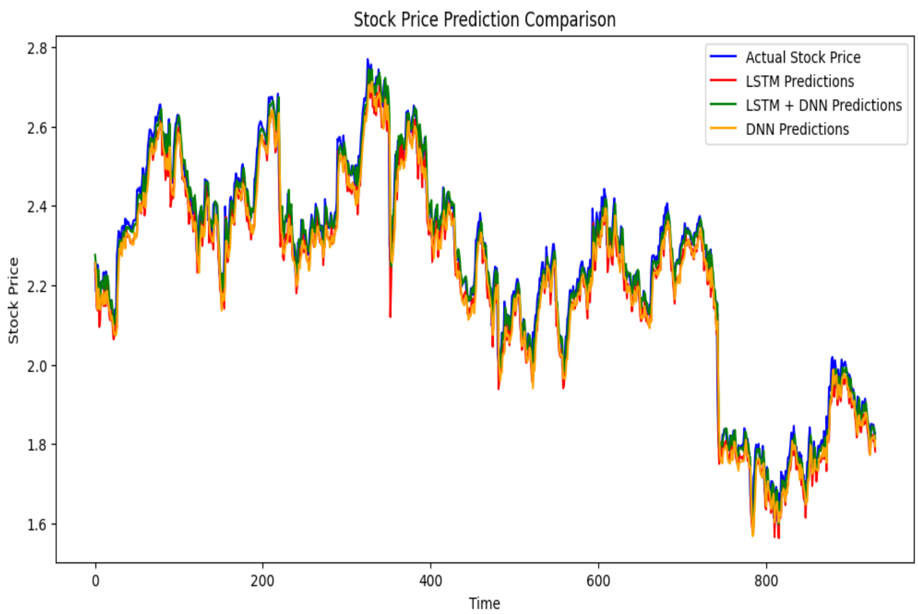

Fig 21: PREDICTION COMPARISON OF DIFFERENT MODEL ON IBM

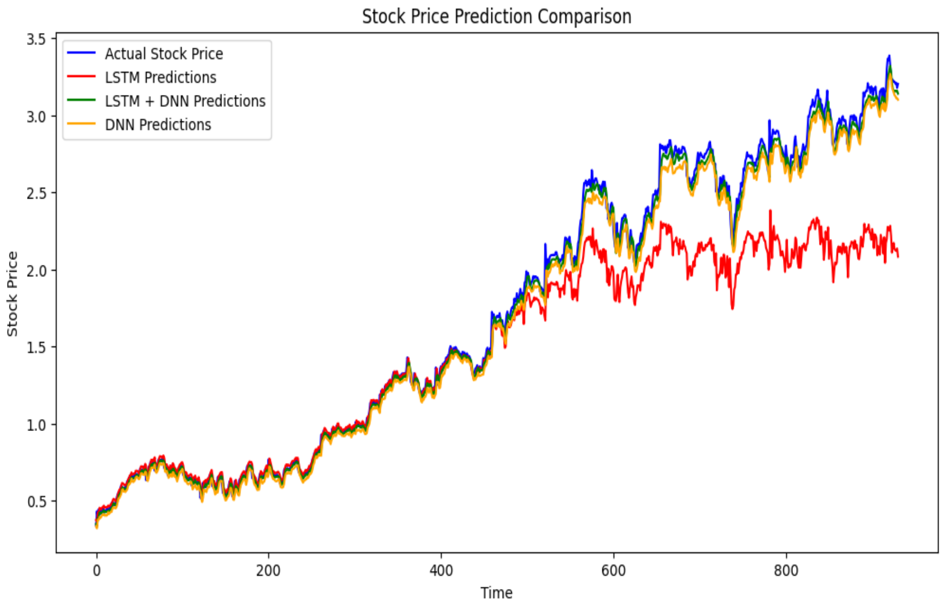

Fig 22: PREDICTION COMPARISON OF DIFFERENT MODEL ON ADOBE SYSTEM

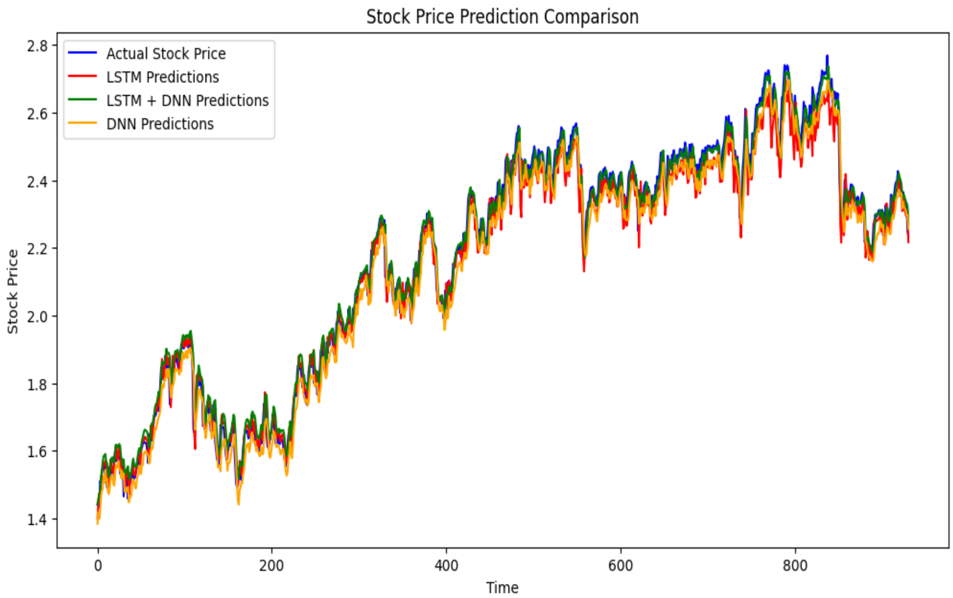

Fig 23: PREDICTION COMPARISON OF DIFFERENT MODEL ON AIRGAS

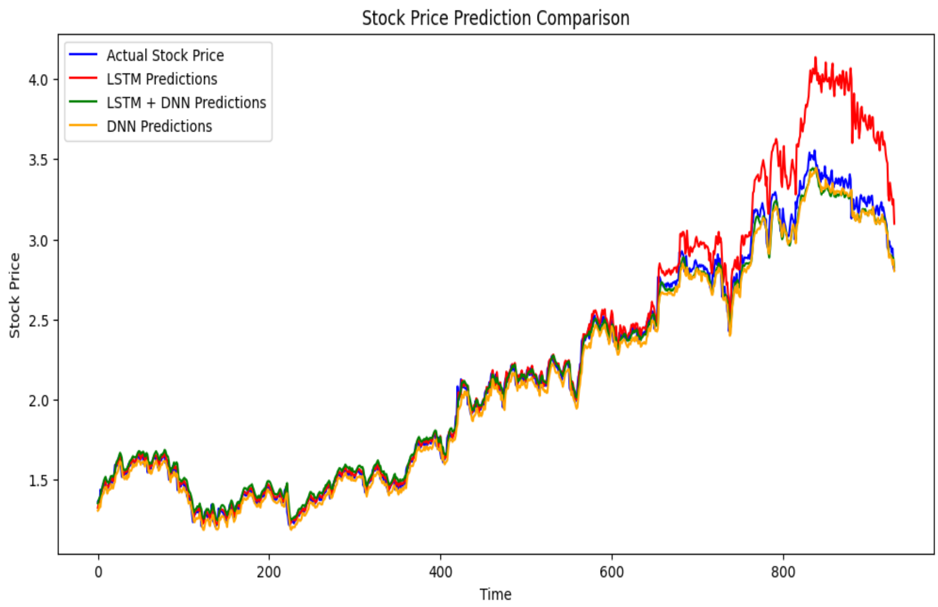

