

# **AI/Machine learning approach to identifying potential statistical arbitrage opportunities with FX and Bitcoin Markets**

**Kuselo Ntsika Ntsaluba**

A dissertation submitted to the Faculty of Commerce, University of Cape Town, in partial fulfilment of the requirements for the degree of Master of Philosophy.

January 13, 2019

*MPhil in Financial Technology,  
University of Cape Town.*



The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

# Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the Degree of Master of Philosophy to the University of Cape Town. It has not before been submitted for any degree or examination.

Signed by candidate

---

Kuselo Ntsika Ntsaluba

January 13, 2019

# Abstract

In this study, a methodology is presented where a hybrid system combining an evolutionary algorithm with artificial neural networks (ANNs) is designed to make weekly directional change forecasts on the USD by inferring a prediction using closing spot rates of three currency pairs: EUR/USD, GBP/USD and CHF/USD. The forecasts made by the genetically trained ANN are compared to those made by a new variation of the simple moving average (MA) trading strategy, tailored to the methodology, as well as a random model. The same process is then repeated for the three major cryptocurrencies namely: BTC/USD, ETH/USD and XRP/USD. The overall prediction accuracy, uptrend and downtrend prediction accuracy is analyzed for all three methods within the fiat currency as well as the cryptocurrency contexts. The best models are then evaluated in terms of their ability to convert predictive accuracy to a profitable investment given an initial investment. The best model was found to be the hybrid model on the basis of overall prediction accuracy and accrued returns.

# Acknowledgements

I would like to express my gratitude to my supervisor, Co-Pierre Georg, for his guidance and input. To my parents and siblings, thank you for all you have done for me over my academic career. To my friends, in particular Dumisani Ngwenza, Masego Modibane and Allan Davids, thank you for your unwavering support and advice during this thesis. Your constant support over this period has been invaluable. Finally, praises to the Almighty for the Grace that has cuddled me throughout my academic career.

# Contents

<b>1. Introduction</b>	3
<b>2. Literature Review</b>	5
<b>3. Data Explanation</b>	9
3.1 Data Generation Process	9
3.1.1 Fiat Currency Data	9
3.1.2 Cryptocurrency Data	10
3.2 Necessary Transformations	10
3.2.1 MinMax Normalisation	10
<b>4. Methodology</b>	12
4.1 Definitions	12
4.1.1 Artificial Neural Network (ANN)	12
4.1.2 Genetic Algorithm (GA)	12
4.1.3 Fitness Function	13
4.2 ANN training process using GA	14
4.3 Prediction Methods	15
4.3.1 Moving Average (MA) Strategy	15
4.3.2 Random Simulation	16
4.3.3 ANN Methodology description	16
4.4 Measures of Performance	17
4.4.1 Out-of-Sample Accuracy	17
4.4.2 TPR and TNR	17
4.5 Returns	17
4.5.1 Evaluation of Model Returns	17
4.5.2 Sharpe Ratio	18
<b>5. Data Analysis and discussion</b>	19
5.1 Input Analysis	19
5.1.1 Fiat Currency Analysis	19
5.1.2 Cryptocurrency analysis	20
5.2 Expected output Analysis	21
5.2.1 Fiat currency Output Analysis	21
5.2.2 Cryptocurrency Output Analysis	21

<b>6. Results</b>	23
6.1 Fiat Signal detection Results	24
6.2 Cryptocurrency Signal detection Results	27
6.3 Model Returns Results	30
6.3.1 Further Results discussion	31
<b>7. Conclusion and Future work</b>	34
7.1 Alternative Methodology approach	34
7.2 Concluding Remarks	35
<b>Bibliography</b>	37
<b>A. Appendix</b>	40
A.1 Section of Figures	40

# List of Figures

4.1	<i>Crossover and Mutation operations.</i>	13
4.2	<i>Iterative process of optimizing ANN parameters using GA.</i>	14
5.1	Box-and-Whisker plot for weekly Forex data	19
5.2	Box-and-Whisker plot for weekly Cryptocurrency data	20
6.1	<i>Histograms of Downtrend, Out-of-Sample and Uptrend Prediction accuracies (fiat currency data)</i>	25
6.2	<i>Histograms of Downtrend, Out-of-Sample and Uptrend Prediction accuracies (cryptocurrency data)</i>	28
A.1	Best ANNs \$100 evolution overtime	40
A.2	Best ANNs \$100 evolution overtime for weekly rolling window Min-Max normalised data	41
A.3	Moving Averages \$100 evolution overtime	42
A.4	Best ANNs vs Moving Averages \$100 evolution overtime	43
A.5	Weekly percentage change returns overtime	44
A.6	Weekly Absolute returns overtime	44
A.7	Weekly Absolute returns with 1,2 and 3 std. dev from the mean overtime (Fiat context)	45
A.8	Weekly Absolute returns with 1,2 and 3 std. dev from the mean overtime (Cryptocurrency context)	45

# List of Tables

5.1	<i>Table of weekly Decision distribution for fiat currency data</i>	21
5.2	<i>Table of weekly Decision distribution for Cryptocurrency data</i>	22
6.1	<i>Table of Average Predictions (fiat currency data)</i>	24
6.2	<i>Table of Accuracies SMA</i>	25
6.3	<i>Table of server Specifications</i>	26
6.4	<i>Table of Predictions Best genetically evolved ANNs using whole normalised set</i>	26
6.5	<i>Table of Predictions Best genetically evolved ANNs for data using weekly rolling window normalisation</i>	27
6.6	<i>Table of Average Predictions (Cryptocurrency data)</i>	27
6.7	<i>Table of Accuracies SMA (Cryptocurrency data)</i>	28
6.8	<i>Table of Predictions Best genetically evolved ANNs using whole normalised set</i>	29
6.9	<i>Table of Predictions Best genetically evolved ANNs for data using weekly rolling window normalisation</i>	29
6.10	<i>Table of Accrued returns, Sharpe ratios and Std. Dev % Returns for Best evolved ANNs, \$100 invested (Crypto and fiat currency)</i>	30
6.11	<i>Table of returns for SMA50 and SMA100, \$100 invested (Crypto and fiat context)</i>	30
6.12	<i>Table of standard deviations from the mean absolute return intervals (Fiat data)</i>	32
6.13	<i>Table of standard deviations from the mean absolute return intervals (Cryptocurrency data)</i>	32
6.14	<i>Table of Adjusted Accrued Returns, RWND not included</i>	33

# List of Abbreviations

EUR - Euro

USD - US Dollar

GBP - Great British Pound

CHF - Swiss Franc

BTC - Bitcoin

ETH - Ethereum

XRP - Ripple

RWND - Rolling Window Normalised Data

NN - Neural Network

ANN - Artificial Neural Network

PCI - Price Chart Input

PLI - Price List Input

SMA - Simple Moving Average

MA - Moving Average

TWEANN - Topology and weight evolving artificial neural network

NEAT - NeuroEvolution of Augmented Topologies

GA - Genetic Algorithm

## Chapter 1

# Introduction

Uncertainty is a critical aspect of everyday life and forecasting the unknown has become an issue of great importance in recent times. This is particularly true for the financial sector where its participants are consistently seeking better predictive models in order to maximize returns and outperform the market. Furthermore, where a firm that accounts for a great market share is concerned, the occurrence of wrong forecasts may produce extreme losses which may lead to the possible collapse of a sector. Also, history shows that financial markets are prone to rare events or events that have never previously occurred. For example, the stock market crash of 2008 which led to large and unforeseen market price fluctuations. This provides sufficient evidence to corroborate the vulnerability of financial markets. These market risks are ever present within the sector and must be mitigated. The development of a model with effective and reliable predictions for such a demanding and complex sector is thus no easy task. Therefore, it is important that any recommended models not just possess a high predictive accuracy but also that they adapt well and learn quickly from new unseen market information.

The prediction of financial time-series data is something of great interest at present (Stepnicka, Cortez, Peralta and Stepnicková, 2013). Recent research shows that computationally intelligent (data driven) methods used as prediction models, specifically neural networks and evolutionary algorithms have features that make them outperform existing statistical and mathematical approaches in most cases. For example, a neural network's intrinsic learning ability enables it to capture the relationships within the complex and nonlinear feature space of financial data. Their self-learning and self-organizing properties provide a prediction model that is applicable to most forecasting problems. While in the case of a prediction model based on a mathematical expression a re-calculation would have to take place for every prediction case (Galeshchuk, 2016).

The application of Artificial Neural Networks (ANN) to time-series data was considered by numerous authors, including Giles *et al.* (2001) and Dunis and Williams (2002) just to name a few. A common method that is employed is the Feed Forward Neural Network (FFNN). Networks of this type, although very effective, suffer from a lack of short term memory. Thus resulting in low levels of knowledge preservation. This problem can be avoided by using a Recurrent Neural Network (RNN). However, the RNN has the problem of being hard to train due to its difficulty to employ precise gradient methods during the training phase (Balabanov, Zankinski and Dobrinkova, 2011). A possible solution, proposed by Yao (1999)

and several other authors, is the training of ANN using evolutionary algorithms. Evolutionary algorithms are able to achieve significantly better results during an optimum search in complex multidimensional spaces possessing many local optima, in situations where many gradient-based methods will probably fail to yield the desired results in terms of outputs and outcomes. (Balabanov, Zankinski and Dobrinkova, 2011).

In this dissertation we present a framework characterised by a self-learning data-driven model infers a prediction on the quoted currency USD in weekly timesteps. The model's performance is then compared to a more traditional technical trading strategy as well as a random prediction model. The data-driven model is based on the artificial evolution of neural networks, a hybrid system using both ANN and genetic algorithms, resulting in a Genetically evolved ANN (Stanley and Miikkulainen, 2002). Furthermore, we find that within the methodology, which will be defined, the Genetically evolved ANN performed significantly better than all other models implemented in the fiat currency and cryptocurrency contexts. This conclusion is based on overall prediction accuracy and accrued returns achieved.

The aim of the study is to develop a model that is best able to translate its predictive accuracy, in terms of signal identification, to returns given an initial investment regardless of how the signals were constructed.

The thesis is organized into the following sections: A Literature review is presented in Chapter 2. Chapter 3 presents the construction of datasets as well as all the necessary transformations required. Chapter 4 introduces all the necessary definitions, strategies and algorithms to be implemented based on the methodology. Chapter 5 presents the data analysis and some other insights. Results are analysed in Chapter 6. Finally Chapter 7 concludes the study with recommendations and areas for further in depth research.

## Chapter 2

# Literature Review

This chapter presents the theoretical underpinnings of ANN framework as presented by various academics and practitioners within the international financial markets. The application of an ANN in foreign exchange markets is very rare. This is due to the continued usage of the more mathematical and statistical approaches (Galeshchuk, 2016). These approaches generally have more readily available literature, particularly within financial markets, on how they work and thus are more well known and well understood. They also enable the assessment of expected returns relative to particular levels of risk assuming that the portfolio manager has an adequate understanding of the distribution of the returns (Guresen, Kayakutlu and Daim, 2011). Mathematical and statistical models are statistically well defined in most cases. Notwithstanding this reality, their computational methods are accompanied by inherent underlying assumptions which can make the underlying method inflexible when applied to nonlinear problems. In the work of Whalley *et al.* (1995), the invalidity of the assumptions made in the original Black-Scholes and some exotic options in hedging and transaction costs is expressed. These relate to the assumption of continuous rehedging (given a hedging strategy) and the absence of transaction costs in trading the underlying asset. They state that generally, both assumptions are invalid and depend mainly on the liquidity of the market being considered. As such, the use of assumptions that are generally invalid in reality in turn contributes negatively to the overall predictive ability of the underlining method when applied in reality.

ANN is distinctively a data driven model. Data driven models inherently do not need any assumptions and have the ability to learn from data even in cases of nonlinear data (Galeshchuk, 2016). Guresen *et al.* (2011) asserts that in their implementation of an ANN to stock index prediction that ANNs are one of the best ways to model market value since they do not have standard formulae and can easily adapt to changing market conditions. Narang (2009) shares a concurring view on data driven models and further indicates that compared to their theoretical mathematical counterparts they are better able to discern underlining market behaviour whether it is included under the existing theoretical banner or not. Such an approach leads to discovery of patterns without having to understand the underlying rationale. He adds that by contrast, the theoretical mathematical models capture behaviours that have already been identified and as such their ability is limited. Kou *et al.* (2004) advances further a concurring view regarding the inherent capabilities and abilities of new models developed through data mining stating that

they have the ability to identify new attacks before human experts do.

