# ABSTRACT

Contents

[ABSTRACT 3](#_Toc106083450)

[Chapter 1: INTRODUCTION 3](#_Toc106083451)

[1.1 Background of Study 3](#_Toc106083452)

[1.1.1 Foreign Exchange (FOREX) 3](#_Toc106083453)

[1.1.2 Deep Learning 3](#_Toc106083454)

[1.1.2.1 Deep Learning vs. Machine Learning 4](#_Toc106083455)

[1.1.2.2 How Deep Learning Works 4](#_Toc106083456)

[1.1.2.3 Deep Learning Applications 5](#_Toc106083457)

[1.1.2.4 Deep Learning Hardware Requirements 6](#_Toc106083458)

[1.1.3 Convolutional Neural Networks (CNN) 6](#_Toc106083459)

[1.1.3.1 Building Blocks of CNN Architecture 6](#_Toc106083460)

[1.1.3.1.1 Convolution Layer 6](#_Toc106083461)

[1.1.3.1.1.1 Convolution 7](#_Toc106083462)

[1.1.3.1.1.2 Nonlinear Activation Function 7](#_Toc106083463)

[1.1.3.1.2 Pooling Layer 8](#_Toc106083464)

[1.1.3.1.2.1 Max Pooling 8](#_Toc106083465)

[1.1.3.1.2.2 Global Average Pooling 8](#_Toc106083466)

[1.1.3.1.3 Fully Connected Layer 8](#_Toc106083467)

[1.1.3.1.4 Last Layer Activation Function 8](#_Toc106083468)

[1.1.3.1.5 Dropout 8](#_Toc106083469)

[1.1.3.2 Training a network 9](#_Toc106083470)

[1.1.3.2.1 Loss Function 9](#_Toc106083471)

[1.1.3.2.2 Gradient Descent 9](#_Toc106083472)

[1.2 Problem statement. 9](#_Toc106083473)

[1.3 Aim and objectives 11](#_Toc106083474)

[1.3.1 Aim 11](#_Toc106083475)

[1.3.2 Objectives 11](#_Toc106083476)

[1.4 Scope and limitations 11](#_Toc106083477)

[1.4.1 Scope 11](#_Toc106083478)

[1.4.2 Limitations 12](#_Toc106083479)

[1.5 Summary and report 12](#_Toc106083480)

[Chapter 2: LITERATURE REVIEW 12](#_Toc106083481)

[2.1 Existing/Similar Systems 12](#_Toc106083482)

[2.1.1 Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach [11] 12](#_Toc106083483)

[2.1.2 Financial Time-series Data Analysis using Deep Convolutional Neural Networks [12] 13](#_Toc106083484)

[2.1.3 CNN-based stock market prediction using a diverse set of variables [9] 13](#_Toc106083485)

[2.1.4 Predicting Financial Prices of Stock Market using Recurrent Convolutional Neural Networks [13] 14](#_Toc106083486)

[2.1.5 Financial Markets Prediction with Deep Learning [14] 15](#_Toc106083487)

[Chapter 3: System Methodology 16](#_Toc106083488)

[3.1 Dataset 16](#_Toc106083489)

[3.2 Architecture 20](#_Toc106083490)

[3.2.1 Pre-processing of Data 20](#_Toc106083491)

[3.2.2 Data Generator 21](#_Toc106083492)

[3.3 Input Layer 21](#_Toc106083493)

[3.4 Convolutional Layer 21](#_Toc106083494)

[Chapter 4: System implementation and Testing 21](#_Toc106083495)

[4.1 System implementation 21](#_Toc106083496)

[4.1.1 Interface design 21](#_Toc106083497)

[4.2 System testing 21](#_Toc106083498)

[4.2.1 Unit testing 21](#_Toc106083499)

[4.2.2 Functional testing 21](#_Toc106083500)

[4.2.3 Integration testing 21](#_Toc106083501)

[4.3 System Evaluation 22](#_Toc106083502)

[4.3.1 (Remote) Expert Testing 22](#_Toc106083503)

[4.3.2 (Remote) User testing 22](#_Toc106083504)

[4.3.3 Results 22](#_Toc106083505)

[4.3.3.1 Expert testing 22](#_Toc106083506)

[4.3.3.1.1 Changes made to the website 22](#_Toc106083507)

[4.3.3.2 User analysis 22](#_Toc106083508)

[4.3.3.3 Questionnaire analysis 22](#_Toc106083509)

[Chapter 5: Conclusion 22](#_Toc106083510)

[5.1 Summary 22](#_Toc106083511)

[5.2 Future Improvements 22](#_Toc106083512)

[REFERENCE 22](#_Toc106083513)

[APPENDIX 22](#_Toc106083514)

# Chapter 1: INTRODUCTION

## Background of Study

In this section, we cover the convolutional neural network, which is the core component of our system, the foreign exchange(forex) market, and deep learning.

### 1.1.1 Foreign Exchange (FOREX)

The conversion of one country's currency into another is known as foreign exchange, FX, or forex. The value of a country's currency in a free economy is determined by supply and demand. In other words, the value of a currency can be tied to that of another country, such as the U.S. dollar (USD), or a basket of currencies. In our case, we use the currency pairs tied to the value of the USD. The government of a country can also set the value of its currency. Many countries on the other hand freely float their currencies against those of other countries, causing them to fluctuate constantly, for example, the Saudi Arabian Riyal (SAR) floats its currency against the USD. [1]

The global economy's beating heart is the forex markets. This market is where the currencies of the world are traded. This is especially important because currencies allow us to purchase goods and services locally and across borders. International currency exchange is necessary for foreign trade and business to occur. [2]

For example, if you live in the United States and wish to purchase cheese from France, you, or the firm from which you purchase the cheese must pay the French in euros (EUR). This means that the importer in the United States would have to convert the same amount of dollars (USD) into euros. The same is true when it comes to traveling. Because euros are not accepted in Egypt, a French tourist visiting the pyramids will be unable to pay in euros. Based on the current exchange rate, the EUR held by the tourist must be converted to the local currency, in this case, the Egyptian pound (EGP). [2]

The marketplace for international markets is decentralized for foreign exchange, which is a unique feature. Rather than trading on a single centralized exchange, currency trading is done electronically over the counter (OTC), which means that all transactions take place over computer networks among traders all over the world. The market is open 24 hours a day, five days a week, and currencies are traded in every time zone in Frankfurt, Hong Kong, London, New York, Paris, Singapore, Sydney, Tokyo, and Zurich, among other important financial locations. This means that when the trading day in the United States finishes, the forex market in Tokyo and Hong Kong restarts. As a result, the currency market can be very lively at any time, with price quotes continuously fluctuating. [2]

Institutional firms and huge banks dominated the currency market in the past, acting on behalf of clients. However, in recent years, the market has become more retail-oriented, and traders and investors with a wide range of holding sizes have begun to participate; anyone with an internet connection and a bank account can participate.

### 1.1.2 Deep Learning

Deep learning can be thought of as a subset of Machine learning. It is a field that is focused on computer algorithms learning and developing on its own. Deep learning is a variation of artificial neural networks, which are supposed to mimic how humans think and learn. Neural networks were previously limited in complexity due to computational power constraints. Larger, more powerful neural networks have been enabled by advances in Big Data analytics, allowing computers to monitor, understand, and react to complicated events faster than people. Image categorization, language translation, and speech recognition have all benefited from deep learning. It can solve any pattern recognition problem without the need for human intervention. [3]

It is a three or more-layer neural network. These neural networks aim to imitate the activity of the human brain by allowing it to learn from enormous amounts of data, albeit they fall far short of its capabilities. While a single-layer neural network may produce approximate forecasts, extra hidden layers could help to optimize or improve accuracy. [4]

Numerous AI apps and services rely on deep learning to improve automation by executing analytical and physical activities without the need for human participation. Daily goods and services (including digital assistants, voice-enabled TV remotes, and credit card fraud detection), as well as upcoming innovations, use deep learning technology (such as self-driving cars). [4]

#### 1.1.2.1 Deep Learning vs. Machine Learning

Deep learning differs from traditional machine learning in the types of data it uses and the learning algorithms it employs. To create predictions, machine learning algorithms use structured, labeled data, which means that certain features are defined from the model's input data and grouped into tables. This is not to say it never uses unstructured data; it just means that if it does, it usually goes through some pre-processing to convert it to a structured format. [4]