Fig 24: PREDICTION COMPARISON OF DIFFERENT MODEL ON AIR PRODUCTS AND CHEMICALS

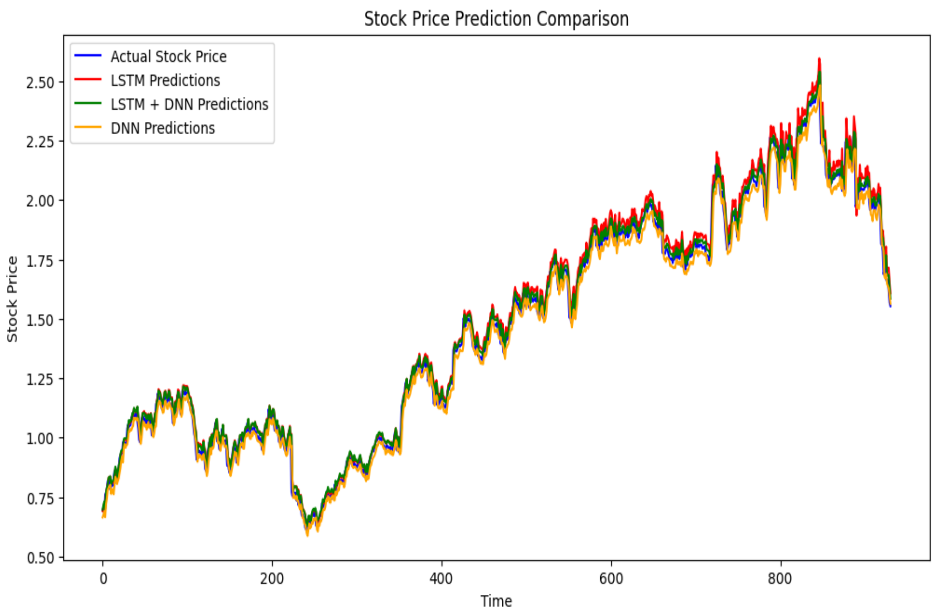

Fig 25: PREDICTION COMPARISON OF DIFFERENT MODEL ON EIDY PONT

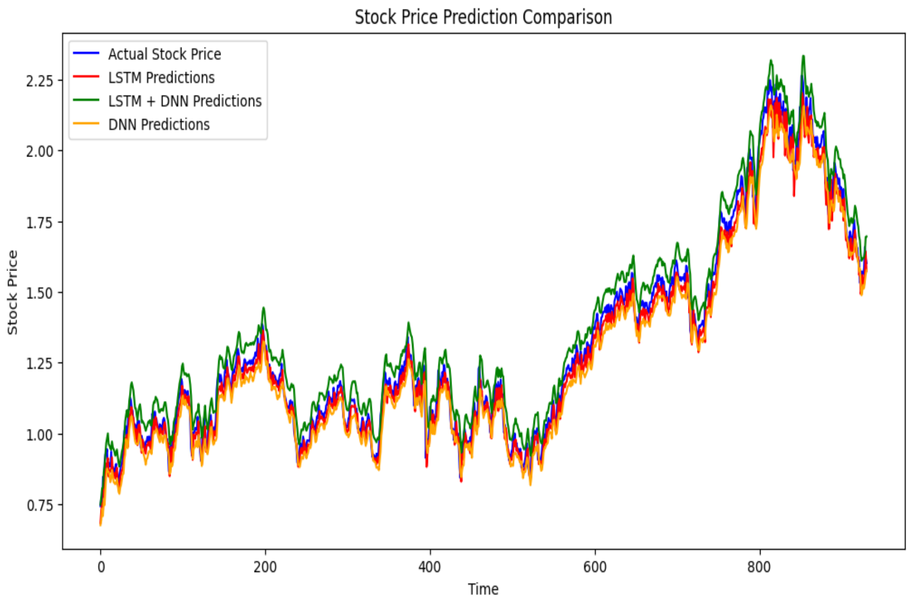

Fig 26: PREDICTION COMPARISON OF DIFFERENT MODEL ON BOSTON PROPERTIES

Table 3.1: Total Loss by DNN,LSTM & both LSTM and DNN

COMPANIES	DNN	LSTM	LSTM & DNN
AMAZON	0.000423	*0.00150*	**0.000242**
ABBOTT	**0.000125**	*0.00042*	0.000167
AES	*0.000069*	0.000036	**0.0000077**
IBM	*0.000380*	0.00023	**0.000144**
ADOBE SYSTEM	0.0001600	*0.00030*	**0.0001425**
AIRGAS	0.001161	*0.00340*	**0.000146**
AIR PRDTS & CHMCLS	0.0003066	*0.000302*	**0.000130**
EIDY PONT	*0.0001287*	0.0001261	**0.000109**
BOSTON PROPERTIES	*0.0001742*	**0.0001198**	0.000233