In the work of [K. V. Prema and Agarwal \(2015\)](#), the model proposed applies Back Propagation on a Multi Layer Perceptron (MLP) network trained using approximately four years worth of BSE Sensex data. Here it was highlighted that the Back Propagation algorithm, when applied by itself in a MLP network, suffers from the possibility of getting stuck in local minima. In an attempt to avoid this, a genetic training is used initially to select the best synaptic weights and node thresholds. After that has been applied, training the MLP using Back Propagation is done. A similar view was stated in the works of [Magnusson and Olsson \(2016\)](#) where they conducted an investigation into investing more computationally intelligent methods within ANN training. A genetic algorithm over a more standard algorithm was used which resulted in higher prediction accuracy while avoiding the local minima problem. The findings reveal that through genetic training, how highly optimised models can be obtained that avoid local minima for more global solutions to a particular problem.

[Brock et al. \(1992\)](#) found nonlinearity in market prices, and their overall results showed strong support for the moving average and trading range break rules as profits generated were statistically significant. The study of [Sullivan et al. \(1999\)](#) expanded on their trading rules and applied them to 100 years of daily data and defined the best moving average (MA) rule as the one which showed the highest cumulative return to date. The signal given by that MA rule would then be followed the next day. Here the MA rule will be used, but implemented with some differences. The major difference being that the decision on whether to buy or sell the underlining asset is not made solely based on a signal from one source but from three.

Efficient market hypothesis (EMH), when defined in its weak form, states that individuals can not earn excess or above average returns by developing strategies based on past market data ([Shazly and Shazly, 1999](#)). However, there are differing degrees of inefficiencies that exist within financial markets. These may be traced using the amount of information contained within the market prices ([Azoff, 1994](#)). The weakness of the EMH assumption has been demonstrated in the work of [Ding et al. \(1993\)](#) where, through an autocorrelation analysis, it was shown that there is a portion of the stock market that is predictable thus showing that EMH assumption does not strictly hold. The work of [Peters \(1994\)](#) also shows the assumption's weakness. Therefore, one could say that these inefficiencies within financial markets show the existence of inherent imperfections within financial markets. Nevertheless, [Chohan \(2017\)](#) shows that there are markets that possess imperfections to a degree not noted in other markets. More specifically, he stated that the Bitcoin market has serious imperfections which are to a degree not noted in other asset classes. Some of these imperfections were highlighted as being:

1. Uninformed investors

2. A weak regulatory presence
3. Illiquidity
4. Market manipulation tactics

Even though, in recent times, there has been a great sell off in the cryptocurrency market, with it losing over 70% of its value in one year, these market imperfections still persist and could affect the accuracy of a model implemented. This market also has the imperfection of a significant number of the major cryptocurrencies still concentrated in the hands of the few.

The highly speculative and volatile nature of the cryptocurrency market is also seen in the work of [Gurdgiev and Corbet \(2018\)](#) where they highlighted how retail investors put billions of their savings into the crypto market, making incredible gains over the period November 2017 to January 2018 only to have over 50% of this new found wealth diminish.

It can therefore be argued and inferred that an increase in imperfections resulting in greater inefficiencies within this market, EMH holds to a lower degree within the cryptocurrency markets than in other markets.

The study presented here makes some unique contributions to existing literature. The unique contributions are enumerated below:

1. Firstly, shows how currency pairs can be used to add meaningfully to making predictions. In doing this, we see that it is possible to construct a model that has an above average prediction accuracy using a relatively small number of currency pairs.
2. Secondly, demonstrates the ability of ANNs to decipher underlining rules and structure between the input and output space. In doing so, it adds to existing literature demonstrating the superiority of the model in comparison to the other models implemented.
3. Thirdly, presents a new variation of the Moving Average strategy that makes its prediction based on more than one source of information or underlining asset. In doing this, an addition is made to existing literature on the implementations of Moving Average Rule.

Furthermore, a demonstration is made of whether the signals given by the best performing models can lead to profitable investment decisions given a particular initial amount investment.

There have been applications for forecasting time series data using neural networks (NNs) such as the work done by [Sher \(2011\)](#), where they analyse the patterns and trends within the currency chart images for trading purposes. Here a Price List Input (PLI) using a directly encoded NN is compared to a Price Chart Input (PCI) using geometrical pattern sensitive NN. The results suggested that a geometrically pattern sensitive NN system generalizes better as well as trades more profitably

than the PLI NN. Furthermore, results showed that a Topology and weight evolving artificial neural network (TWEANN) could effectively evolve currency trading agents. However, the work of [Stanley and Miikkulainen \(2002\)](#) revealed a shortfall with TWEANNs. This work showed that the NeuroEvolution of Augmented Topologies (NEAT) has significant performance advantages over TWEANN since NEAT will only grow network structure when it is required. As such it is likely to introduce unnecessarily complicated network structures. NEAT is a hybrid model combining a neural network with genetic training for a highly optimizer classifier.

Other applications using genetic training for highly optimized classifiers are found in the works of [Huang and Wang \(2006\)](#) as well as [Hulley and Marwala \(2007\)](#). Here support vector machines were optimized resulting in significantly improved classification accuracy using fewer inputs. Lastly, in the work of [Shazly and Lou \(2016\)](#) the forecasting of West Texas Intermediate (WTI) Crude Oil prices is done using a hybrid system combining neural networks with genetic training. The hybrid system outperformed in terms of accuracy as well as correctness.

In this study, we are more concerned with correctness rather than accuracy where correctness is defined similar to [Shazly and Lou \(2016\)](#). Though we are not concerned with accuracy, we are concerned with it within the context of correctness. Models enhanced with genetic training have been shown to yield significantly better results with more generalized solutions within financial markets and outside of them.

This chapter has provided literature that prevails regarding ANN and its utility within the financial markets.

## Chapter 3

# Data Explanation

### 3.1 Data Generation Process

#### 3.1.1 Fiat Currency Data

The data used in this study for the fiat currency data was taken from *Investing.com*. The data is roughly 3 years worth of data, for the period 20 September 2015 - 16 September 2018, on a weekly timeframe for EUR/USD, GBP/USD and CHF/USD. The reason for using a weekly timeframe is that longer timeframes, such as the monthly and weekly, tend to capture the major trend better than the shorter timeframes do. Furthermore, the shorter timeframes tend to be more speculative and volatile in nature. Thus, in an attempt to reduce model exposure to highly speculative markets, all prediction methods applied in this study will be applied on the weekly timeframe. This is done in order to see which of the methods most accurately captures the major trend on a weekly basis.

The full dataset consists of a set of inputs, which are the 3 currency pairs EUR/USD, GBP/USD and CHF/USD. An expected output is derived from these inputs pairs. This expected output was obtained by simply looking at the closing exchange rates for the currency pairs for a particular week and comparing them to those of the next week. Thereafter, using each of the currency pairs individually, a majority decision was taken on the direction of the quoted currency (USD), where the quoted currency is common amongst all 3 fiat currency pairs. The decision taken in a particular week (at time  $t$ ) by a fiat currency pair on the direction of USD using its exchange rate ( $x_t$ ) is decided as follows:

1. If  $x_t < x_{t+1}$  then decision = Sell
2. If  $x_t > x_{t+1}$  then decision = Buy

The decision is then used as the expected output at time  $t$ , which each of the prediction methods in the study have the task of predicting. Since this is time-series data and evaluation of method accuracy needs to be done in a consistent manner, the dataset is divided into a training set (first 80% of data) and a unseen set (last 20% of data).<sup>1</sup> As such, the ultimate goal of each prediction method used in this study is to

---

<sup>1</sup> unseen set is also known as a test set

decipher the inherent nature of this data through how it was constructed to make predictions on the unseen data, which was constructed in the very same way.

### 3.1.2 Cryptocurrency Data

The data used in this section was taken from *Yahoo Finance*. Due to the lack of reliable data within the crypto market as it is relatively new and in an attempt to obtain results that are comparable, data for the period 20 September 2015 - 16 September 2018 will also be used. For the same reasons stated in 3.1.1, the data here will be on a weekly timeframe for three major cryptocurrencies namely BTC/USD, ETH/USD and XRP/USD.

Furthermore, similar to 3.1.1, the 3 major cryptocurrencies BTC/USD, ETH/USD and XRP/USD will be used as inputs and an expected output is derived from them. The expected output is obtained in exactly the same way as in the previous section as well as the majority decision on the direction of the quoted currency (USD).<sup>2</sup> The decision taken for a particular week is decided in exactly the same way as previously stated.<sup>3</sup>

## 3.2 Necessary Transformations

### 3.2.1 MinMax Normalisation

For data driven methods implemented, a transformation of the weekly exchange rates may be necessary in order to remove the numerical effect of the exchange rates with greater numeric ranges dominating those with smaller numeric ranges (Huang and Wang, 2006). In this study the *MinMax Normalization* is used since no models requiring normality of data will be implemented. Value ( $z_t$ ) is the MinMax normalized value used for a particular week (time  $t$ ) and defined as follows:

$$z_t = \frac{x_t - (Min)_r}{(Max)_r - (Min)_r}$$

Where  $(Min)_r$  is the minimum value for a particular exchange rate and  $(Max)_r$  is the maximum value for a particular exchange rate. The *MinMax Normalization* linearly transforms exchange rate ( $x_t$ ) while retaining its original distribution as well as transforming all inputs to a common range of  $[0, 1]$ .

In addition to applying this approach to the whole set in the crypto and fiat contexts, it will also be applied by week in an attempt to get rid of the trends in the inputs. Therefore, this normalisation will be applied on a week by week rolling window and results obtained will be compared.

<sup>2</sup> Note that in this section, the quoted currency is also common amongst the 3 pairs BTC/USD, ETH/USD and XRP/USD

<sup>3</sup> Stated in the Fiat currency data section

---

Generally, this input transformation technique is helpful as it increases the accuracy of the ANNs that are optimized with genetic training according to experimental results. However, it is important to note that there are instances where such a normalisation may not be applicable in time series data. In particular, its applicability depends on knowledge of the maximum and minimum values of the data which are not always possible to attain since future values (or out-of-sample data) may be out of bounds [Ogasawara \*et al.\* \(2010\)](#).

## Chapter 4

# Methodology

### 4.1 Definitions

#### 4.1.1 Artificial Neural Network (ANN)

ANN is a neural network created for prediction as well as classification problems. ANNs are used mostly to model complex functional relationships between inputs and outputs or fitting a particular pattern in a dataset. The use of neurons, dendrites, axons and synapses in an ANN's topology is inspired from the biological central nervous system.

The network, through a process of supervised learning undergoes a process of training until a pre-stated level of fitness is achieved. The learning process relies on an iterative procedure known as back propagation, where inputs are processed through the network, errors are measured using the predictions and expected outputs. Thereafter, weights are adjusted accordingly (Shazly and Lou, 2016). ANNs are built to find associations and formulate relationships to generate predictions that are free from any model constraints (Shazly and Lou, 2016).

ANNs obtained are optimized through the use of an evolutionary algorithm.

#### 4.1.2 Genetic Algorithm (GA)

This is a process that mimics biological evolution. The algorithm is a stochastic search for optimum solutions through repeated iterations of randomly selecting a population and evaluating each of its members according to some fitness function. In the case of a genetic algorithm applied to evolve a Neural Network, the members of the population are referred to as genomes or chromosomes (Stanley and Miikkulainen, 2002).<sup>1</sup> The fitness function relates the optimization problem with the GA and assess the quality of the proposed solution in the evaluation step (Hulley and Marwala, 2007; Huang and Wang, 2006). Once the population is initialized, the following operators are applied: selection, crossover and mutation.

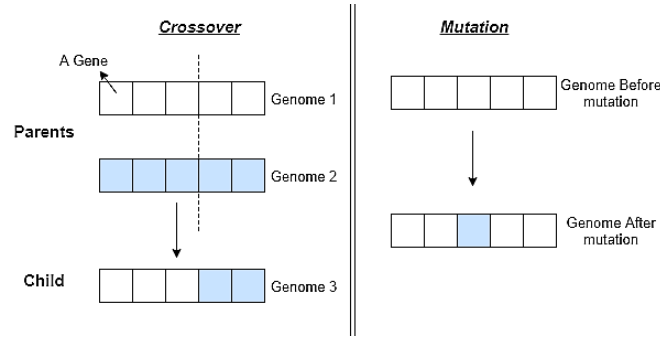
Selection is when the genomes of the population are assigned a probability of survival based on their level of fitness. Crossover is a process of artificial mating requiring two parent networks where some characteristics (neurons or connections)

---

<sup>1</sup> A Genome is a set of genes, where each gene represents a particular aspect (connection or neuron) of the information contained in the Network topology

from each parent are taken to make a child network. Mutation is the random adjustment of a parent network's genetic topology and is used to explore the search space (Shazly and Lou, 2016).