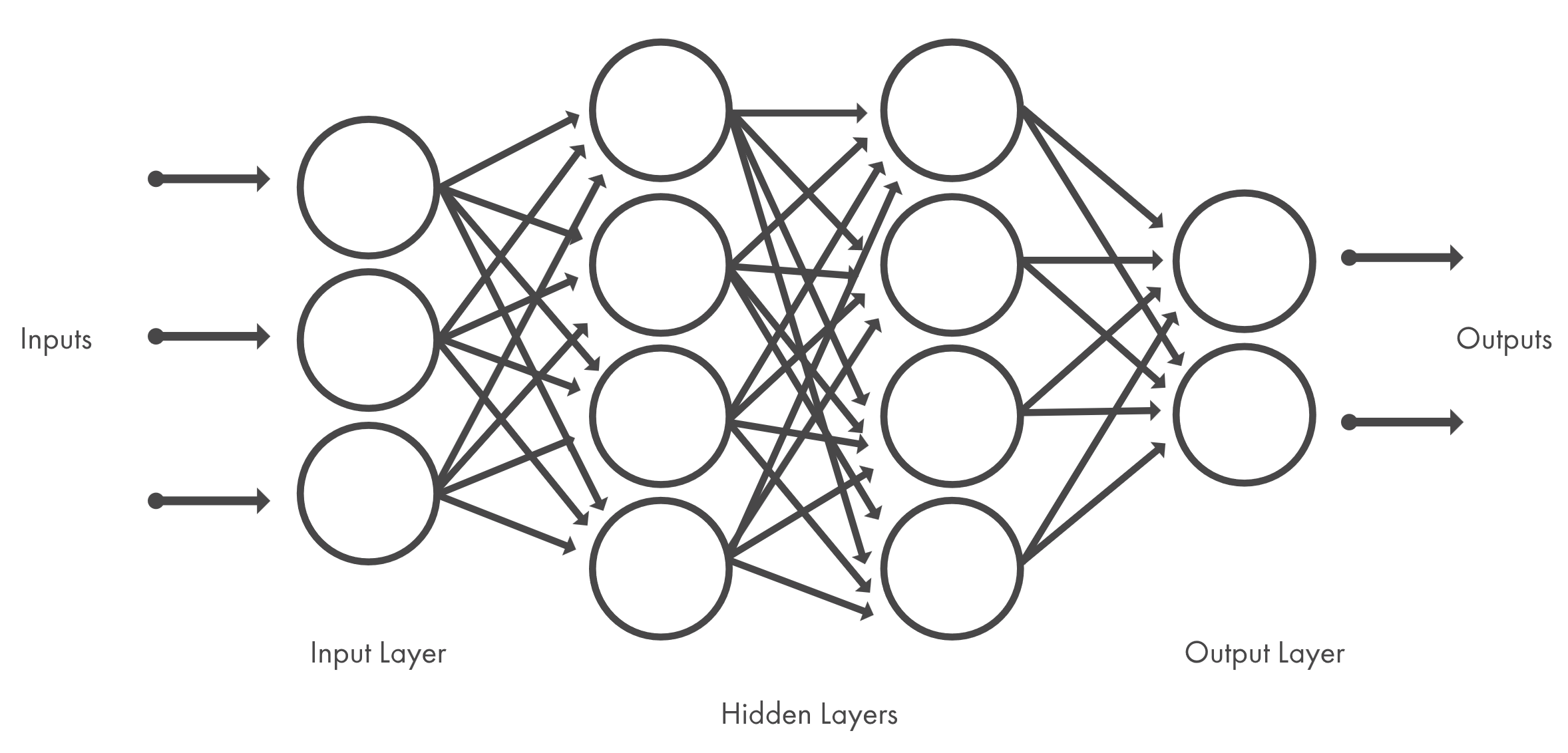
Deep learning removes some of the data pre-processing that machine learning entails. These algorithms can absorb and interpret unstructured data such as text and photos, as well as automate feature extraction, which reduces the need for human specialists. Let us imagine we had a collection of images of various pets that we intended to categorize by "cat", "dog", or "hamster,". Deep learning methods can figure out which characteristics (such as ears) are most essential in distinguishing one animal from another. This feature hierarchy is manually created by a human specialist in machine learning. [4]

The deep learning algorithm then changes and fits itself for accuracy via gradient descent and backpropagation, allowing it to generate more precise predictions about a fresh snapshot of an animal. [4]

Diverse types of learning, such as supervised learning, unsupervised learning, and reinforcement learning, are possible with machine learning and deep learning models. Supervised learning categorizes or predicts using labeled datasets; this involves some type of human interaction to appropriately classify input data. Unsupervised learning, on the other hand, does not require labeled datasets; instead, it discovers patterns in the data and clusters them according to any differentiating criteria. Reinforcement learning is a learning process in which a model improves its accuracy for completing an action in each scenario and received feedback to increase the reward. [4]

#### 1.1.2.2 How Deep Learning Works

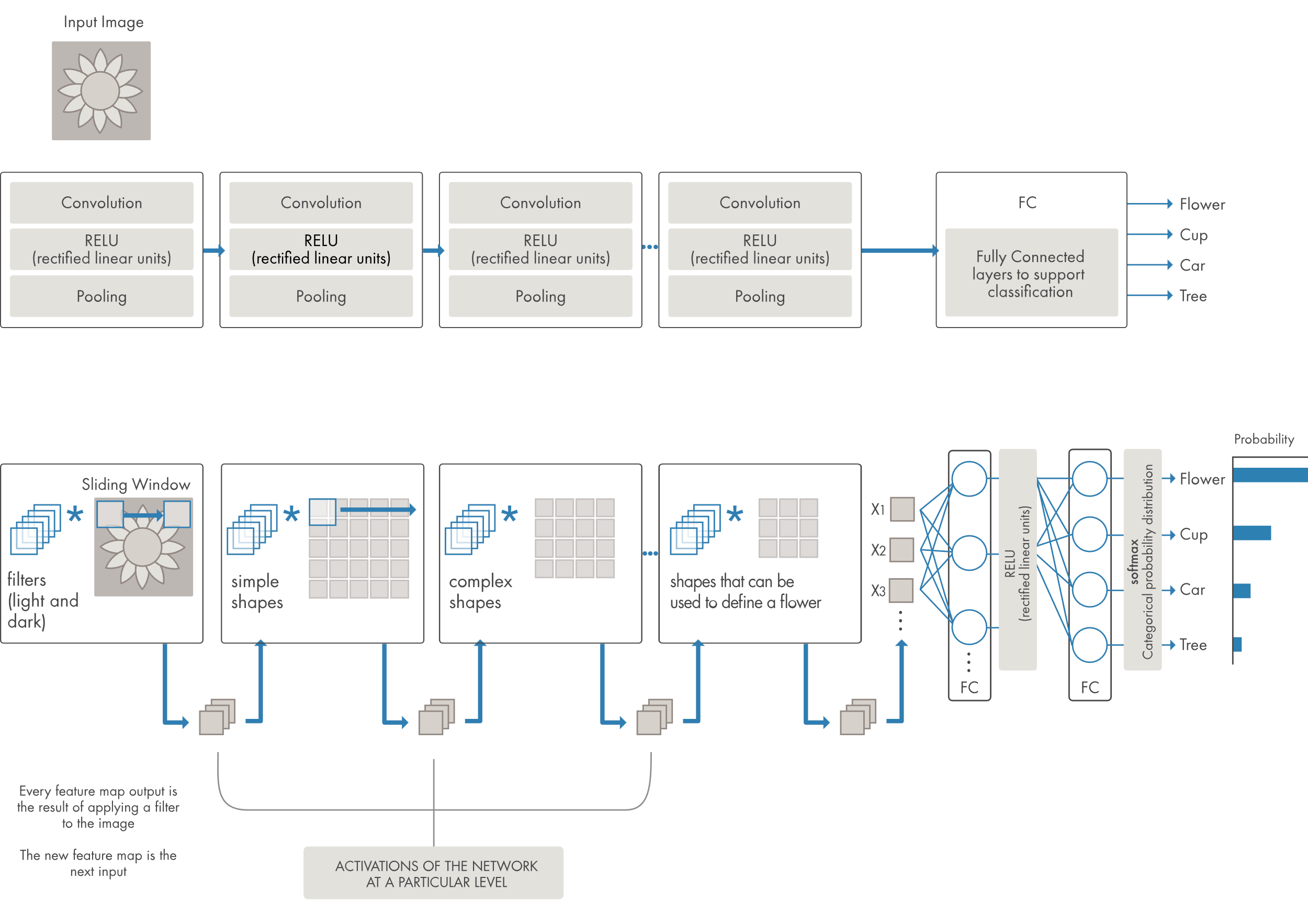
Deep learning models are sometimes referred to as deep neural networks because most deep learning approaches use neural network designs. The number of hidden layers in a neural network is commonly referred to as "deep." Deep neural networks can have up to 150 hidden layers, whereas traditional neural networks only have 2-3. Large sets of labeled data and neural network topologies that learn features directly from the data without the requirement for manual feature extraction are used to train deep learning models. [5]



Link: <https://www.mathworks.com/discovery/deep-learning/_jcr_content/mainParsys/band_2123350969_copy_1983242569/mainParsys/columns_1635259577/1/image_2128876021_cop_1731669336.adapt.full.medium.svg/1653027391134.svg>