In table 3.1, the total loss for three different models, namely LSTM, DNN, and LSTM+DNN, has been computed. Row in the uppermost section corresponds to a specific model, while column in the leftmost section represents a company. The values within the table denote the total loss incurred by each model for predicting stock prices of the respective companies. In this context, a lower loss indicates superior performance of the model for the corresponding company. Therefore, the numbers highlighted in bold signify lower losses, suggesting that the model performs well for those companies. Conversely, numbers in italic represent higher losses, indicating suboptimal performance of the model for the corresponding companies. Based on the analysis conducted on 9 assets from the S&P 500, employing both LSTM and DNN together has yielded superior performance compared to using either LSTM or DNN alone. Specifically, when evaluating the minimum loss across the 9 assets, the combination of LSTM and DNN exhibited the lowest loss in 7 assets. Upon comparing the individual performance of LSTM and DNN, it is evident that while there are marginal differences in loss for some assets, a clear trend emerges. Out of the 9 assets, DNN demonstrated lower loss in 5 assets, whereas LSTM outperformed in the remaining 4 assets.

CONCLUSION AND FUTURE DIRECTIONS

In conclusion, the collaborative utilization of LSTM and DNN models appears to enhance predictive capabilities, showcasing superior performance across the majority of assets compared to using either model individually. This suggests that leveraging the complementary strengths of both LSTM and DNN can be advantageous in asset prediction tasks within the S&P 500.

In future endeavors, my aim is to conduct a comprehensive comparison across a larger number of assets within the financial market. I intend to develop a robust machine learning model capable of predicting tomorrow's stock price using historical data in conjunction with relevant news headlines. This forthcoming analysis will involve an expanded dataset encompassing a diverse array of assets, enabling a more comprehensive evaluation of predictive models. Furthermore, I plan to explore advanced techniques and algorithms to enhance the accuracy and reliability of the predictive framework. By integrating both quantitative historical data and qualitative textual information from news headlines, the proposed model seeks to provide deeper insights into the dynamics driving asset price movements. This future direction aims to contribute to the advancement of predictive analytics within the financial domain, facilitating more informed decision-making for investors and market participants.

Bibliography

[1] Vaswani Ashish. Attention is all you need. *Advances in neural information processing systems*, 30:I, 2017.

[2] OS Eluyode and Dipo Theophilus Akomolafe. Comparative study of biological and artificial neural networks. *European Journal of Applied Engineering and Scientific Research*, 2(1):36–46, 2013.

[3] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European journal of operational research*, 270(2):654–669, 2018.

[4] Achyut Ghosh, Soumik Bose, Giridhar Maji, Narayan Debnath, and Soumya Sen. Stock price prediction using lstm on indian share market. In *Proceedings of 32nd international conference on*, volume 63, pages 101–110, 2019.

[5] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[6] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.

[7] Shihao Gu, Bryan T Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *Chicago Booth Research Paper*, (18-04):2018–09, 2019.

[8] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al.

Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[9] J Matthew Helm, Andrew M Swiergosz, Heather S Haeberle, Jaret M Karnuta, Jonathan L Schaffer, Viktor E Krebs, Andrew I Spitzer, and Prem N Ramkumar. Machine learning and artificial intelligence: definitions, applications, and future directions. *Current reviews in musculoskeletal medicine*, 13:69–76, 2020.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[11] Yilin Ma, Ruizhu Han, and Weizhong Wang. Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, 165:113973, 2021.

[12] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.

[13] Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 2014.

[14] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316, 2017.

[15] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.