Specifically, GAs belong to a larger class of evolutionary algorithms (EAs) and are used to generate high-quality optimized solutions relying mainly on biologically inspired operators of Selection, Mutation and Crossover. Figure 4.1 illustrates the Crossover and Mutation operators.



**Fig. 4.1:** Crossover and Mutation operations.

The Offspring that result from the operations replace the old population using a diversity replacement strategy forming the new population for the next generation (Huang and Wang, 2006).

### 4.1.3 Fitness Function

Each genome, representing the topological structure of a particular ANN, is evaluated by a particular fitness function. Genomes with high fitness values have a high probability of being preserved to the next generation of networks (Huang and Wang, 2006). The fitness function used in this study is defined as follows:

$$fitness = n_{train} - \sum_{i=1}^{n_{train}} ((Exp)_i - (Pred)_i)^2$$

Where  $n_{train}$  is the number of observations in the training set,  $(Exp)_i$  is the expected value or known correct value for particular set of inputs at  $i$  to the ANN.  $(Pred)_i$  is the prediction made by the ANN for particular set of inputs  $i$  to the ANN. Note also that in order for the specified fitness function to be used effectively, decisions "Buy" and "Sell" will be represented as "1" and "0".

The objective here is to find a solution that maximizes the fitness function specified. It follows that maximizing the fitness is equivalent to minimizing the difference between predicted value  $((Pred)_i)$  and expected value  $((Exp)_i)$  for particular set of

inputs at  $i$ . The desired level of fitness on the training set is specified according to the user's requirements.<sup>2</sup>

## 4.2 ANN training process using GA

We can therefore state the iterative process of optimizing the ANN parameters through genetic training in the following steps:

1. Generation of a random population of  $n$  ANNs represented as genomes.
2. Each genome is evaluated by the fitness function, if the optimal fitness criterion<sup>3</sup> has been reached by any of the genomes in the population, iterative process will terminate and the optimal genome solution will be presented.
3. If fitness criterion has not been reached, genomes with the lowest levels of fitness are discarded whilst those with higher fitness levels are kept for the next generation. The discarded genomes will be replaced through mutation and crossover using the GA (Hulley and Marwala, 2007).
4. A new population of  $n$  genomes is formed. Steps 2 and 3 are repeated until the maximum number of generations<sup>4</sup> is reached or fitness level is obtained.

The specified fitness function is for the iterative process and is of great importance. Its form depends on the problem it is being applied to. It enables the GA to do an effective random search of the space for possible optimized solutions. As such, if the fitness function is incorrectly formulated it could lead to a false indication of model fitness and therefore not reflect a model's true predictive ability.

Figure 4.2 illustrates the iterative process of optimizing the ANN parameters.

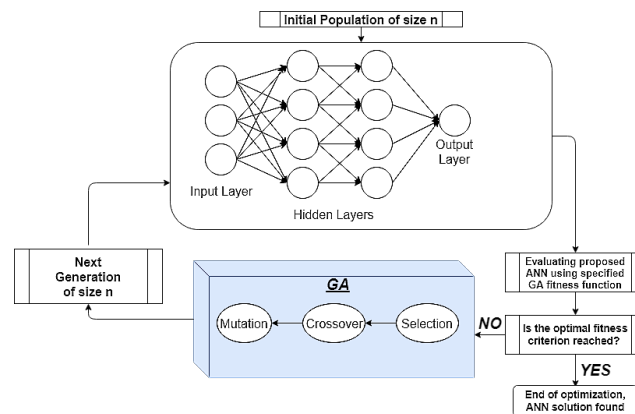


Fig. 4.2: Iterative process of optimizing ANN parameters using GA.

<sup>2</sup> The desired level of fitness on the training set also referred to as the optimal fitness criterion

<sup>3</sup> Specified in the form of a fitness level (e.g. 75% fitness on training set)

<sup>4</sup> This is also user specified

## 4.3 Prediction Methods

### 4.3.1 Moving Average (MA) Strategy

The MA crossover rule is applied with a slight modification to the way that it is applied when used for technical analysis. The rule is believed to be one of the few technical trading rules that is well defined in a statistical sense since its signals are generated based only on information available to date. Normally and in its weak form, the MA crossover rule works on the assumption that sell signals are generated when the price of the underlying asset crosses its moving average from above and buy signals are generated when the price crosses the moving average from the below.<sup>5</sup> The reasoning used for this interpretation is that whenever the price of the underlying asset pierces through the moving average it is believed that a trend has emerged. Now specifically, a bullish trend is said to have emerged when price goes above its moving average and a bearish trend is said to have emerged when price goes below its moving average (Fong and Yong, 2005).

Traditionally, the MA crossover rule is applied to an individual asset's price or currency pair in an attempt to infer a prediction on the underlying. Here a variation of the Simple Moving Average (SMA) rule is applied in an attempt to infer a prediction on the quoted currency using 3 different currency pairs,<sup>6</sup> where the base currencies differ.<sup>7</sup> To the best of the author knowledge, this variation has not previously been applied and is defined as follows:

1. SMA is calculated individually for the 3 different currency pairs at particular points in time to make a prediction of the next timestep.
2. Each of the 3 predictions at a particular timestep is assigned an equal weight.
3. A prediction of the quoted currency is made for the next timestep based on the majority prediction using equal weighting of the 3 different currency pairs. Since the number of currency pairs used to make the prediction is odd (three), there will never be a tie on the number of buys or sells predicted.

The construction of the MA strategy in this manner enables it not just to make a prediction on the quoted currency using a particular pair, but it gives the method an ability to make a more holistic prediction on the direction of the common quoted currency using information from 3 major currency pairs instead of just one. In this study, a short-term SMA (10 day SMA), a medium-term SMA (50 day SMA) and a long-term SMA (100 day SMA) will be used.

---

<sup>5</sup> Note that if the EUR/USD spot rate were to go below its MA, this is a sell signal on the currency pair. However this signal is also equivalent to a buy signal on USD, this is the signal of interest in this study.

<sup>6</sup> SMA is an arithmetic moving average obtained by summing up the closing prices then dividing by the number of time periods in the calculation and is a particular type of MA. There are other types of the MA such as the Exponential Moving Average (EMA)

<sup>7</sup> In the EUR/USD currency pair, EUR is the base currency and USD is the quote currency or the counter currency

### 4.3.2 Random Simulation

This is equivalent to a coin flip of a coin that is balanced. Accept in this instance, the different sides of the coin represent a "sell" and a "buy" prediction. This random model will be used as a base model for prediction with the idea that any proposed model should do better than a model that is completely random.

### 4.3.3 ANN Methodology description

The methodology presented here revolves around the use of an ANN for inferring a prediction on the overall direction of a specific quoted currency on a weekly timestep of data. Three different currency pairs (base currencies differ) are used as inputs to the ANN. The ANN will then make a weekly prediction on the direction of the quoted currency,<sup>8</sup> which is common amongst the three inputs. It has two options to choose from, namely:

1. Buy
2. Sell

The model's predictions will be evaluated on its correctness rather than its accuracy, where accuracy measures how precise the prediction is (e.g. using mean absolute forecast errors), correctness looks at the models ability to predict direction of movement (Shazly and Lou, 2016).

The level of confidence of the ANN in its prediction will not be looked at in this study.

In this study, the performance of the ANN will be compared to the short-term SMA, medium-term SMA and long-term SMA (SMA 100). Their performance will be compared to will be compared to that of the Random Simulation.

To be more specific on the ANN model that will be implemented. A hybrid model that combines ANNs with genetic training optimization, a Genetically Evolved ANN is applied. The reason for this choice is based on the complexity and ever changing behaviour of foreign exchange and crypto markets to move in ways that they have never done previously. One would expect this behaviour to give an ANN problems (decreasing its accuracy) since an ANN would need a specific unseen event to have occurred a particular number of times (i.e. get enough data points) in order to make an accurate prediction on future occurrences. However, in the case of a ANN optimized with an evolutionary algorithm this is believed to not be the case since an optimal search of the ANN topology is done enabling it to adapt to these situations quite effectively. This will be tested.

All prediction methods will be implemented at first using EUR/USD, GBP/USD and CHF/USD as inputs and predicting the USD direction weekly. Thereafter, the same methodology will be repeated in the cryptocurrency data using BTC/USD, ETH/USD and XRP/USD in an attempt to infer a prediction on USD.

---

<sup>8</sup> What is referred to here as predicting the "direction" of the quoted currency is equivalent to predicting the "decision" Buy or Sell that was stated in the Data Generation section 3.1

## 4.4 Measures of Performance

### 4.4.1 Out-of-Sample Accuracy

In this study, the underlying model's accuracy is obtained using the predictions from the test set or unseen data. This will be a good indicator of the underlying model's out-of-sample accuracy.

### 4.4.2 TPR and TNR

Statistical measures of performance, obtained from the Confusion Matrix, used under a binary classification is implemented. Namely, True Positive rate (TPR) and True Negative rate (TNR) also known as Sensitivity and Specificity. Firstly, it is necessary to defined what a *Positive* and a *Negative* is in terms of this study. They are defined as follows:

1. *Positive* is a Buy
2. *Negative* is a Sell

As such, it follows then that a TPR is the number of times a Buy was correctly predicted out of the total number of predictions. Similarly, TNR is the number of times a Sell was correctly predicted out of the total number of predictions. More specifically, TPR and TNR will refer to Uptrend and Downtrend prediction accuracy respectively. As such, the two measures will be used as an indicator of the underlying model's ability to identify an upward or downward movement of the inferred currency under the defined methodology.

For certain prediction methods, such as the Genetically Evolved ANN and the Random Simulation which possess an element of randomness in their application, an average TPR, TNR and Out-of-Sample Accuracy will be obtained through 1000 repetitions of the prediction process to get better understanding of its forecasting ability.<sup>9</sup>

## 4.5 Returns

### 4.5.1 Evaluation of Model Returns

In this section, a methodology is constructed for assessing whether the best performing models are able to translate their superior predictive ability into profitable investment decisions using the signals predicted on a weekly basis in the unseen set and an initial invested amount. The methodology for evaluation of returns is defined as follows:

1. At time  $t$ , if the underlying model makes a correct prediction then the initial investment  $C$  will be increased by  $1 + (EUR/USD_t - EUR/USD_{t-1}) +$

<sup>9</sup> It is important to not that the term "accuracy" used in this research within the context of correctness and as such does not refer to any numeric measure such as mean absolute forecast errors

$(GBP/USD_t - GBP/USD_{t-1}) + (CHF/USD_t - CHF/USD_{t-1})$  meaning that a correct prediction leads to an accrued amount of  $C(1 + (EUR/USD_t - EUR/USD_{t-1}) + (GBP/USD_t - GBP/USD_{t-1}) + (CHF/USD_t - CHF/USD_{t-1}))$ .

2. At time  $t$ , if the underlying model makes an incorrect prediction then the initial investment  $C$  will be decreased by  $1 - (EUR/USD_t - EUR/USD_{t-1}) - (GBP/USD_t - GBP/USD_{t-1}) - (CHF/USD_t - CHF/USD_{t-1})$  meaning that an incorrect prediction leads to an accrued amount of  $C(1 - (EUR/USD_t - EUR/USD_{t-1}) - (GBP/USD_t - GBP/USD_{t-1}) - (CHF/USD_t - CHF/USD_{t-1}))$ .