Figure 1: Neural Network Layout

Convolutional neural networks are one of the most popular types of deep neural networks (CNN or ConvNet). A CNN uses 2D convolutional layers to combine learned features with input data, making it well suited to processing 2D data like photos. CNNs do away with the requirement for manual feature extraction, so you will not have to figure out what features are used to classify images. The CNN operates by extracting features from photos directly. The relevant features are not pre-trained; instead, they are discovered when the network trains on a set of images. Deep learning models are particularly accurate for computer vision applications such as object categorization thanks to this automated feature extraction. [5]



Link: <https://www.mathworks.com/discovery/deep-learning/_jcr_content/mainParsys/band_2123350969_copy_1983242569/mainParsys/columns_1635259577/1/image_2128876021_cop.adapt.full.medium.svg/1653027391221.svg>

Figure 2: Convolutional Neural Network Layout

#### 1.1.2.3 Deep Learning Applications

Deep learning applications in the real world are all around us, but they are usually so effectively integrated into products and services that users aren't aware of the intricate data processing going on in the background. Some of the real-world applications of deep learning include law enforcement, financial services, customer service, healthcare, automated driving, aerospace and defense, medical research, industrial automation, and electronics. [4] [5]

#### 1.1.2.4 Deep Learning Hardware Requirements

Deep learning demands a massive amount of processing power. Graphic processing units (GPUs) with high performance are desirable because they can manage a big volume of operations in several cores with enough memory. Managing numerous GPUs on-premises, on the other hand, can place a significant strain on system resources and be extremely costly to expand. [4]

### 1.1.3 Convolutional Neural Networks (CNN)

A convolutional neural network is a type of deep neural network that is often used to evaluate visual imagery in deep learning. When we think about neural networks, we usually think of matrix multiplications, but this is not the case with CNN. It employs a technique known as Convolution. Convolution is a mathematical procedure that produces a third function that expresses how the shape of one is affected by the other. However, we do not need to grasp what a CNN is or how it works to understand what it is. In the end, CNN's job is to compress the images into a format that is easier to manage while preserving elements that are important for a decent prediction. CNN models excel in recognizing patterns in pictures, such as lines. This feature should also be able to recognize trading chart trends. CNN can also find linkages between images that humans cannot see, and the topology of neural networks can help detect complicated interactions between features. [6]

Artificial Intelligence has made considerable progress in closing the gap between human and computer capabilities. Researchers and hobbyists alike work on a variety of facets of the subject to achieve incredible results. The field of computer vision is one of several such disciplines. The goal of this field is to enable machines to see and perceive the world in the same way that humans do, and to use that knowledge for a variety of tasks such as image and video recognition, image analysis and classification, media recreation, recommendation systems, or natural language processing. Advancements in Computer Vision using Deep Learning have been built and developed through time, mostly using a single algorithm – the Convolutional Neural Network. [7]

The AI system, dubbed AlexNet (after its designer, Alex Krizhevsky), took first place in the 2012 ImageNet computer vision challenge with an incredible 85 percent accuracy. On the test, the runner-up received a respectable 74 percent. Convolutional Neural Networks were at the heart of AlexNet. CNNs have become an integral feature of many computer vision applications over the years and are thus covered in any online computer vision course. [6]

#### 1.1.3.1 Building Blocks of CNN Architecture

Convolution layers, pooling layers, and fully connected layers are among the building components of the CNN architecture. A typical architecture comprises one or more completely linked layers followed by a stack of many convolution layers and a pooling layer. Forward propagation refers to the process of transforming input data into output data using these layers. Although the convolution and pooling methods described are for 2D-CNNs, comparable operations can be done for 3D-CNNs as well. [8]

##### 1.1.3.1.1 Convolution Layer

A convolution layer is a crucial element of said CNN architecture which facilitates feature extraction using a blend of linear and nonlinear processes, such as convolution and activation functions. The Convolution Operation's goal is to extract high-level characteristics from the input image, such as edges. There is no need to limit Convolutional networks to just one convolutional Layer. The first convolutional layer is traditionally responsible for capturing Low-Level information such as edges, color, or gradient direction. With the addition of layers, the architecture adjusts to the High-Level characteristics as well, giving us a network that understands the photos in the dataset in the same way that we do. [8] [7]

###### 1.1.3.1.1.1 Convolution

A convolution is a form of linear operation used mostly for feature extraction in which a tiny array of numbers called a kernel is applied over the input, which is a tensor of numbers. At each point of the tensor, and element-wise product between each element of the kernel and the input tensor is calculated and summed to generate the output value in the corresponding place of the output tensor, referred to as a feature map. This technique is repeated with different kernels to generate an arbitrary number of feature maps that represent distinct features of the input tensors, different kernels may thus be regarded as different feature extractors. The size and number of kernels are two fundamental hyperparameters that determine the convolution operation. The former is usually 3 x 3, but it can also be 5 x 5 or 7 x 7. The depth of output feature maps is determined by the latter, which is arbitrary. [8]

The convolution method prevents the center of each kernel from overlapping the input tensor's outermost element, resulting in a smaller output feature map in terms of height and breadth. Padding, most commonly zero padding, is a technique for dealing with this problem that involves adding rows and columns of zeros on each side of the input tensor to fit the center of a kernel on the outermost element while maintaining the same in-plane dimension through the convolution operation. To keep in-plane dimensions so that more layers may be applied, modern CNN architectures commonly use zero padding. Without zero padding, following the convolution procedure, each succeeding feature map would be smaller. A stride is a separation between two consecutive kernel points, and it also specifies the convolution procedure. A stride of one is the most usual choice, but a stride of greater than 1 is occasionally used to accomplish feature map downsampling. A pooling process is an alternate method for downsampling. [8]

Weight sharing is an essential component of a convolution operation, kernels are shared among all image places. Convolution operations include the following properties because of weight sharing:

1. Allowing kernels to extract local feature patterns that are translation constant as they go across all image locations and detect learned local patterns.
2. Downsampling in combination with a pooling operation allows for the learning of spatial rankings of feature patterns, resulting in the capture of an ever-broader field of view.
3. In comparison to fully connected neural networks, model efficiency is improved by lowering the number of parameters to train. [8]

The procedure of training a CNN model for the convolution layer is to discover the kernels that perform great for a given job based on a training dataset, as discussed below. Kernels are the only parameters in the convolution layer that are automatically learned during the training process, however, the size of the kernels, number of kernels, padding, and stride are hyperparameters that must be defined before the training process begins. [8]

###### 1.1.3.1.1.2 Nonlinear Activation Function

A nonlinear activation function is applied to the results of a linear operation like convolution. Although smooth nonlinear functions such as the sigmoid or hyperbolic tangent (tanh) function were formerly utilized since they are mathematical representations of biological neuron action, the rectified linear unit (ReLU) is now the most used nonlinear activation function. f(x) = max(0, x) [8]

##### 1.1.3.1.2 Pooling Layer

A pooling layer performs a standard downsampling operation on the feature maps, reducing their in-plane dimensionality to introduce translation invariance to tiny shifts and distortions and reducing the number of learnable parameters. It is worth noting that none of the pooling layers have learnable parameters, although filter size, stride, and padding are hyperparameters in pooling operations, much like convolution operations. [8]

###### 1.1.3.1.2.1 Max Pooling

Max pooling is the most common type of pooling procedure, which takes regions from the input feature maps, produces the largest value for each patch, and discards the rest. In practice, max pooling with a size 2 x 2 filter and a stride of two is usually utilized. The in-plane dimension of feature maps is reduced by a factor of two. The depth dimension of feature maps, unlike the height and width dimensions, does not change. [8]

###### 1.1.3.1.2.2 Global Average Pooling

A global average pooling operation is also worth mentioning. A global average pooling is an extreme sort of downsampling in which a feature map of height width is downsampled into a 1 x 1 array by simply taking the average of all the components in each feature map, while the depth of feature maps is preserved. Before the fully connected layers, this step is usually performed only once. The following are some of the benefits of using global average pooling:

1. Lowers the number of parameters that can be learned.
2. Allows the CNN to accept variable-size inputs. [8]

##### 1.1.3.1.3 Fully Connected Layer

The final convolution or pooling layer's output feature maps are typically flattened, that is, converted into a 1D array of numbers (or vector), and linked to one or many fully connected layers, also known as dense layers, whereby each input is connected to each output by a learnable weight. Once the features that are extracted by the convolution layers and downsampled by the pooling layers are formed, they are transferred to the network's final outputs, such as the probabilities for each class in classification tasks, by a subset of fully connected layers. The number of output nodes in the final fully connected layer is usually equal to the number of classes. As previously mentioned, each fully linked layer is followed by a nonlinear function, such as ReLU. [8]

##### 1.1.3.1.4 Last Layer Activation Function

The last fully connected layer's activation function is frequently distinct from the others. Each activity necessitates the selection of an appropriate activation function. The SoftMax function, which normalizes output true values from the last fully connected layer to target class probabilities, where each value ranges between 0 and 1 and all values total to one, is an activation function used in the multiclass classification problem. [8]