### 4.5.2 Sharpe Ratio

Normally, better returns on an investment imply that it is a good investment. However, it can happen that very risky investments can generate better returns. It is always better for individuals to seek investments that generate good returns with an optimal or reasonable level of risk. The Sharpe ratio is a method of identifying such investments. A higher ratio implies better investment. The standard deviation of the weekly percentage returns is also presented as it is a measure of risk. Here we define the Sharpe ratio similar to the way it was defined in Carrick (2016). It is defined as follows:

$$SharpeRatio = \frac{returns_{Avg.} - return_{RiskFreeRate}}{std.dev.of\%returns}$$

Where the risk free investment rate of return used here is the US Treasuries. The reason for this choice is that it is a government bond in a currency acknowledged as the world's reserve currency. The  $returns_{Avg.}$  of the underlying model are defined as follows:

1. Looking at each of the weeks in the test set starting from the first, the weekly percentage return is calculated.  
For the first week, the initial investment will be increased by  $[1 + (EUR/USD_t - EUR/USD_{t-1}) + (GBP/USD_t - GBP/USD_{t-1}) + (CHF/USD_t - CHF/USD_{t-1})]$  for a correct week prediction. It is decreased by  $[1 - (EUR/USD_t - EUR/USD_{t-1}) - (GBP/USD_t - GBP/USD_{t-1}) - (CHF/USD_t - CHF/USD_{t-1})]$  for an incorrect week prediction as defined in 4.5.1.  
This continues for the weeks that follow, where at time  $t$  the initial investment is equivalent to the accrued invested amount at time  $t-1$ . Then for the calculation of the weekly percentage return for each week  $t$  the following is used:

$$PercentageReturn = \frac{AccruedAmount_t - AccruedAmount_{t-1}}{AccruedAmount_{t-1}}$$

2. The weekly percentage returns are averaged.

The *std. dev. of the % returns* is simply just the standard deviation of the weekly percentage returns of the underlying model.

## Chapter 5

# Data Analysis and discussion

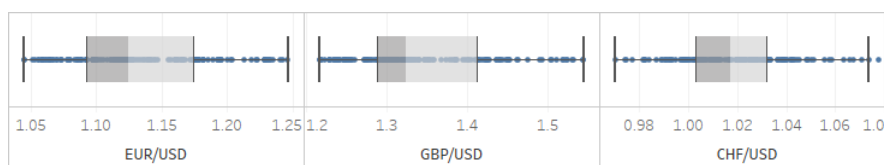
In this chapter a pre-analysis of fiat currency and cryptocurrency datasets is presented. As such, an attempt is made to obtain insight into the characteristics inherently within each dataset given the process used to construct them in 3.1.

## 5.1 Input Analysis

### 5.1.1 Fiat Currency Analysis

Here we take a look at the Box-and-Whisker plots for each of the fiat currency pairs used as inputs. This should give basic insight into the distribution of currency pairs used as inputs, as well as give an indication of whether a *MinMax Normalization* may be required.

Figure 5.1 illustrates Box-plots for the weekly currency pairs EUR/USD, GBP/USD and CHF/USD.



**Fig. 5.1:** Box-and-Whisker plot for weekly Fiat currency data

*This figure shows the distribution of the weekly rates for the different fiat currency pairs*

### Observations made

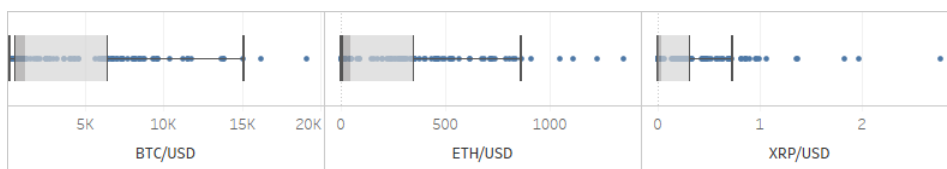
In figure 5.1 we see that both the currency pairs EUR/USD and GBP/USD possess greater numeric ranges than CHF/USD. This could cause numerical difficulties during calculations as well as cause inputs with greater numerical ranges to dominate those with narrower ranges in their contribution to model predictions. This

often increases training time. Furthermore, figure 5.1 also shows that the distribution of CHF/USD is relatively symmetric whereas the distribution of EUR/USD and GBP/USD is more positively skewed.<sup>1</sup>

These points collectively indicate a strong case for the use of normalization, particularly for models whose methods for gathering intelligence relies heavily on computations such as ANNs. We also see the suggestion of a possible outlier in the CHF/USD Box-plot, this observation will not be removed and all prediction methods implemented will work with it.

### 5.1.2 Cryptocurrency analysis

Similar to section 5.1.1, Box-and-Whisker plots will be used to gain insight into the distribution of the three cryptocurrencies used as inputs. Figure 5.2 illustrates Box-plots for the weekly cryptocurrency pairs BTC/USD, ETH/USD and XRP/USD.



**Fig. 5.2:** Box-and-Whisker plot for weekly Cryptocurrency data

*This figure shows the distribution of the weekly rates for the different cryptocurrency pairs*

#### Observations made

In figure 5.2 we see a situation that is somewhat similar to that in figure 5.1 but more extreme. Firstly, the distributions of BTC/USD, ETH/USD and XRP/USD are all positively skewed. Secondly, ETH/USD and XRP/USD in particular, possess a significant number of outliers. This is expected given the high levels of volatility within crypto markets which may increase the likelihood of extreme observations in data. Lastly, similar to the fiat currency analysis section 5.1.1, the numeric ranges of BTC/USD, ETH/USD and XRP/USD differ greatly.


The points stated here indicate that the need for normalization in the cryptocurrencies data is greater than in the fiat currency data and must be carried out in order to prevent inputs with greater numerical ranges from dominating predictions. Furthermore, since the outliers account for a significant amount of the data they cannot be removed. They will be considered as not being outliers and all prediction models used will work with them. This will also provide some insight into how robust any suggested model is in the presents of outliers.

<sup>1</sup> Also know as Skewed Right

## 5.2 Expected output Analysis

### 5.2.1 Fiat currency Output Analysis

In this section we take a look at how the expected outputs,<sup>2</sup> which will be predicted, are distributed over the respective years for the period *20 September 2015 - 16 September 2018*. The distribution of weekly decisions in the dataset is shown in table 5.1.

Decision	Year of Date				Grand Total	Number of Records
	2015	2016	2017	2018		
buy	7	29	24	22	82	
sell	8	23	29	15	75	
Grand Total	15	52	53	37	157	

**Tab. 5.1:** Table of weekly Decision distribution for fiat currency data

#### Yearly observations

In table 5.1 we see that the years 2015 and 2018 will have significantly less data that is contributed to any data driven model that is implemented. As such, most of the intelligence that is gathered during computational processes is expected to come mostly from the weeks within the years of 2016 and 2017. Note also that 2017 had an extra week therefore contributing one extra weekly observation.

#### Decision analysis

The decisions shown in table 5.1 reveal that there are more "buy" than "sell" signals overall within the data.

In particular, for the years 2016 and 2018, the "buy" signals were significantly more than the "sell" signals. These margins are greater than any other year when "sell" signals exceeded "buy" signals.

### 5.2.2 Cryptocurrency Output Analysis

Similarly to section 5.2.1, the expected output distribution over the years within the period *20 September 2015 - 16 September 2018* is shown. The distribution of weekly decisions in the Cryptocurrency dataset is shown in table 5.2.

#### Yearly observations

In the analysis of table 5.2 we find that in this case the years 2015 and 2018 have less data than the other years considered. We also see here that 2017 had an extra

<sup>2</sup> Also known as the decisions explained in the Data Generation process

Decision	Year of Date				Grand Total	Number of Records	
	2015	2016	2017	2018		5	32
buy	5	26	21	21	73	5	32
sell	10	26	32	16	84		
Grand Total	15	52	53	37	157		

**Tab. 5.2:** Table of weekly Decision distribution for Cryptocurrency data

week.

### Decision analysis

When evaluating decisions in table 5.2 it is revealed that, contrary to the fiat currency output analysis 5.2.1, this dataset has significantly more "sell" than "buy" signals overall. In particular, for the year 2017 there is a much greater amount of "sell" than "buy" signals, and "sell" signals were double the number of "buy" signals for the period in 2015. Lastly, there is an equal amount of "buy" and "sell" signals in the year 2016 which was not observed in the fiat currency analysis 5.2.1.

## Chapter 6

# Results

In this chapter we analyse the test results obtained in the implementation of each prediction model. We analyse the results within the fiat currency as well as the cryptocurrency context. Furthermore, each model is discussed in terms of its Uptrend, Downtrend and Out-of-Sample prediction accuracies respectively. As stated in previous chapters, the Uptrend and Downtrend prediction accuracies refers to the particular model's ability to correctly predict change of direction with respect to the signals "Buy" and "Sell" inferred on the quoted currency USD. Out-of-Sample prediction Accuracy refers to overall predictive ability on the unseen set.

In both the fiat currency and cryptocurrency sections a sigmoid activation function is used for the implemented ANN optimised through a genetic training. It possesses many desirable qualities, one being that the function and its derivatives are continuous everywhere (Shazly and Lou, 2016), it also gives the proposed ANN the ability to estimate nonlinear functions. The function is also symmetric and in line with the suggestions of Refenes *et al.* (1993) who showed that symmetric functions yield faster convergence and a better generalised performance. Firstly, we state some experimental insights which give some justification for the chosen approach.

**Experimental insights** During the process of trial and error in the experimental stages, an original split of 80% training and 20% unseen was attempted. The findings showed that this split put the ANN optimised with genetic training at a significant disadvantage to the other implemented models. In particular, the larger unseen set brought about a greater prediction horizon. This in turn means that an ANN model, trained and optimised to a specific fitness on the training set, would then make predictions on all the weeks in the unseen set using only knowledge obtained from a static training set. However, this is not the true for other prediction methods implemented such as the SMA where it uses all previously available needed information to make a prediction for the next time step. Then at the next time step, it will use all available needed information at that time step and so on. This gives the SMA the ability to take into account the latest and older market data when making its forecasts. As such, the SMA is in a process of continuous learning whereas the ANN optimised with a GA no longer learns after training and relies only on the knowledge obtained in the training set. Therefore, in an attempt to put models on a more equal footing the following options were identified:

1. A split of 90% training and 10% unseen can be used in order to compensate

for models that are not continuously learning.

2. Implement a genetically optimised ANN using a moving training set approach. In this approach, after each weekly prediction on the unseen data, the observed values form part of the training set. Thereafter, using all the data available till that week, retraining through the genetic algorithm takes place to a specified fitness level before the next prediction takes place and so on.

In this study, a moving training set approach is implemented since this would enable the genetically optimised ANN to continuously learn from new market information thus increasing its forecasting ability. This is also believed to be the best approach since it would mirror an asset manager's approach.

## 6.1 Fiat Signal detection Results

The data set of weekly observations was for the complete period of 20 September 2015 - 16 September 2018 and was divided into two subsets: training set period 20 September 2015 - 21 January 2018 and unseen set 28 January 2018 - 16 September 2018. The training and unseen sets therefore consisted of 123 and 34 weekly observations respectively, which is sufficiently close to the 80/20 split suggested in earlier chapters. Note also that since this is timeseries data data-points are indexed in order of time, therefore the unseen set's weekly observations are strictly after those of the training set and all implemented prediction models should take this into account.

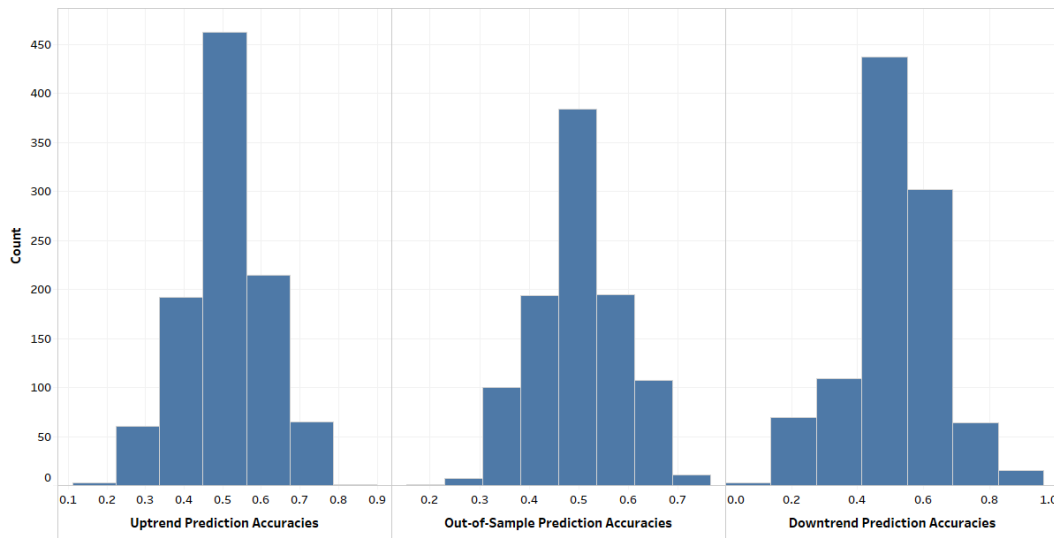
### Random simulation on Fiat

Here the results obtained from the Random simulation are presented. Since the method inherently random, it is important for a general illustration of its predictive capabilities to be presented. This is what figure 6.1 illustrates through the use of histograms showing the Downtrend, Uptrend and Out-of-Sample Prediction Accuracies for 1000 repetitions of the method on the unseen set. Table 6.1 shows the Average Downtrend, Uptrend and Out-of-Sample Prediction Accuracy.

Avg. Uptrend Prediction Accuracy	0.5018
Avg. Out-of-Sample Prediction Accuracy	0.5014
Avg. Downtrend Prediction Accuracy	0.5008

**Tab. 6.1:** Table of Average Predictions (fiat currency data)

We see from figure 6.1 that the Downtrend, Uptrend and Out-of-Sample Prediction Accuracies generally exhibit the shape of a normal distribution centered around 50%. This suggests that most random models recorded over 1000 repetitions obtained accuracies of around 50%, particularly in the case of Downtrend Prediction Accuracies where almost half of all random models had an accuracy of around 50%. The averages seen in table 6.1 are also about 50% and therefore are in



**Fig. 6.1:** Histograms of Downtrend, Out-of-Sample and Uptrend Prediction accuracies (fiat currency data)

line with the observations made from the histograms where there is a high concentration of accuracies around 50%. This all indicates that the model generally has an average predictive ability with respect to detecting “Bull” and “Sell” signals. Any recommended model should at the very least be able to show a predictive ability that is better than the random model.

### Moving Average strategy on Fiat

Now we present the results attained through the application of the SMA strategy that was introduced in 4.3.1. Table 6.2 shows the results for the implementation of the short-term SMA (10 day SMA), a medium-term SMA (50 day SMA) and a long-term SMA (100 day SMA).

	10 day SMA	50 day SMA	100 day SMA
Uptrend Prediction Accuracy	0.5455	0.6364	0.5455
Out-of-Sample Prediction Accuracy	0.4412	0.50	0.4412
Downtrend Prediction Accuracy	0.25	0.25	0.25

**Tab. 6.2:** Table of Accuracies SMA

The results in table 6.2 show that the Moving Average strategy was able to outperform the random model generally on detection of “Buy” signals however they do not do so well in the identification of “Sell” signal on all the SMAs implemented. Furthermore, when looking at the Out-of-Sample Prediction Accuracy we find that generally the overall predictive ability of the MA strategy is not better than that

of the Random Model in this market under the methodology used to construct the signals.

### Genetically Evolved ANN on fiat

An evaluation of the results obtained through the implementation of the ANN optimised through genetic training is done in this section. The process of a genetic algorithm optimising suggested ANN configurations to a specified level of fitness over a maximum of 1000 generations was repeated 100 times.

Since the genetic training is also implemented within the context of a moving training set, this implies that the data available at each of the weeks in the unseen data differs. Therefore, it is highly likely that ANNs suggested at each week  $t$  will differ in structure from those suggested in the earlier weeks.<sup>1</sup> As such, it is more correct to say that the best genetically evolved ANNs results reported here refers to the best combination of ANNs found during the 100 repetitions of this process on the entire unseen set where the returns generated were highest. But in this section we look at the Uptrend, Downtrend and Out-of-Sample prediction accuracies.

A fitness level of 65% was used.<sup>2</sup> In table 6.3 the specs of the server partition used to perform the repetitions of genetic training process as well as an estimation of the average time it took for the entire process to complete are shown. We depict the Downtrend, Uptrend and Out-of-Sample Prediction Accuracy in table 6.4. Table 6.5 shows the same but for the data normalised using weekly rolling window.

CPU	Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz
System Memory	5949MiB
Avg. Time to Complete Optimisation	8 hours

**Tab. 6.3:** Table of server Specifications

Uptrend Prediction Accuracy	0.9027
Out-of-Sample Prediction Accuracy	0.5955
Downtrend Prediction Accuracy	0.0993

**Tab. 6.4:** Table of Predictions Best genetically evolved ANNs using whole normalised set

When looking at the results provided in table 6.4, we find that the model was significantly better at detecting "Buy" than "Sell" signals. This finding is not shocking since most of the signals that were found in the data were "Buy" signals as was shown in table 5.1. However, the percentage difference in ability to identify "Buy"

<sup>1</sup> Structure here refers to the synaptic weights, hidden layers and number of neurons

<sup>2</sup> A fitness level of 65% means that during the optimisation in the training process at each week  $t$  in the unseen set, the moment 65% of training set observations are classified correctly the process terminates and the ANN meeting the fitness level is used to make the next prediction.

Uptrend Prediction Accuracy	0.8372
Out-of-Sample Prediction Accuracy	0.6096
Downtrend Prediction Accuracy	0.242

**Tab. 6.5:** *Table of Predictions Best genetically evolved ANNs for data using weekly rolling window normalisation*

and "Sell" signals is much greater than the difference in the number of "Buy" and "Sell" signals as shown in table 5.1. Additionally, the results obtained using a week by week rolling window normalisation <sup>3</sup> in table 6.5 shows similar results to table 6.4 in addition to a slightly better ability to identify "Sell" signals.

We also see an overall prediction accuracy in both tables 6.4 and 6.5 suggesting that this model generally outperforms all the models previously implemented.

## 6.2 Cryptocurrency Signal detection Results

In this section the data set of weekly observations was for the complete period 20 September 2015 - 16 September 2018 and similarly to the previous section 6.1 two subsets were formed: training set period 20 September 2015 - 21 January 2018 and unseen set 28 January 2018 - 16 September 2018. Furthermore, the training and unseen sets therefore consisted of 123 and 34 weekly observations respectively. Note also that, similarly to the 6.1, all predictions made on the unseen set are for weeks within 2018.

### Random simulation on crypto

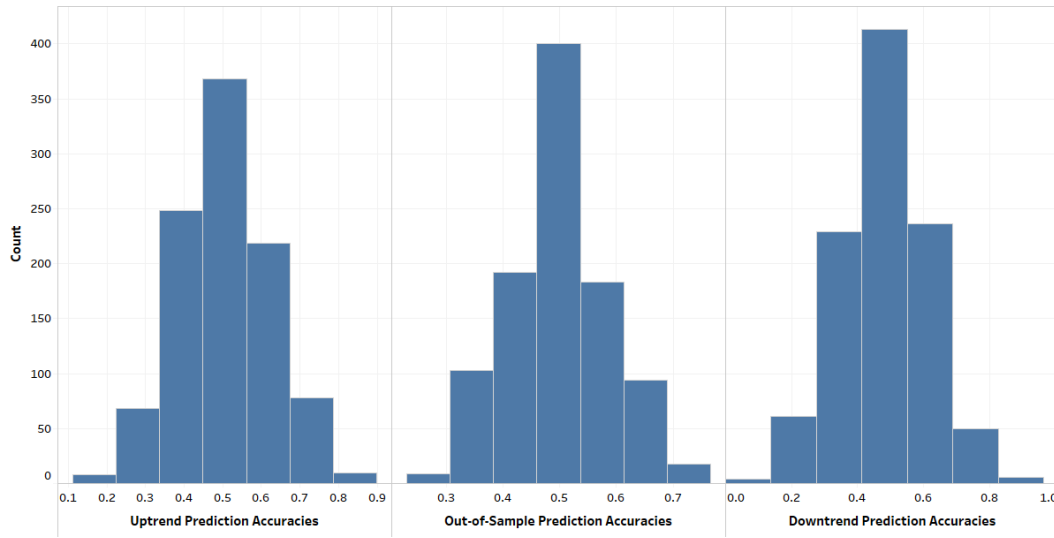
Here, similarly to the section 6.1, we present the results for the Random simulation in the cryptocurrency context. Figure 6.2 shows histograms of Downtrend, Uptrend and Out-of-Sample Prediction Accuracies for 1000 repetitions of the method on the unseen set and the averages are in table 6.6.

Avg. Uptrend Prediction Accuracy	0.5010
Avg. Out-of-Sample Prediction Accuracy	0.5002
Avg. Downtrend Prediction Accuracy	0.4992

**Tab. 6.6:** *Table of Average Predictions (Cryptocurrency data)*

We see in the histograms of figure 6.2 that Downtrend, Uptrend and Out-of-Sample Prediction Accuracies exhibit a bell-shape similar to those in 6.1. Overall, the findings here are very similar to those found for 6.1 and similarly any recorded model should at the very least perform better than the random model.

<sup>3</sup> See section 3.2.1



**Fig. 6.2:** Histograms of Downtrend, Out-of-Sample and Uptrend Prediction accuracies (cryptocurrency data)

### Moving Average Strategy on crypto

The prediction results for the MA strategy within the cryptocurrency context are present here. Table 6.7 shows the results for the implemented short-term, medium-term and long-term SMA. When assessing the results in table 6.7 we observe a

	10 day SMA	50 day SMA	100 day SMA
Uptrend Prediction Accuracy	0.5263	0.8421	0.8421
Out-of-Sample Prediction Accuracy	0.4706	0.50	0.5294
Downtrend Prediction Accuracy	0.40	0.0667	0.1333

**Tab. 6.7:** Table of Accuracies SMA (Cryptocurrency data)

situation that differs from that observed in the 6.1 implementation of the method. Specifically, we find a slightly better overall predictive ability of the models when implemented in this context particularly for the 100 SMA. Although this overall predictive performance is not significantly better than the average ability of the random model, we see that it is better than the shorter term moving averages. This can be attributed to the fact that the shorter term moving averages have a shorter time span of data that is used to calculate the arithmetic averages and so it is more susceptible to false signals in highly volatile markets even though it is able to adapt quicker to new market developments.<sup>4</sup>

Furthermore, table 6.7 shows a generally better predictive ability in the identification of "buy" over "sell" signals.

<sup>4</sup> Highly volatile markets such as the Cryptocurrency markets

### Genetically Evolved ANN on crypto

Similarly to 6.1, here we report the results attained through the implementation of the genetic enhanced ANN model within the cryptocurrency context. The model setup in terms of the maximum number of generations and the number of repetitions is exactly the same as in 6.1. Furthermore, the same fitness level of 65% was used and the whole process to an **average time of 12 hours** to complete. Here best genetically evolved ANNs results reported also refers to the best combination of ANNs found during the 100 repetitions of this process on the entire unseen set where the returns generated were highest.

Uptrend Prediction Accuracy	0.7607
Out-of-Sample Prediction Accuracy	0.5427
Downtrend Prediction Accuracy	0.2667

**Tab. 6.8:** *Table of Predictions Best genetically evolved ANNs using whole normalised set*

Uptrend Prediction Accuracy	0.4853
Out-of-Sample Prediction Accuracy	0.4788
Downtrend Prediction Accuracy	0.4704

**Tab. 6.9:** *Table of Predictions Best genetically evolved ANNs for data using weekly rolling window normalisation*

When evaluating the results presented in 6.8, we see that the implemented model is better at identifying "Buy" than "Sell" signals. However, we see that in this context we see that the percentage difference between the ability to identify "Buy" and "Sell" signals is a lot narrower than in 6.1.

This is not a shocking finding for the following reasons:

1. Since most of the signals within this data were "Sell" signals as was shown in table 5.2 so the ANNs suggested in this context would have an inherently better ability to identify "Sell" signals than those in the fiat context.
2. The year 2018, which contains all of the unseen set, has more "Buy" than "Sell" signals. As such, even if the an ANN was trained initially with data consisting mostly of sell signals, it is continuously learning. Therefore this implies that it would get better at identifying "Buy" signals as it progresses through the weeks in the unseen set.

The accuracies in table 6.9 using a week by week rolling window normalised set suggest that the model does not do as well in signal identification. However, it does do better at "Sell" signal detection.

Furthermore, looking at the average accuracies in tables 6.8 and 6.9 we see the overall prediction accuracy suggesting that this model generally under-performs in comparison to its previous implementation in 6.1.