##### 1.1.3.1.5 Dropout

We employed a technique called dropout, which was originally designed for training deep neural networks, in addition to pooling. The goal of the dropout strategy is to keep the model from learning too much from the training data. So, in each learning cycle during the training phase, each neuron has a chance of not being taught in that cycle equal to some dropout rate. This keeps the model from being excessively flexible, which aids the learning algorithm in coming to a model that is not overly fitted to the training data and can instead be generalized well for predicting unlabeled future data. [9]

#### 1.1.3.2 Training a network

Finding kernels in convolution layers and weights in fully connected layers that minimize disparities between output predictions and supplied ground truth labels on a training dataset is the process of training a network. A backpropagation algorithm is a popular approach for training neural networks, and it relies heavily on the loss function and the gradient descent optimization process. A loss function calculates a model's performance under specific kernels and weights using forward propagation on a training dataset, and learnable parameters, such as kernels and weights, are updated based on the loss value using optimization algorithms such as backpropagation and gradient descent, among others. [8]

##### 1.1.3.2.1 Loss Function

The compatibility of the network's output predictions by forwarding propagation and provided ground truth labels is measured by a loss function, also known as a cost function. Cross entropy is a frequently used loss function for multiclass classification, although the mean squared error is commonly used for regression to continuous values. One of the hyperparameters is the type of loss function, which must be determined based on the specified jobs. [8]

##### 1.1.3.2.2 Gradient Descent

Gradient descent is a popular optimization approach that iteratively adjusts the network's learnable parameters, such as kernels and weights, to minimize loss. Each trainable parameter is adjusted in the negative reverse path with an arbitrary step size set by a hyperparameter usually called learning rate, and the gradient of the loss function supplies us with the path in which the function has the highest rate of rising. The gradient is a partial derivative of the loss concerning each trainable parameter, and the gradient is a partial derivative of the loss concerning each trainable parameter. A single update of a parameter is formulated as follows:

where w represents each learnable parameter, α represents a loss function, and L represents a learning rate. It is worth noting that one of the most critical hyperparameters to set before the training begins is the learning rate. In practice, the gradients of the loss function concerning the parameters are computed using a subset of the training dataset termed mini-batch and applied to parameter updates due to memory constraints. This method is known as mini-batch gradient descent, also known as stochastic gradient descent (SGD), and the mini-batch size is also a hyperparameter. Furthermore, several enhancements to the gradient descent algorithm, such as SGD with momentum, RMSprop, and Adam, have been suggested and widely used. [8]

## Problem statement.

Financial markets are the beating heart of the global economy, with billions of dollars changing hands every day. A strong forecast of market behavior in the future would be incredibly useful in a variety of situations. FOREX markets play a key role in economic growth. Behavior analysis and future prediction can be extremely helpful in reaching these economic goals. Another major application of forex market predictions can be found in forex market trading systems, these usually consist of several modules used in prediction, risk analysis, and trading strategies. The goal of a trading module would be to maximize the overall profit over loss ratio in favor of the profits. However, a prediction module focuses on the sub-problem of predicting the future of the markets, which can be an unbelievably valuable piece of information in the process of stock trading. Hence, the performance of the trading module and the extent of the performance of the whole trading system would be influenced by the quality of predictions the module can make. In fact, without reliable predictions, it is impossible to have an excellent trading system.

A proven technique for these predictions would be those done with the use of machine learning. The most common algorithms utilized for this purpose are artificial neural networks (ANN) and support vector machines (SVM). Some other tools that have been applied for feature extraction from raw financial data and/or making predictions based on a set of variables are statistical methods, random forests, linear discriminant analysis, logistic regression, and evolutionary computing algorithms, especially genetic algorithms (GA).

Deep learning (DL) is a family of current techniques that can extract and predict features automatically. DL approaches have been demonstrated to be capable of progressively constructing meaningful complex features from raw data or simpler variables in a variety of disciplines, including machine vision and natural language processing. Because forex market behavior is complicated, unpredictable, and messy, extracting features that are useful enough to make predictions is a major difficulty, and DL is a viable solution. Deep learning algorithms used to predict stock markets include deep multilayer perceptron (MLP), restricted Boltzmann machine (RBM), long short-term memory (LSTM), autoencoder (AE), and convolutional neural network (CNN), so the hope is that this can also be applied to forex markets.

It is crucial to take note of the variety of variables that can be used to build predictive models. For a market prediction job, raw price data, technical indicators derived from historical data, other markets with a relationship to the target market, currency exchange rates, oil price, and a variety of other variables can be beneficial. Unfortunately, aggregating such a broad variety of data in such a way that an autonomous market prediction algorithm can use it is usually a difficult undertaking. As a result, most of the existing research in this sector has concentrated on a range of technical indicators that represent a particular market's recent history.

Automatic feature extraction is another key issue in the field. The initial variables are basic since they were designed to be utilized by human experts, and even if they were selected by a finance expert with sufficient knowledge and expertise in this sector, they may not be the best feasible choices for machine prediction. To put it another way, an ideal automatic approach to stock market prediction can extract valuable features from various variables that are good for predicting markets, train a prediction model based on those derived features, and then make forecasts using the model. This research focuses on the first step in the process: creating a model for feature extraction from a set of variables containing data from previous market records. This information contains initial basic factors such as raw price history, technical indications, and the change in those variables over time. A deep learning algorithm such as CNN is a potential approach for this feature extraction problem, due to the diversity of the input space and the sophistication of the feature set that may be necessary for a decent prediction.

We build our architecture on CNN because of its demonstrated capabilities in other domains. As a test case, we'll demonstrate how CNN can be used in our proposed framework, CNNpred, to capture potential relationships among different variables for retrieving combined features from a diverse given set of inputs from eight currency pairs related to the US Dollar to Nigerian Naira (USD/NGN) currency pair: Australian Dollar to U.S. Dollar (AUD/USD), European Euro to U.S. Dollar (EUR/USD), Great Britain Pound to U.S. Dollar (GBP/USD), U.S. Dollar to Japanese Yen (USD/JPY), as well as other variables including exchange rate of currencies, futures contracts, price of commodities, important indices of markets around the world, price of major companies in U.S. market, treasury bill rates, and USD/NGN as well. In addition, the filters are created in a way that is suitable to the financial characteristics of the variables.

This work's primary contributions can be summed up as follows:

1. Using a CNN-based framework to combine numerous factors for extracting features and market forecasting. Because the behavior of financial markets is influenced by a variety of factors, it is critical to obtain as much information as possible. Our initial variable collection covers various elements of forex-related variables, and it can be simply expanded to cover more variables.
2. This research proposes a CNN that uses a 3-dimensional tensor to combine and align a diverse set of input variables before training the network to extract valuable features for predicting each of the relevant forex markets.

According to [10] the major disadvantages forex traders face is.

1. Changes in the exchange rate: Frequent fluctuations in currency exchange rates can be negative, as economic, and political influences can produce market price uncertainty.
2. Absence of centralized Exchanges: There is no central regulator or exchange to set base prices in the spot market, resulting in pricing differences amongst brokers.
3. Higher Leverage: Although traders can conduct forex transactions at a minimal cost, increased leverage can represent a bigger risk of loss if exchange rate swings impair earnings.

All these factors increase the risk of trading foreign currency markets but using a solid prediction algorithm can drastically lower the risk.

Forex trading can also be a daunting task and keeping track of factors that influence the market can be extremely time-consuming even experienced traders find this extremely difficult and tiring.

Even though the Forex market is the world's largest, most of the work on financial market prediction using deep learning focuses on the stock market and general market prediction.

## Aim and objectives

### Aim

The goal of this study is to predict the direction of the Forex market for the next day, which is a binary classification issue.

### Objectives

1. To use a deep learning architecture that can be used to predict forex markets.
2. To generate a dataset that can be applied to the developed architecture.
3. To train the deep learning model using the generated dataset.
4. To evaluate the trained model.

## Scope and limitations

### 1.4.1 Scope

The purpose of this study is to provide traders with a better understanding of market movements so they can make better trade decisions. It is not the intention to completely replace human analysis.

### 1.4.2 Limitations

In financial market prediction systems, as in other prediction efforts, it is always impractical to assume high accuracy. This is a project that traders and investors may find useful, but no miracle can make you rich overnight.

## 1.5 Summary and report

With more than five trillion dollars transacted each day, Forex is the only financial market that is open 24 hours a day. There is currently no completely reliable approach for predicting market direction. To generate the best potential prediction regarding the market's direction, we proposed using a Convolutional Neural Network to discover patterns in the dataset.

# Chapter 2: LITERATURE REVIEW

Since the use of deep learning approaches to predict foreign exchange markets is mostly unexplored, the reviewed literature is based on similar literature.

## 2.1 Existing/Similar Systems

### 2.1.1 Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach [11]

They introduce CNN-TA, a novel algorithmic trading model based on image processing properties, and utilize a 2-D Convolutional Neural Network in this research. To convert financial time series into 2-D images, fifteen different technical indicators with various parameter values are used. Each indicator instance generates data for 15 days. As a result, 15x15-inch 2-D images are formed. Each image is then labeled as Buy, Sell, or Hold depending on the original's hills and valleys. A set of circumstances When compared to the Buy & Hold and Buy & Sell strategies, the results reveal that the trained model beats the untrained model over an extended out-of-sample period across other widely used trading platforms.