### 6.3 Model Returns Results

In this section, an attempt is made to assess whether the best performing models namely the 50 SMA, 100 SMA and the genetically optimized models are able to translate their predictive ability into profitable investment decisions using the methodology defined under 4.5.1 with an initial investment amount of \$100. We also present the Sharpe ratios and standard deviations.

However, the returns for the random model will not be considered since the returns can not be attributed to a particular methodology because the model is equivalent to flipping a balance coin.

Table 6.11 shows the returns, sharpe ratios and standard deviations obtained for the 50 SMA and 100 SMA. Similarly, table 6.10 shows the returns, sharpe ratios and standard deviations for the best ANNs obtained.

<b>Best Genetically Evolved ANNs (crypto context)</b>		
<b>Returns generated</b>	<b>Sharpe Ratio</b>	<b>Std. Dev % Returns</b>
\$3755.12	0.42595	0.36623
<b>Returns generated (RWND)</b>	<b>Sharpe Ratio(RWND)</b>	<b>Std. Dev % Returns(RWND)</b>
\$1351.85	0.34927	0.37681
<b>Best Genetically Evolved ANNs (Fiat currency context)</b>		
<b>Returns generated</b>	<b>Sharpe Ratio</b>	<b>Std. Dev % Returns</b>
\$1144.90	0.34484	0.43115
<b>Returns generated (RWND)</b>	<b>Sharpe Ratio(RWND)</b>	<b>Std. Dev % Returns(RWND)</b>
\$1306.77	0.3784	0.4261

**Tab. 6.10:** Table of Accrued returns, Sharpe ratios and Std. Dev % Returns for Best evolved ANNs, \$100 invested (Crypto and fiat currency)

<b>Moving Averages (crypto context)</b>			
	<b>Returns generated</b>	<b>Sharpe Ratio</b>	<b>Std. Dev % Returns</b>
<b>SMA50:</b>	\$36.54	-0.1376	0.1748
<b>SMA100:</b>	\$47.75	-0.1403	0.1747
<b>Moving Averages (Fiat currency context)</b>			
	<b>Returns generated</b>	<b>Sharpe Ratio</b>	<b>Std. Dev % Returns</b>
<b>SMA50:</b>	\$28.24	-0.2976	0.1418
<b>SMA100:</b>	\$37.63	-0.2376	0.1430

**Tab. 6.11:** Table of returns for SMA50 and SMA100, \$100 invested (Crypto and fiat context)

Looking at the results in tables 6.10 and 6.11 we see that not only do the best genetically optimized ANNs generate generally higher returns for the test set period that is common amongst all models implemented, they also have generally higher

Sharpe ratios. This implies that the genetically trained ANNs takes much less of a risk to generate the same level of unit returns as the Moving Average models. Also, there are significant losses on the initial investment made in the implementation of the Moving Average.

Furthermore, when comparing the standard deviation of the percentage returns in the crypto and fiat currency context for the Moving Averages we see that returns/losses obtained in the fiat context were much more consistent than in the crypto context (due to a standard deviation that is less than half the other).

We can also see that the even though the genetically optimized ANNs standard deviations are higher implying greater risk, you are definitely compensated for the level of risk taken by higher unit returns looking at the Sharpe ratios. As such, the genetically evolved model is the best choice within both contexts. Additionally, table 6.10 shows that the rolling window normalised data (RWND) gives better returns and a higher sharpe ratio in the fiat context but this is not the case in the crypto context. This confirms that the removal the trend in the inputs through weekly Min Max normalisation led to increased ability to convert predictive ability to a profitable investment in the fiat context. However, it did not do so well at this in the crypto context.

The finding that the standard deviation of the percentage returns in the fiat context is higher than that for the crypto context in the case of the genetically optimized ANNs was a strange finding and not in line with expectations. Furthermore, this finding tells us that the returns achieved in the crypto context are more consistent than in the fiat context for the genetically optimised models.

However, these findings can possibly be justified by figure A.5 which shows the weekly percentage changes in returns overtime. Here we see that there seems to be more deviations from the mean percentage change of returns in the fiat context. As such the standard deviations figures shown in the table 6.10 do actually make sense.

The above results also show that none of the models implemented have a Sharpe ratio of 1 or greater therefore they are not fairly consistent performers. But the genetically optimized models do have a Sharpe ratio greater than zero showing that they may be better investments than the risk-free rate.

Furthermore, we see that even though the best genetically evolved ANNs in the crypto context had a weak overall signal identification ability in comparison to fiat, they are better able to translate predictions into profitable investment decisions. However, the Moving Average methods were the worst performers at this objective.

### 6.3.1 Further Results discussion

The cryptocurrency market, in addition to having weak regulatory oversight has significantly less investors than the foreign exchange market. Furthermore, Chohan (2017) highlighted an imperfection of this market as having a significant amount of the major cryptocurrencies concentrated in the hands of the few. As such, the crypto market, in addition to being significantly smaller, is heavily monopolised.

Therefore, the likelihood of a practitioner earning excessively high returns is very likely but the likelihood of losing you investment is almost equally as high if not higher particularly if you are an early adopter.

However, the foreign exchange market, in addition to having high levels of regulatory oversight, is significantly less monopolised. Therefore, the market is significantly more resistant to the views of a few major stack holders as well as possessing more informed investors. As such, the market is more efficient and fair to its practitioners. It can thus be said that it is possible for a market practitioner to earn excessively high returns or losses but this is significantly less likely than in the crypto market.

Now, we attempt to contextualise the returns obtained in the crypto and fiat contexts for the best performing model, the genetically enhanced ANN. Firstly, we define weakly the Empirical Rule. In the application of the Empirical rule, **1 standard deviation from the mean absolute weekly return** contains **approximately 68.2%** of the absolute weekly returns generated, **2 standard deviations** contains **approximately 95%** and **3 standard deviations 99.7%** of the absolute weekly returns generated.

The Empirical Rule assumes normality in of the underlying. This is not the case, however it is a good way to see how likely the weekly absolute returns obtained are. More specifically, the rule will be used to identify abnormal weekly absolute returns realised by comparing them to how the other weekly absolute returns in the accrued amount behave. The abnormal weekly returns realised in the accrued amount will be referred to as outliers.

Table 6.12 and 6.13 shows the intervals obtained using standard deviations from the mean absolute return.

1 standard deviation from mean	(-116.461 ; 205.153)
2 standard deviation from mean	(-277.268 ; 365.960)
3 standard deviation from mean	(-438.076 ; 526.768)

**Tab. 6.12:** Table of standard deviations from the mean absolute return intervals (Fiat data)

1 standard deviation from mean	(-403.406 ; 590.096)
2 standard deviation from mean	(-900.157 ; 1086.847)
3 standard deviation from mean	(-1396.907 ; 1583.597)

**Tab. 6.13:** Table of standard deviations from the mean absolute return intervals (Cryptocurrency data)

Looking at the weekly absolute returns plot figure A.8 in the appendix, we see the possible suggestion of an outlier around the 31<sup>st</sup> of August 2018 also suggesting

that the absolute return achieved for that week is an outlier. This is highlighted by the fact that most of the weekly absolute returns are contained within 2 standard deviations thus this absolute return is abnormal. Similarly, there are also two suggested outliers in the fiat context as shown in figure A.7.

The implication here is that the weekly absolute returns obtained and therefore the accrued returns are possible but not to the amounts stated in table 6.10. Furthermore, the outliers highlighted in both figure A.8 and A.7 represent the greatest weekly absolute returns obtained. Therefore their removal will adjust the accrued amounts in table 6.10. A more realist reflection of the accrued returns that can be obtained is shown in table 6.14:

Returns generated (Best Genetically Evolved ANNs, Fiat context)	\$366.42
Returns generated (Best Genetically Evolved ANNs, Crypto context)	\$2534.50

**Tab. 6.14:** *Table of Adjusted Accrued Returns, RWND not included*

The RWND was not included in table 6.14 because the method did not yield much benefit in terms of the accrued returns observed looking at table 6.10. As shown in table 6.14 we see that although profits are still obtained they are significantly reduced.

## Chapter 7

# Conclusion and Future work

### 7.1 Alternative Methodology approach

The methodology presented in this dissertation, revolving around the use of a ANN obtained through the use of an evolutionary algorithm, could be presented alternatively in the manner stated in this section. Where the ANN optimised through using a GA infers a prediction on the underlining common quoted currency where the base currencies for all 3 input currency pairs is different. Here the ANN would have four options to choose from, namely:

1. Sell
2. Buy
3. Strong Sell
4. Strong Buy

Model predictions will still be evaluated on correctness rather than accuracy. The distinction made between "Sell" and "Strong Sell", or "Buy" and "Strong Buy" is just based on the level of confidence that is to be used to figure out how confident the ANN is in its response. In a case where the Expected Output for a particular week is "Strong Sell" but the ANN predicts "Sell", the ANN is correct in the direction chosen, however, it lacks accuracy since it's not completely correct in terms of its prediction. Note also that this particular formulation of the methodology would require a new fitness function that takes into account the added complexity of the new options "Strong Sell" and "Strong Buy" that were introduced.

Furthermore, the alternative approach used here makes it easier to determine how profitable the use of the Genetically Evolved ANN is in comparison to the other strategies. By simply assigning a particular quantity (or scalar multiple) to the level of confidence of its prediction combine with the model accuracy at each timestep, the models profitability can be quantified. However, this change of methodology would most likely decrease the accuracy of the random simulation since the changes of it guessing correctly will be halved. But this is a subject for future study.

## 7.2 Concluding Remarks

A methodology has been presented where the models implemented were given the task of making weekly predictions on the currency and cryptocurrency unseen sets. It has also shown that, the proposed Genetically Evolved ANN generally outperforms a random chance model equivalent to a coin flipping strategy and the more established trading method SMA on the basis of overall prediction accuracy as well as accrued returns. In addition to this, we found that the genetically optimised ANNs generate an higher average percentage return that is in excess of the risk free rate per unit of overall risk taken through the sharpe ratios reported. However, the following criticism may be stated on the methods implemented:

1. the new variation of the SMA strategy implemented here differs from its original implementation and as such could have significantly affected its performance;
2. the methodology used to construct the signals where a prediction was inferred on the quoted currency is hypothetical. Thus the models used cannot be implemented in practice in their current form. This implies that the returns obtained are not absolute;
3. the reliability of the methodology, in how signals were constructed, has not been verified for obtaining consistent results in differing data periods and other markets;
4. the models would need to be modified to account for other factors (e.g. transaction costs).

Although these views may be correct and bring to light issues for further study, the study presented here does show that even though the SMA is in a continuous process of learning through how it is defined, the method of learning it uses is inferior to the more computationally intelligent Genetically Evolved ANN. In fact, the implementation of a the data driven ANN with continuous learning coupled with computationally intelligent methods such as genetic training gives the Genetically Evolved ANNs much greater forecasting ability as well as more robustness. A similar approach is used in [Refenes \*et al.\* \(1993\)](#) with their implementation of a single-step prediction model. This would give it a chance to self correct any mistakes that it has made and remove any knowledge previously acquired that is no longer applicable which should increase accuracies reported. However, this is a more computationally intensive process, increasing the size of the training set at each timestep has major implications for training time since it is highly likely to take much longer to reach the desired fitness levels. It must also be stated that the fitness levels used in this study were obtained through trail and error, but through the use more sophisticated method the accuracies reported could be improved. However, this is also a subject for further study.

Additionally, a truly great predictive model in this study, should be able to show good predictive ability regardless of how the signals or output was constructed

from the data. The ANN optimised with genetic training has shown this ability through how it deciphered the relations between the input space and expected outputs within the respective datasets. Thus obtaining characteristics inherent within the data much better than the other prediction methods implemented.

The finding that the standard deviation of the weekly percentage returns in the crypto context was lower than in the fiat was an unexpected finding. However, this could have been a finding that was specific to the test set period used in this study and would probably change if applied on different historical data. This is also a subject for further study.

It has also been demonstrated through the genetically trained ANNs that superior predictive ability in a model does not necessarily translate into superior investment returns given an initial particular amount invested. This was seen through how the model was able to better identify the signals in the fiat context but it obtained superior returns in the crypto context. In light of this, the model best able to translate predictive ability into investment returns was the genetically evolved ANNs.

Lastly, the sell off in cryptocurrencies, where they lost over 70% of their value within a year, could also be an indication that the investors involved have come to realise that the market has great prospects, but unfortunately these have not materialised as yet. We might just see the crypto market climb to highs even greater than those previously seen if these prospects become a reality.

# Bibliography

- Azoff, E. M. (1994). *Neural Network Time Series Forecasting of Financial Markets*, John Wiley & Sons.
- Balabanov, T., Zankinski, I. and Dobrinkova, N. (2011). Time series prediction by artificial neural networks and differential evolution in distributed environment, *Springer* pp. 198–205.
- Brock, W., Lakonishok, J. and LeBaron, B. (1992). Simple technical trading rules and the stochastic properties of stock returns, *Journal of Finance* **47**: 1731–1764.
- Carrick, J. (2016). Bitcoin as a complement to emerging market currencies, *Emerging Markets Finance and Trade* .
- Chohan, U. W. (2017). Bitcoins and bank runs: Analysis of market imperfections and investor hysteresis, *SSRN* .
- Ding, Z., Granger, C. W. J. and Engle, R. F. (1993). A long memory property of stock market returns, *Journal of Empirical Finance* pp. 83–106.
- Dunis, C. L. and Williams, M. (2002). Modelling and trading the eur/usd exchange rate: Do neural network models perform better?, *Derivatives Use, Trading and Regulation* pp. 211–239.
- Fong, W. M. and Yong, L. H. M. (2005). Chasing trends: recursive moving average trading rules and internet stocks, *Journal of Empirical Finance* **12**: 43–76.
- Galeshchuk, S. (2016). Neural networks performance in exchange rate prediction, *ScienceDirect* pp. 446–452.
- Giles, C. L., Lawrence, S. and Tsoi, A. C. (2001). Noisy time series prediction using recurrent neural networks and grammatical inference, *Machine Learning* **44**: 161–183.
- Gurdgiev, C. and Corbet, S. (2018). Ripples in the crypto world: Systemic risks in crypto-currency markets, *International Banker* .
- Guresen, E., Kayakutlu, G. and Daim, T. U. (2011). Using artificial neural network models in stock market index prediction, *Expert Systems with Applications* **38**(8): 10389 – 10397.  
URL: <http://www.sciencedirect.com/science/article/pii/S0957417411002740>

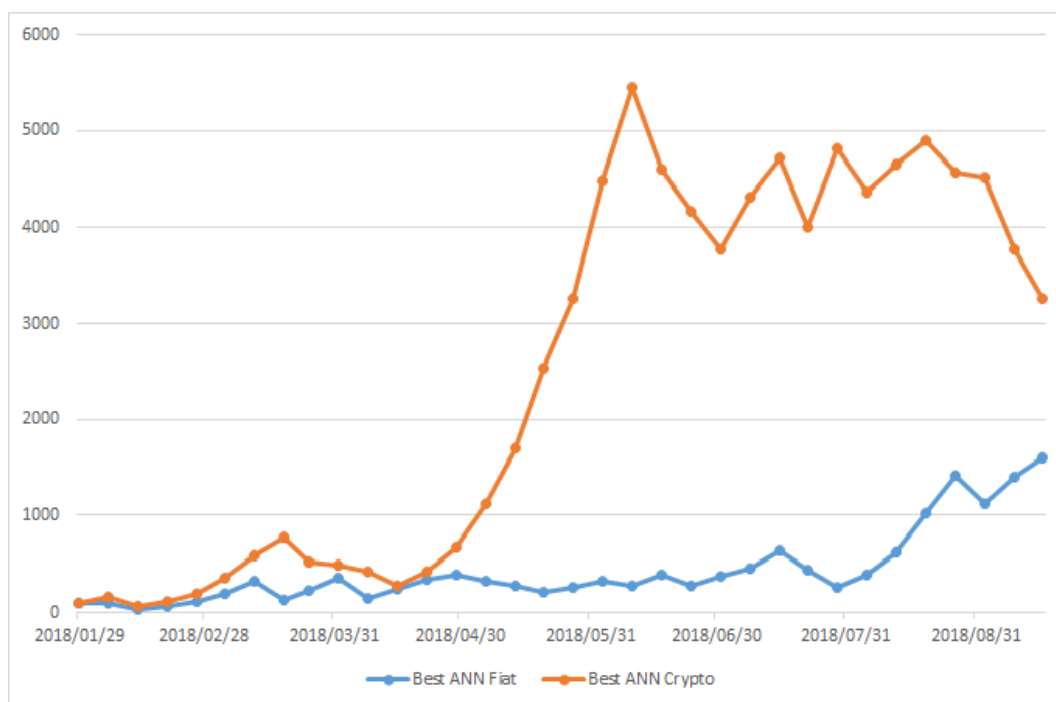
- Huang, C.-L. and Wang, C.-J. (2006). A ga-based feature selection and parameters optimization for support vector machines, *Expert Systems with Applications* **31**(2): 231 – 240.
- Hulley, G. and Marwala, T. (2007). Evolving classifiers: Methods for incremental learning, *ResearchGate* .
- K. V. Prema, N. Manish Agarwal, M. K. and Agarwal, V. (2015). Stock market prediction using neuro-genetic model, *Indian Journal of Science and Technology* .
- Kou, Y., Lu, C., Sirwongwattana, S. and Huang, Y. (2004). Survey of fraud detection techniques, *IEEE International Conference on Networking, Sensing and Control* p. 749–754.
- Magnusson, K. and Olsson, T. (2016). Training artificial neural networks with genetic algorithms for stock forecasting, *Technical report*, KTH ROYAL INSTITUTE OF TECHNOLOGY.
- Narang, R. K. (2009). *Inside the Black Box: The Simple Truth About Quantitative Trading*, number p.256, John Wiley & Sons.
- Ogasawara, E., Martinez, L. C., de Oliveira, D., Zimbrão, G., Pappa, G. L. and Mattoso, M. (2010). Adaptive normalization: A novel data normalization approach for non-stationary time series, *IEEE International Joint Conference on Neural Networks (IJCNN)* .
- Peters, E. E. (1994). *Fractal Market Analysis: Applying Chaos Theory to Investment and Economics*, Wiley.
- Refenes, A. N., Azema-Barac, M., L. Chen, I. and Karoussos, S. A. (1993). Currency exchange rate prediction and neural network design strategies, *Springer* p. 46–58.
- Shazly, M. E. and Lou, A. (2016). Comparing the forecasting performance of futures oil prices with genetically evolved neural networks, *International Advances in Economic Research* **22**: 361–376.
- Shazly, M. R. E. and Shazly, H. E. E. (1999). Forecasting currency prices using a genetically evolved neural network architecture, *International Review of Financial Analysis* pp. 67–82.
- Sher, G. I. (2011). Evolving chart pattern sensitive neural network based forex trading agents, *Research Gate* .
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies, *MIT Press* **10**(2): 99–127.
- Stepnicka, M., Cortez, P., Peralta, J. and Stepnicková, L. (2013). Forecasting seasonal time series with computational intelligence: On recent methods and the potential of their combinations, *Expert Syst. Appl.* **40**: 1981–1992.

- 
- Sullivan, R., Timmermann, A. and White, H. (1999). Data-snooping, technical trading rule performance, and the bootstrap, *The Journal of Finance* **54**: 1647–1691.
- Whalley, A. E., Dewynne, J. N. and Wilmott, P. (1995). *Mathematical Models in Finance*, number P.71, Chapman & Hall.
- Yao, X. (1999). Evolving artificial neural networks, *Proceedings of the IEEE* **87**: 1423 – 1447.

## Appendix A

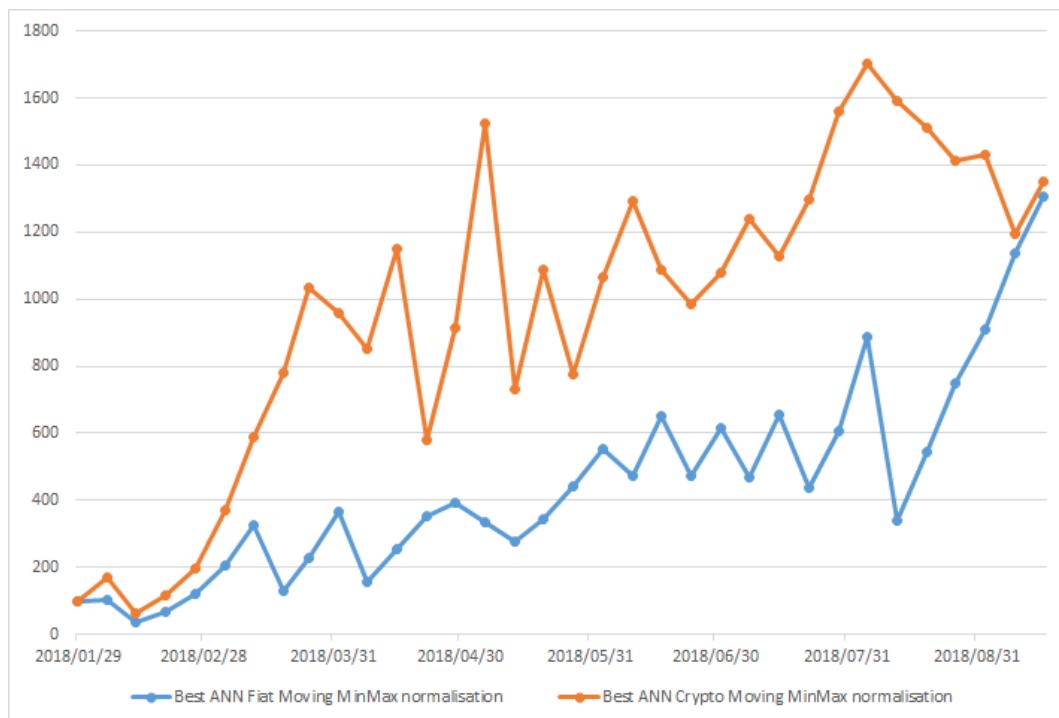
# Appendix

### A.1 Section of Figures



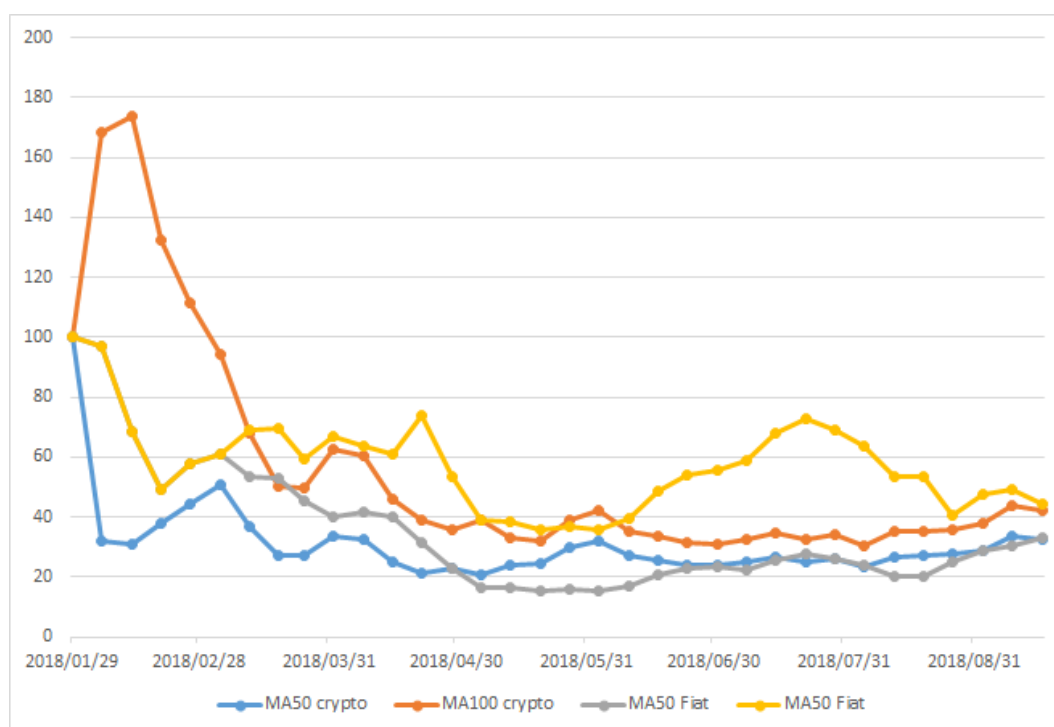
**Fig. A.1:** Best ANNs \$100 evolution overtime