In this paper, they suggest a unique method for converting 1-D financial time series into a 2-D image-like data representation such that deep convolutional neural networks can be used to power an algorithmic trading system. Fifteen separate technical indicator instances with varying parameter settings, each with 15 days, were modified to reflect the values in each column to produce such a representation. Similarly, each row's x-axis contains a time series of 15 days' worth of data for each technical indicator. The rows are also arranged in such a way that related indicators are grouped to meet the y-axis locality requirements. As a result, pictures with a resolution of 15x15 pixels are created and input into the deep convolutional neural network. According to them, the 2-D representation of financial technical analysis time series data and feeding it as the input for a 2-D image classification based on deep CNN, namely CNN-TA, for a financial trading system is unique, because it has never been used not only for any trading system but also in any financial prediction model in the way they propose. According to performance evaluations, such a strategy operates astonishingly effectively even over extended periods. On short and long out-of-sample periods, the proposed model outperformed Buy & Hold, common technical indicator-based models, the most generally used neural network, namely MLP, and the state-of-the-art deep learning time series forecasting model, namely LSTM. Even though this is one of the first attempts at adopting such an unusual technique, they think the proposed model has potential. Additionally, they anticipate that parameter optimization and model fine-tuning could improve performance even more.

### 2.1.2 Financial Time-series Data Analysis using Deep Convolutional Neural Networks [12]

Based on deep learning techniques, this study suggests a novel financial time-series analysis tool. This study focuses on time-series data processing and financial market forecast. This paper's most significant contribution is to improve algorithmic trading.

This research proposes a new financial time-series analysis method based on deep learning techniques. Deep learning research has exploded in popularity in recent years, resulting in a slew of successful applications in artificial intelligence and multimedia disciplines like visual recognition, robot vision, and natural language processing. They focus on time-series data processing and prediction in financial markets in this research. Several technical indicators and expert rules are employed to extract numerical features in traditional feature extraction methodologies in intelligent trading decision support systems. The proposed planar feature representation approaches and deep convolutional neural networks improve the algorithmic trading framework, which is the paper's main contribution (CNN). The suggested approach is constructed and benchmarked using Taiwan Stock Index Futures’ historical datasets. The findings of the experiments suggest that deep learning is effective in their trading simulation application and that it may have more potential to model noisy financial data and complicated social science problems. They anticipate that the presented methodologies and deep learning framework would be used in more novel applications in the next generation of financial technology (FinTech).

The main contribution of this paper is:

1. To turn time series data into 2D graphics, they develop the mean average mapping method (MAM) and the double moving average mapping method (DMAM). The modified photos must not lose any information and must be recognized by CNN to be used in training.
2. CNN can collect useful information and accurately categorize the price trend using their proposed altered images. CNN will no longer be confined to 2D data visualization and will be capable of financial forecasting and other time series analysis.

They believe that this study is an excellent example of novel Financial Technology (FinTech) applications that have attempted to use modern technology to solve financial challenges and produce novel applications. To extract more hidden knowledge and contribute to service automation, artificial intelligence and big data analysis play essential roles in this burgeoning study field. They also believe it can be used in intelligent trading systems including high-frequency trading and algorithmic trading, as well as a variety of customized, personalized, and unmanned service designs.

### 2.1.3 CNN-based stock market prediction using a diverse set of variables [9]

In this study, they suggest a CNN-based framework for analyzing data from a variety of sources, including diverse markets. The suggested framework has been used to forecast the direction of the next day's weather, as well as the movement of the S&P 500, NASDAQ, DJI, NYSE, and RUSSELL indexes, depending on a range of starting variables, to find qualities that can be used to forecast the future of such markets. The evaluations show a significant improvement in prediction performance when compared to the state-of-the-art baseline.

One of the most essential difficulties in the market prediction arena is feature extraction from financial data, for which several methodologies have been proposed. Convolutional neural networks (CNN) have lately been used for autonomous feature selection and market prediction, among other things. However, in the tests so far, the correlation between different marketplaces as a prospective source of information for extracting features has received less attention. In this research, they propose a CNN-based framework that can be used to extract features from a collection of data from a range of sources, including multiple markets, to forecast the future of those markets. Based on various sets of initial variables, the proposed framework has been used to predict the next day's direction of movement for the S&P 500, NASDAQ, DJI, NYSE, and RUSSELL indexes. When compared to state-of-the-art baseline methods, the evaluations demonstrate a considerable improvement in prediction performance.

They build their approach on CNN because of its demonstrated skills in other domains, as well as successful previous studies in the market prediction arena. As a test case, they show how CNN can be used in their proposed framework, CNNpred, to capture possible correlations among different variables for extracting combined features from a diverse set of input data from five major U.S. stock market indices: S&P 500, NASDAQ, Dow Jones Industrial Average, NYSE, and RUSSELL, as well as other variables such as currency exchange rates, futures contracts, commodity price 90, and important market indices. Additionally, their filters are created in a method that is suitable for financial variables' properties.

The main contributions of this paper can be summarized as follows:

1. Aggregating several variables in a CNN-based framework for feature extraction and market prediction. Since financial market behavior is affected by many factors, it is important to gather related information as much as possible. Their initial variable set covers various aspects of stock-related variables well and, it can be easily extended to cover other possible variables
2. To their knowledge, this is the first work suggesting a CNN which takes a 3-dimensional tensor to aggregate and align a diverse set of variables as input and then trains the network in a way that extracts useful features for predicting each of the pertinent stock markets.

### 2.1.4 Predicting Financial Prices of Stock Market using Recurrent Convolutional Neural Networks [13]

They propose combining architectures for a trading signal prediction that takes advantage of both CNN and RNN advantages at the same time in this study. Their model is fed into a GRU layer to capture long-term signal relationships and then presented to financial time series forecasting signals via a CNN layer. In sequential learning tasks, the GRU model outperforms standard RNNs and overcomes the vanishing gradients and exploding problems. They compare their model to existing deep learning approaches using three datasets from the Hang Seng Indexes (HSI), the Deutscher Aktien Index (DAX), and the S&P 500 Index from 2008 to 2016.

Financial time-series prediction has been one of the most difficult problems in financial market analysis for a long time. Deep neural networks are an effective data mining approach that has gotten a lot of attention from academics in numerous fields of time-series prediction during the last decade. "For financial predictions, convolutional neural network (CNN) and recurrent neural network (RNN) models have become the standard. In this research, they suggest combining architectures for the trading signal prediction that take advantage of the advantages of both CNN and RNN at the same time. Their model is fundamentally fed into a gated recurrent unit (GRU) layer to capture long-term signal dependencies and presented to financial time series forecasting signals via a CNN layer. The GRU model outperforms ordinary RNNs in sequential learning tasks and solves the vanishing gradients and exploding problem. They evaluate their model on three datasets for stock indexes from the Hang Seng Indexes (HSI), the Deutscher Aktienindex (DAX), and the S&P 500 Index from 2008 to 2016 and compare it to existing deep learning methods. According to the results of the experiments, the suggested GRU-CNN model had the best prediction accuracy of 56.2 percent on the HIS dataset, 56.1 percent on the DAX dataset, and 56.3 percent on the S&P500 dataset.

They use a combination approach in this study and design two network architectures; their contributions are summarized as follows:

1. They focus on updating the standard gated recurrent unit (GRU) model, which tackles the vanishing gradient and expanding issue of standard RNNs over the gating mechanism while keeping the influence of LSTM (is a well-known variant of traditional RNNs).
2. They proposed combining GRU and CNN architecture to find financial marketing predictions based on return predictive signals.
3. They used an attention mechanism to train their model (GRU-CNN) and compared its performance to that of typical deep learning models.
4. Experiments reveal that their enhanced GRU-CNN model outperforms earlier standard techniques in terms of prediction accuracy.

The current GRU-based model achieves good accuracies and higher returns in statistics and economics. However, the suggested GRU-CNN model outperforms the GRU-based model by a little margin.

### 2.1.5 Financial Markets Prediction with Deep Learning [14]

Because of their complex system dynamics, financial markets are hard to predict. Although several recent research has used machine learning approaches to predict financial markets, the results have not been satisfying in terms of financial returns. To predict financial market movement, this research introduces a novel one-dimensional convolutional neural networks (CNN) model. Different forms of data, such as prices and volume, share parameters (kernels) as the tailored one-dimensional convolutional layers scan financial trading data over time. Instead, then using standard technical indicators, their algorithm extracts characteristics automatically, avoiding biases induced by technical indicator selection and pre-defined correlations in technical indicators. They use strict back-testing on historical trading data of six futures from January 2010 to October 2017 to evaluate the performance of their prediction model. Their CNN model extracts more generic and informative characteristics than traditional technical indicators and achieves more stable and profitable financial performance than earlier machine learning approaches, according to the findings of their trial.

According to up-to-date information, this is one of the first attempts to utilize deep CNN to predict financial markets, and they conduct strict back-tests to check the model's success. Six futures from the Chicago Mercantile Exchange and the New York Mercantile Exchange are used to evaluate their concept. Back-test findings reveal that their 1-D CNN model outperforms earlier techniques based on Nearest Neighbour, SVM, and Deep Feedforward Networks in terms of average yearly return and robustness (better Sharpe ratio1).

They also notice that their 1-D CNN model, which does not employ technical indicators as input, produces better outcomes than the model that does. This demonstrates that their 1-D CNN model can extract more generic and informative characteristics than traditional technical indicators can. Their findings back up the idea that conventional machine learning metrics like accuracy and F1 score are not suited for predicting financial markets since different forms of prediction errors have varied effects on financial performance. They suggest a Weighted-F-Score, a tweaked version of the F-measure score, to address this issue in this study. In their back-test results, their tests reveal that Weighted-F-Score has a strong correlation with average annual return and Sharpe ratio, with minimal cross-correlate coefficients of 0.79 and 0.84, accordingly.

They summarize their contributions as follows:

1. **Cross-Data-Type One-Dimensional Convolution.** Regular 2-D convolutions are not immediately applicable to this study since they may not be able to convolve different pieces of historical financial trading data, such as price and volume, to yield significant results. Regular 1-D convolution is also inapplicable because it is unable of capturing the features required to depict the combined distribution of elements in financial historical trade data. To solve the problem, they offer Cross-Datatype 1-D convolution, a version of 1-D convolution. Their model outperforms previous methods by 6.1 percent -53.0 percent on average annual return and 53.0 percent -199.0 percent on Sharpe ratio, based on customized 1-D convolution.
2. **Deep Features Auto-Extraction.** Technical indicators are used as input in many machine learning systems for financial market prediction. To capture more generalized and useful information, this work recommends replacing typical technical indicators with deep features recovered using Cross-Datatype 1-D convolution. To the best of their knowledge, this is one of the first systems that use CNN instead of typical technical indicators to extract features. Their research shows that this strategy produces state-of-the-art results in both finance and machine learning benchmarks.
3. **Correlation between Finance and Machine Learning Metrics.** The effects of various forms of prediction errors on financial market trading are diverse. For example, if the prediction is up when the price goes down or vice versa, the trader will lose more money.

However, standard machine learning measurements like accuracy and F1 score do not distinguish between different sorts of errors, hence they only have a poor correlation with financial metrics. To overcome the weak link between Machine Learning and Finance measurements, they suggest Weighted-F-Score, a modified form of the F score for Type I and Type II mistakes. According to their findings, the Weighted-F-Score has a strong relationship with financial measures (the minimum Cross-correlation value is 0.79).

# Chapter 3: System Methodology

## Dataset

The dataset employed in this paper is mostly influenced by the dataset created by [9], except for the fact that the dataset used in [9] is of five different U.S stock market indices.

We present our data as a matrix in CNN models, the features across multiple timesteps are presented as a 2D array. The data are 5 CSV files, each for a different currency pair. The input data has a date column and a name column to identify the ticker symbol for the currency pair. We leave the date column as the time index and remove the name column. The remaining values are numerical.

As previously stated, our goal is to construct a model for predicting the direction of price fluctuations in the Forex market. Our method was used to forecast the movement of the USD/NGN currency pair. We employ eighty-two variables to represent each day of each currency pair in this prediction job. Some of these variables are particular to a currency pair, while the remainder is economic variables that are reproduced across all currency pairs in the data set. Primitive variables, technical indicators, world stock market indexes, the exchange rate of the US dollar against other currencies, commodities, data from significant corporations in the US markets, futures contracts, and other crucial factors are among the many variables available. A few of these variables are significant because they indicate mechanisms that directly or indirectly affect the Forex markets. Other variables, on the other hand, are important because they provide hints or signals that can aid the system in predicting the markets' short-term future, even if they do not represent causal relationships. We will go over the diverse groups in our variable set briefly here, with more information in the table below.