*This shows a comparison of how the best ANN in both contexts grow the initial investment over the test set period 29 January 2018 - 16 September 2018. The conversion of the weekly signals into returns was done using the method described in [4.5.1](#)*



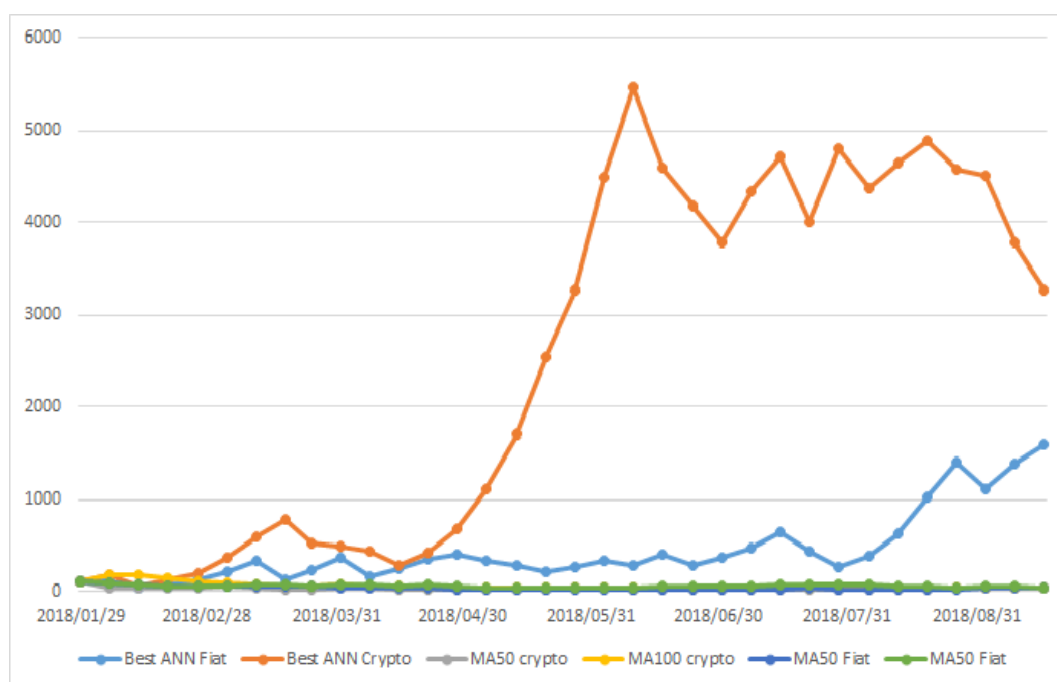
**Fig. A.2:** Best ANNs \$100 evolution overtime for weekly rolling window MinMax normalised data

This shows a comparison of how the best ANN in both contexts grow the initial investment over the test set period using the rolling window normalised data (RWND). The conversion of the weekly signals into returns was done using the method described in 4.5.1



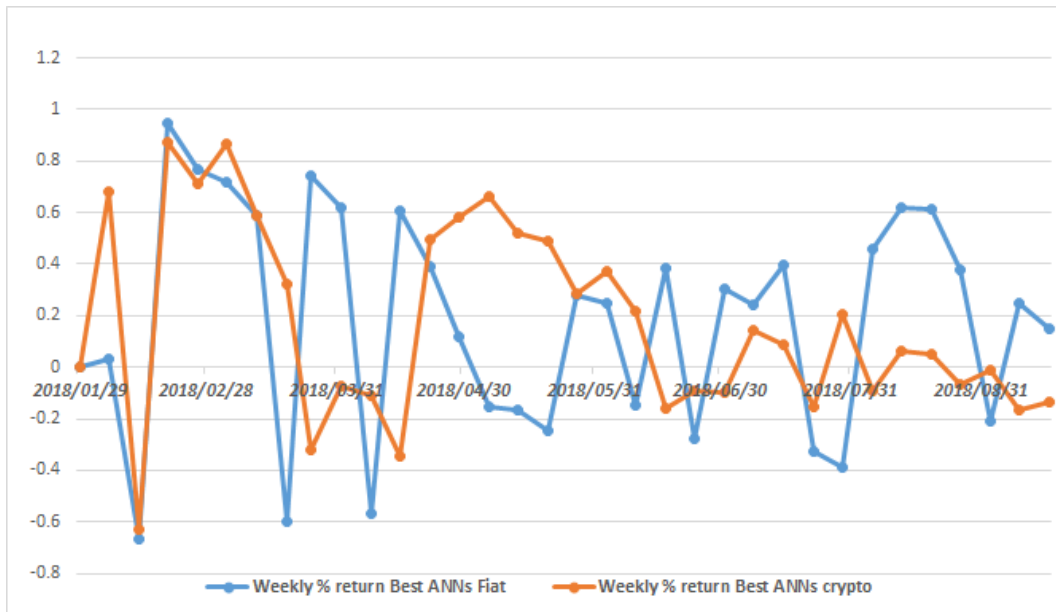
**Fig. A.3:** Moving Averages \$100 evolution overtime

*This shows a comparison of how the Moving Averages in both contexts grow the initial investment over the test set period 29 January 2018 - 16 September 2018. The conversion of the weekly signals into returns was done using the method described in [4.5.1](#)*



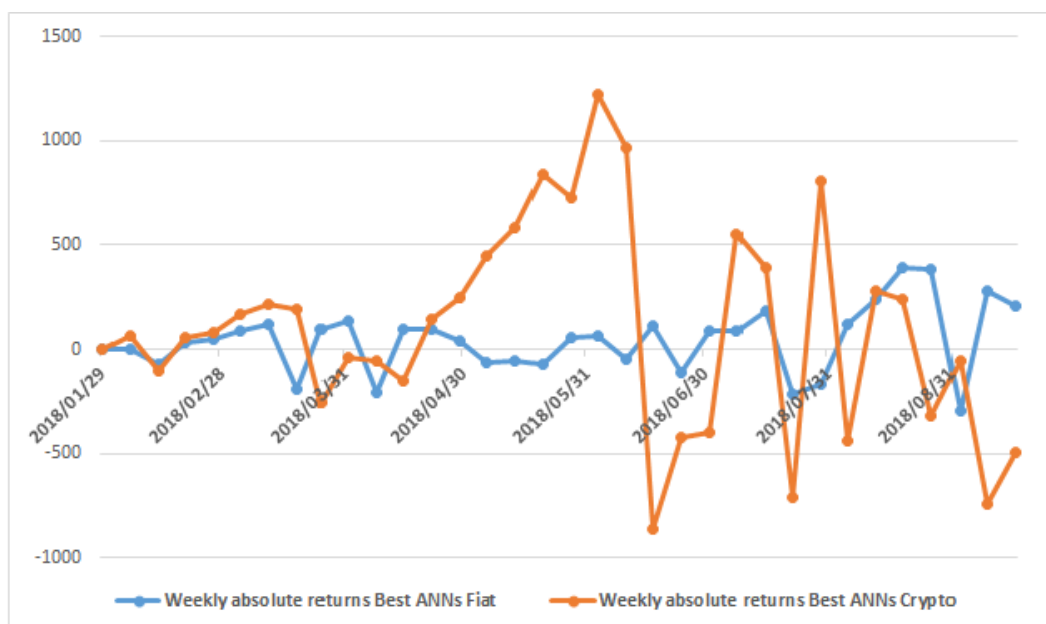
**Fig. A.4:** Best ANNs vs Moving Averages \$100 evolution overtime

*This shows a comparison of how the Moving Averages and the best ANNs in both contexts grow the initial investment over the test set period 29 January 2018 - 16 September 2018. The conversion of the weekly signals into returns was done using the method described in 4.5.1*



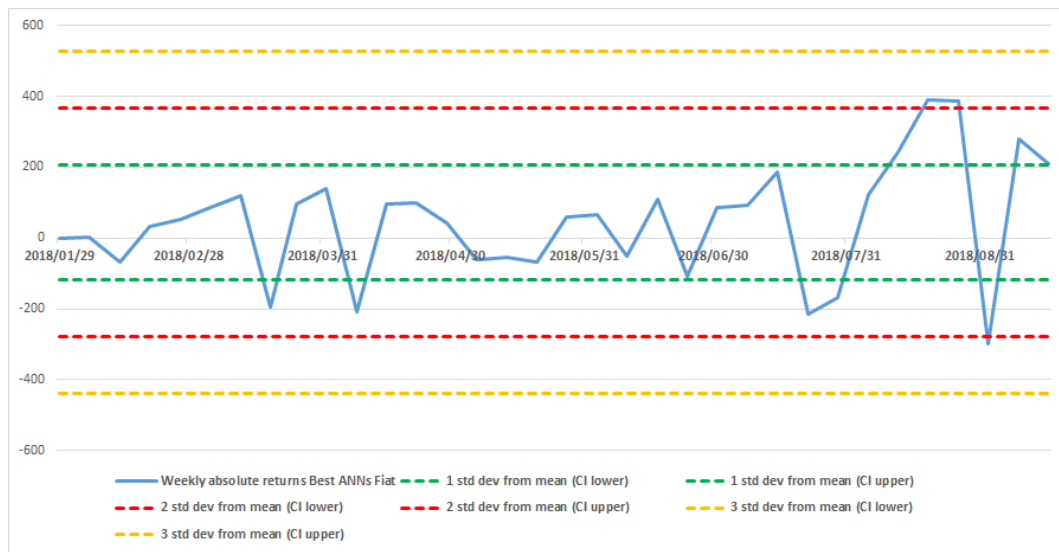
**Fig. A.5:** Weekly percentage change returns overtime

This shows a comparison of weekly percentage returns for the best ANNs in both contexts over the test set period 29 January 2018 - 16 September 2018.



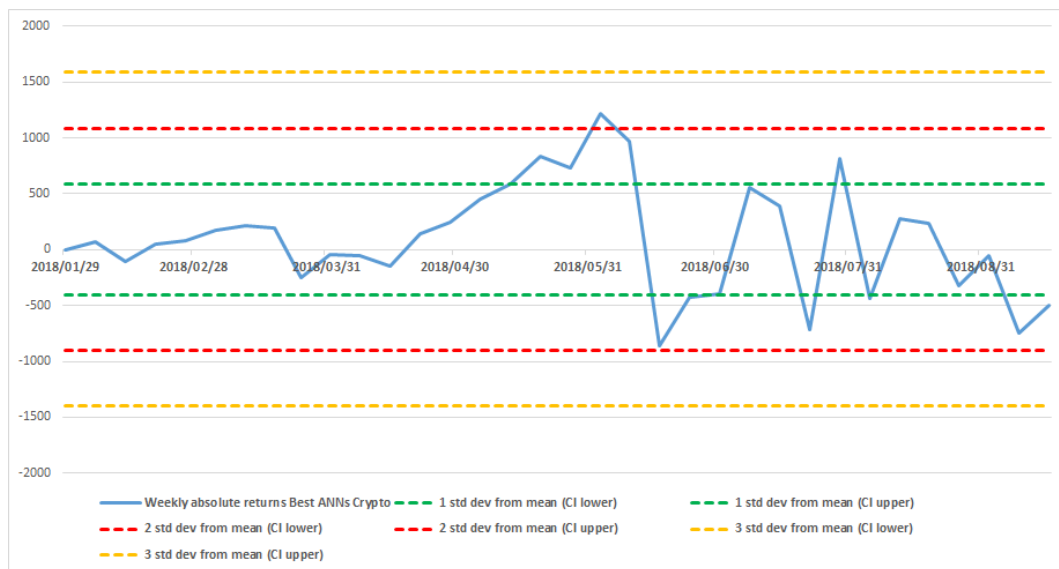
**Fig. A.6:** Weekly Absolute returns overtime

This shows a comparison of weekly Absolute returns for the best ANNs in both contexts over the test set period 29 January 2018 - 16 September 2018.



**Fig. A.7:** Weekly Absolute returns with 1,2 and 3 std. dev from the mean overtime (Fiat context)

This shows the weekly absolute returns with the 1,2 and 3 std. deviations from the mean superimposed on the graph it for the Fiat context. The two outliers highlighted in section 6.3.1 can be seen here.



**Fig. A.8:** Weekly Absolute returns with 1,2 and 3 std. dev from the mean overtime (Cryptocurrency context)

This shows the weekly absolute returns with the 1,2 and 3 std. deviations from mean superimposed on the graph for the crypto context. The outlier highlighted in section 6.3.1 can be seen here.