1. Primitive Variable: In this study, the close price, and the day of the week in which the forecast is to be made are the only primitive variables employed.
2. Technical Indicators: To examine short-term price movement, technical analysts employ technical indicators derived from historical data of Forex prices and trading information. In Forex market research, they are quite common. This sort of variable includes things like moving averages.
3. World Stock Markets: Because of the phenomenon of economic globalization, Forex markets all over the world usually interact with one another. When we consider the time differences between countries, this relationship becomes even more valuable, as it allows us to obtain insight into the future of a country's market by monitoring the markets of other countries. Consider the impact of other countries' markets on the US market, such as China, Japan, and South Korea.
4. The Exchange Rate of U.S. Dollar: Multinational corporations import their requirements from many other nations or distribute their products to other countries. As a result, the fluctuation of the US dollar against other currencies such as the Canadian dollar and the European Euro has an impact on the profit of these businesses. When a shift in profit is disclosed, demand for these companies' stock fluctuates, as does the price of their stock. The stock values of domestic enterprises are also influenced by changes in multinational company demand. As a result, stock prices are influenced by currency exchange rates directly or indirectly, and vice versa.
5. Commodities: Another element that can be utilized to forecast Forex market behavior is the price of commodities such as gold, silver, oil, or wheat. This type of data might provide insight into the global market. Researchers have discovered a correlation connecting commodities and Forex markets, particularly during the US monetary crisis of 2007-2008. Furthermore, commodities, like equities and forex trading, have become an essential part of portfolios. This means that information regarding commodity prices can be beneficial in predicting Forex price variations.
6. Big U.S. Companies: Different equities are used to generate market indices in the United States. In this formula, each stock has a weight that corresponds to its market share. To put it another way, big corporations matter more than tiny ones when it comes to predicting market indices in the United States. Exxon Mobil Corporation and Apple Inc. are two examples of this. Because of the influence, they have on the US economy and, as a result, the US Dollar, the usage of US market index prices is critical.
7. Futures Contracts: Futures contracts are agreements in which one party promises to provide stocks, commodities, or other items in the future. These contracts show the estimated future value of the product. Stocks with a higher predicted value than their present worth are more likely to be purchased by investors. S&P 500 Futures, DJI Futures, and NASDAQ Futures, for example, could have an impact on the current price of the S&P 500 and other indices.
8. Other Useful Variables: Other variables, such as Treasury bill rates, term, and default spreads, be effective in predicting the US stock market.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Variable | Description | Type | Source / Calculation |
| 1 | Day | Which Day of the Week | Primitive | Pandas |
| 2 | Close | Close Price | Primitive | Excel STOCKHISTORY Formula |
| 3 | MOM – 1 | Return 2 days before | Technical Indicator | Formula |
| 4 | MOM – 2 | Return of 3 days before | Technical Indicator | Formula |
| 5 | MOM – 3 | Return of 4 days before | Technical Indicator | Formula |
| 6 | ROC – 5 | 5 days Rate of Change | Technical Indicator | Formula |
| 7 | ROC – 10 | 10 days Rate of Change | Technical Indicator | Formula |
| 8 | ROC – 15 | 15 days Rate of Change | Technical Indicator | Formula |
| 9 | ROC – 20 | 20 Days Rate of Change | Technical Indicator | Formula |
| 10 | EMA – 10 | 10 days Exponential Moving Average | Technical Indicator | Formula |
| 11 | EMA – 20 | 20 days Exponential Moving Average | Technical Indicator | Formula |
| 12 | EMA – 50 | 50 days Exponential Moving Average | Technical Indicator | Formula |
| 13 | EMA – 200 | 200 days Exponential Moving Average | Technical Indicator | Formula |
| 14 | DTB4WK | 4 – Week Treasury Bill: Secondary Market Rate | Other | FRED |
| 15 | DTB3 | 3 – Month Treasury Bill: Secondary Market Rate | Other | FRED |
| 16 | DTB6 | 6 – Month Treasury Bill: Secondary Market Rate | Other | FRED |
| 17 | DGS5 | 5 – Year Treasury Constant Maturity Rate | Other | FRED |
| 18 | DGS10 | 10 – Year Treasury Constant Maturity Rate | Other | FRED |
| 19 | DAAA | Moody’s Seasoned Aaa Corporate Bond Yield | Other | FRED |
| 20 | DBAA | Moody’s Seasoned Baa Corporate Bond Yield | Other | FRED |
| 21 | TE1 | DGS10 – DTB4WK | Other | FRED |
| 22 | TE2 | DGS10 – DTB3 | Other | FRED |
| 23 | TE3 | DGS10 – DTB6 | Other | FRED |
| 24 | TE5 | DTB3 – DTB4WK | Other | FRED |
| 25 | TE6 | DTB6 – DTB4WK | Other | FRED |
| 26 | DE1 | DBAA – DAAA | Other | FRED |
| 27 | DE2 | DBAA – DGS10 | Other | FRED |
| 28 | DE4 | DBAA – DTB6 | Other | FRED |
| 29 | DE5 | DBAA – DTB3 | Other | FRED |
| 30 | DE6 | DBAA – DTB4WK | Other | FRED |
| 31 | CTB3M | Change in the market yield on U.S Treasury securities at 3-month Constant maturity, quoted on an investment basis | Other | FRED |
| 32 | CTB6M | Change in the market yield on U.S Treasury securities at 6-month Constant maturity, quoted on an investment basis | Other | FRED |
| 33 | CTB1Y | Change in the market yield on U.S Treasury securities at 1-Year Constant maturity, quoted on an investment basis | Other | FRED |
| 34 | OIL | Relative change of oil price(WTI), Oklahoma | Commodity | FRED |
| 35 | OIL | Relative change of oil price(Brent) | Commodity | Investing.com |
| 36 | OIL | Relative change of oil price(WTI) | Commodity | Investing.com |
| 37 | GOLD | Relative change of gold price(London markets) | Commodity | FRED |
| 38 | GOLD – F | Relative change of gold price futures | Commodity | Investing.com |
| 39 | XAU – USD | Relative change of gold spot U.S. Dollar | Commodity | Investing.com |
| 40 | XAG – USD | Relative change of silver spot U.S. Dollar | Commodity | Investing.com |
| 41 | GAS | Relative change in gas price | Commodity | Investing.com |
| 42 | SILVER | Relative change in the silver price | Commodity | Investing.com |
| 43 | COPPER | Relative change of copper future | Commodity | Investing.com |
| 44 | IXIC | Return of NASDAQ Composite index | World Indices | Yahoo Finance |
| 45 | GSPC | Return of S&P 500 index | World Indices | Yahoo Finance |
| 46 | DJI | Return of Dow Jones Industrial Average | World Indices | Yahoo Finance |
| 47 | NYSE | Return of NY stock exchange index | World Indices | Yahoo Finance |
| 48 | RUSSELL | Return of RUSSELL 2000 index | World Indices | Yahoo Finance |
| 49 | HIS | Return of Hang Seng index | World Indices | Yahoo Finance |
| 50 | SSE | Return of Shang Hai Stock Exchange Composite index | World Indices | Yahoo Finance |
| 51 | FCHI | Return of CAC 40 | World Indices | Yahoo Finance |
| 52 | FTSE | Return of FTSE 100 | World Indices | Yahoo Finance |
| 53 | GDAXI | Return of DAX | World Indices | Yahoo Finance |
| 54 | USD – Y | Relative change of U.S. Dollar to Japanese Yen exchange rate | Exchange Rate | Yahoo Finance |
| 55 | USD – GBP | Relative change of U.S. Dollar to Great Britain Pound exchange rate | Exchange Rate | Yahoo Finance |
| 56 | USD – CAD | Relative change of U.S. Dollar to Canadian Dollar exchange rate | Exchange Rate | Yahoo Finance |
| 57 | USD – CNY | Relative change of U.S. Dollar to Chinese Yuan exchange rate | Exchange Rate | Yahoo Finance |
| 58 | USD – AUD | Relative change of U.S. Dollar to Australian Dollar exchange rate | Exchange Rate | Investing.com |
| 59 | USD – NZD | Relative change of U.S. Dollar to New Zealand Dollar exchange rate | Exchange Rate | Investing.com |
| 60 | USD – CHF | Relative change of U.S. Dollar to Swiss Franc exchange rate | Exchange Rate | Investing.com |
| 61 | USD – EUR | Relative change of U.S. Dollar to European Euro exchange rate | Exchange Rate | Investing.com |
| 62 | USDX | Relative change in U.S. Dollar index | Exchange Rate | Investing.com |
| 63 | XOM | Return of Exon Mobil Corporation | U.S. Companies | Yahoo Finance |
| 64 | JPM | Return of JP Morgan Chase & Co. | U.S. Companies | Yahoo Finance |
| 65 | AAPL | Return of Apple Inc. | U.S. Companies | Yahoo Finance |
| 66 | MSFT | Return of Microsoft Corporation | U.S. Companies | Yahoo Finance |
| 67 | GE | Return of General Electric Company | U.S. Companies | Yahoo Finance |
| 68 | JNJ | Return of Johnson & Johnson | U.S. Companies | Yahoo Finance |
| 69 | WFC | Return of Wells Fargo & Company | U.S. Companies | Yahoo Finance |
| 70 | AMZN | Return of Amazon.com Inc. | U.S. Companies | Yahoo Finance |
| 71 | FCHI – F | Return of CAC 40 Futures | Futures | Investing.com |
| 72 | FTSE – F | Return FTSE 100 Futures | Futures | Investing.com |
| 73 | GDAXI – F | Return of DAX Futures | Futures | Investing.com |
| 74 | HSI – F | Return of Hang Seng Index futures | Futures | Investing.com |
| 75 | NIKKEI – F | Return of Nikkei index Futures | Futures | Investing.com |
| 76 | KOSPI – F | Return of Korean stock exchange Futures | Futures | Investing.com |
| 77 | IXIC – F | Return of NASDAQ Composite index Futures | Futures | Investing.com |
| 78 | DJI – F | Return of Dow Jones Industrial Average Futures | Futures | Investing.com |
| 79 | S&P – F | Return of S&P 500 index Futures | Futures | Investing.com |
| 80 | RUSSELL – F | Return of RUSSELL Futures | Futures | Investing.com |
| 81 | USDX – F | Return of U.S. Dollar Index Futures | Exchange Rate | Investing.com |
| 82 | TNX | Treasury Yield 10 Years | Other | Yahoo Finance |

Figure 3: List of Features in dataset

## 3.2 Architecture

The architecture of this paper is built around the work done on [15], by making improvements to [9]. This paper takes the architecture of [15] and implements the dataset we created, while also adjusting it to improve the accuracy of the model in tandem with our dataset.

### 3.2.1 Pre-processing of Data

[15] attempts to construct a classification label before attempting to forecast market direction. The market direction is determined by comparing the closing value of tomorrow to the closing value of today. We can get the % change if we read the data into a pandas DataFrame, which is a positive change for the market going up. As a result, they change the label to a single-time step back. They calculate the closing index's percentage change and compare it to the previous day's data. Then, depending on whether the percentage change is positive or negative, they convert the data to 1 or 0.

We read each of the eight data files in the folder as a distinct pandas DataFrame and store them in a Python dictionary. The result is a DataFrame for each currency pair, with the column "Target" serving as the classification label and the other columns serving as input features. [15] also uses a common scaler to standardize the data. In time series problems, it is usually better to specify a cut-off point where the data before the cut-off is the training set and the data after the cut-off is the test set, rather than randomly splitting the data into training and test sets. The scaling is centered on the training set, but it is applied throughout the full dataset.

### 3.2.2 Data Generator

The architecture defined by [15] does not use all the time steps at once; instead, they use a set of N time steps to forecast the market's direction at step N+1. The window of N time steps in this model can begin anywhere. We can simply create a large quantity of DataFrames with many overlaps. They then create a data generator for training and validation to save memory.

In Python, a generator is a particular function that does not output a value but instead produces a stream of data in iterations. To be employed in Keras training, a generator must produce a batch of data inputs and a target. This generator is designed to continue running indefinitely. As a result, the generator function is built using an infinite loop that begins with True. In each iteration, it selects one DataFrame at random from the Python dictionary, then starts from a random point and takes N time steps using the pandas "iloc[start: end]" syntax to create an input under the variable frame inside the set of time steps of the training set (i.e., the beginning section). A 2D array was used for this DataFrame. The last time step's label is used as the target label. After that, the input data and label are merged into the list batch. It was then deployed from the generator once it had amassed enough for one batch size.

The data generator's final phase is to send out a batch for training or validation. [15] then puts a list of input data (each a 2D array) and a list of target labels into variables X and y, then transform them to NumPy arrays so that their Keras model can deal with them. Because of the network model's design, they then call "np. expand dims()" to add one extra dimension to the NumPy array X.

### 3.2.3 The Model

The model used in [15] are two, the first being a 2D CNN model and the other being a 3D CNN model, both gotten from [9].

[9]’s 2D CNN model takes an input tensor of type N x m x 1 with N being the number of time steps and m being the number of features in each time step. N = 60 and m = 82 are used in [15]. The model is made up of three convolutional layers, the first of which contains eight units and is applied to all features in each time step. It is followed by a second convolutional layer that considers three consecutive days at once because it is widely believed that three days would create a trend in the trading markets. It is then flattened into a one-dimensional array and applied to a fully connected layer with sigmoid activation for binary classification before being applied to a max-pooling layer and another convolutional layer. [9] called the mentioned model “2D-CNNpred” and there is a version called “3D-CNNpred.

The goal is to consider not only the numerous elements of a single market index but also to compare it to a variety of other market indices to aid the prediction of a single index. A 2D array is used to represent the data for one market index. A 3D array is created by stacking numerous such data from distinct indices. While the target label remains the same, looking at a new market may provide some additional data to aid prediction. Because the structure of the data changed, the convolutional network had to be re-defined, and the data generators had to be tweaked appropriately.

### 3.2.4 Training, Validation, and Test

[15] employed MAE as the loss metric, as well as accuracy and F1 score, to determine the model's quality. It is worth noting that the F1 score is based on precision and recall ratios, both of which take positive classification into account. The mean of the F1 from positive and negative categorization is used in [15]. It is, explicitly, the F1-macro metric:

The fraction is the number of true positives and false positives with TP and FP precision. Similarly, is the recollection The normal F1 measure, which took positive classifications into account, is the first word in the big parenthesis above. The second term is the inverse, which considers negative classifications. While “sklearn.metrics.f1 score()” is present in sci-kit-learn, there is no equivalent in Keras. As a result, [15] made their own using code from this stack exchange question: <https://datascience.stackexchange.com/questions/45165/how-to-get-accuracy-f1-precision-and-recall-for-a-keras-model>

The training procedure can take several hours. As a result, [15] preserved the model in the middle of the training so that they can stop and restart it later. In Keras, we can employ checkpoint features. They created a checkpoint path filename template and asked Keras to populate it with the epoch number and validation F1 score. We save it by keeping an eye on the validation's F1 measure, which is meant to rise as the model improves. As a result, they provided it with the "mode="max"" parameter. Their model should now be simple to train. There are two things to keep in mind. To the "build()" method, they passed "acc" as the accuracy parameter and the function f1macro as the metrics parameter. As a result, during training, these two parameters will be tracked. Because the function is called f1macro, they call this metric val f1macro in the checkpoint's monitor argument. Separately, in the fit() function, they used the datagen() generator to supply the input data, as described previously. When this function is called, a generator is created, and during the training loop, batches are fetched one by one from it. Similarly, the generator also provides validation data. Because a generator's nature is to send data indefinitely. They need to explain how to define an epoch in the training process. Remember that a batch in Keras is one iteration of gradient descent update. An epoch is a single cycle through all the data in a dataset. Validation should be done after an epoch. It is also a suitable time to run the checkpoint we set up earlier. We must tell Keras how many batches it should process in one epoch using the steps per epoch parameter because it has no method of inferring the size of the dataset from a generator. The validation steps option, on the other hand, indicates how many batches are used in each validation step. The validation has no bearing on the training, but it provided them with the metrics they care about. The measure for the training set is updated every batch, but the one for the validation set is only available at the conclusion, then we evaluate the model using unseen data, or the test set after it has completed training. Rather than generating the test set at random, they deterministically generate it from the dataset. The structure of testgen() is like that of datagen(), which they had previously established. Except in datagen(), the first dimension of the output data is the number of samples in a batch, whereas in testgen() it is the total number of test samples. Because we are employing the sigmoid activation function, utilizing the prediction model will result in a floating point between 0 and 1. Using the 0.5 thresholds, we will convert this to 0 or 1. The accuracy, mean absolute error and F1 score are then computed using scikit-learn methods (which accuracy is just one subtracted by the MAE).

# Chapter 4: Data Analysis and Presentation

## 4.1 System implementation

This system was implemented using a convolutional neural network deep learning method. This method was used because of its demonstrated capabilities in other domains.

### 4.1.1 Packages and tools used:

#### 4.1.1.1 Software tools used are:

1. Google Colab – Used to implement the system and make improvements based on the proposed dataset
2. Python v 3.7 is the programming language used to implement the model. It is an interpreted language that contains multiple libraries.

#### 4.1.1.2 Packages used are:

1. NumPy - This is a library in python which provides support for large multidimensional arrays and matrices and a vast collection of high-level mathematical functions to perform operations on these arrays and matrices.
2. Pandas - This is a library in python that provides the Data Frame object for storing datasets during manipulation it also provides reading & writing capabilities to ".CSV".
3. Scikit-Learn - This is a machine learning library in python that comes bundled with various machine learning algorithms and tools. It also provides data mining tools. It was used to implement Logistic Regression when creating the model.
4. TensorFlow - TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on the training and inference of deep neural networks.

#### 4.1.1.3 System Requirements:

1. RAM – 16GB
2. CPU – intel corei7 7th generation
3. GPU – Nvidia GeForce 10 series/AMD Radeon series
4. Storage – 1TB HDD/256 SSD
5. OS – Windows/Linux/macOS

## 4.2 Results

### 4.2.1 2D Model

Text

Description automatically generated

Figure 4: Visual representation of 2D Model

Figure 5: 2D Model Metrics

### 4.2.2 3D Model

Graphical user interface, text

Description automatically generated

Figure 6: Visual representation of 3D Model

Figure 7: 3D Model Metrics

# Chapter 5: Conclusion

## 5.1 Summary

It is impractical to anticipate a high degree of accuracy, as it is with any financial market forecast projects. Both the 2D and 3D models had and average accuracy of 53% for the USD/NGN currency pair over 10 instances of running the model, but the 3D model had a much better F1 score with 0.6 compared to the 2D model’s 0.4. This is already performing better than some of the literature reviewed. This research may be of some use but there is no magic formula to making money.

## 5.2 Future Improvements

Given the nature of the research and the time allocated to it, I believe given enough time improvements can be made to:

1. The Dataset – More features could be added to improve the prediction model and a larger timeframe of the data can be explored.
2. The model - With enough time there would be ample room for experimentation to get even better results.

# References

|  |  |
| --- | --- |
| [1] | N. Lioudis, S. Anderson and P. Rathburn, "Investopedia," Dotdash Meredith, 21 July 2021. [Online]. Available: https://www.investopedia.com/ask/answers/08/what-is-foreign-exchange.asp#:~:text=Foreign%20exchange%2C%20or%20forex%2C%20is,to%20a%20basket%20of%20currencies.. [Accessed 9 May 2022]. |
| [2] | J. Chen, G. Scott and D. Constagliola, "Investopedia," Dotdash Meredith, 8 December 2021. [Online]. Available: https://www.investopedia.com/articles/forex/11/why-trade-forex.asp. [Accessed 10 June 2022]. |
| [3] | K. Reyes, "Simplilearn," Disqus, 21 February 2022. [Online]. Available: https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-deep-learning. [Accessed 9 May 2022]. |
| [4] | I. C. E. "IBM," 1 May 2020. [Online]. Available: https://www.ibm.com/cloud/learn/deep-learning. [Accessed 11 June 2022]. |
| [5] | "MathWorks," MathWorks inc., [Online]. Available: https://www.mathworks.com/discovery/deep-learning.html. [Accessed 11 June 2022]. |
| [6] | M. Mandal, "Analytics Vidhya," Analytics Vidhya, 1 May 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/. [Accessed 9 May 2022]. |
| [7] | S. Saha, "Towards Data Science," Medium, 15 December 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed 11 June 2022]. |
| [8] | R. Yamashita, M. Nishio, R. Kinh Gian Do and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging,* vol. 9, no. 4, pp. 611-629, 2018. |
| [9] | E. Hoseinzade and S. Haratizadeh, "CNNpred: CNN-based stock market prediction using a diverse set of variables," *Expert Systems with Applications,* vol. 129, no. 0957-4174, pp. 273-285, 2019. |
| [10] | P. Juneja, " Management Study Guide," Management Study Guide, [Online]. Available: https://www.managementstudyguide.com/advantages-and-disadvantages-of-forex-market.htm. [Accessed 9 May 2022]. |
| [11] | O. B. Sezar and A. M. Ozbayoglu, "Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to image conversion approach," *Applied Soft Computing,* vol. 70, no. 1568-4946, pp. 525-538, 2018. |
| [12] | J.-F. Chen, W.-L. Chen, C.-P. Huang, S.-H. Huang and A.-P. Chen, "Financial Time-Series Data Analysis Using Deep Convolutional Neural Networks," *7th International Conference on Cloud Computing and Big Data (CCBD),* pp. 87-92, 2016. |
| [13] | M. Zulqarnain, R. Ghazali, M. Ghouse, Y. Mazwin, H. Mohmad and I. Javid, "Predicting Financial Prices of Stock Market using Recurrent Convolutional Neural Networks," *International Journal of Intelligent Systems and Applications,* 2020. |
| [14] | J. Wang, T. Sun, B. Liu, Y. Cao and D. Wang, " Financial Markets Prediction with Deep Learning," *17th IEEE International Conference on Machine Learning and Applications (ICMLA),* pp. 97-104, 2018. |
| [15] | A. Tam, "Machine Learning Mastery," 15 November 2021. [Online]. Available: machinelearningmastery.com/using-cnn-for-financial-time-series-prediction/. [Accessed 15 June 2022]. |
| [16] | "Tipalti," Tipalti, [Online]. Available: https://tipalti.com/the-top-3-forex-problems/. [Accessed 9 May 2022]. |
| [17] | I. E. T. "Indeed," Indeed, 30 September 2021. [Online]. Available: https://www.indeed.com/career-advice/career-development/what-is-forex-trading. [Accessed 9 May 2022]. |
| [18] | M. Mandal, "Analytics Vidhya," Analytics Vidhya, 1 May 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/. [Accessed 11 June 2022]. |

# APPENDIX