A Methodology for Securities and Cryptocurrency Trading Using Exploratory Data Analysis and Artificial Intelligence

Ali Al-Ameer
Consulting Services Dept.
Saudi Aramco
Dhahran, Saudi Arabia
ali.alameer.1@aramco.com

Fouad AL-Sunni
Dept. of Systems Engineering
King Fahd University of Petroleum & Minerals
Dhahran, Saudi Arabia
alsunni@kfupm.edu.sa

2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA) | 978-1-6654-1511-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/CAIDA51941.2021.9425223

***Abstract*— This paper discusses securities and cryptocurrency trading using artificial intelligence (AI) in the sense that it focuses on performing Exploratory Data Analysis (EDA) on selected technical indicators before proceeding to modelling, and then to develop more practical models by introducing new reward loss function that maximizes the returns during training phase. The results of EDA reveal that the complex patterns within the data can be better captured by discriminative classification models and this was endorsed by performing back-testing on two securities using Artificial Neural Network (ANN) and Random Forests (RF) as discriminative models against their counterpart Naïve Bayes as a generative model. To enhance the learning process, the new reward loss function is utilized to retrain the ANN with testing on AAPL, IBM, BRENT CRUDE and BTC using auto-trading strategy that serves as the intelligent unit, and the results indicate this loss superiorly outperforms the conventional cross-entropy used in predictive models. The overall results of this work suggest that there should be larger focus on EDA and more practical losses in the research of machine learning modelling for stock market prediction applications.**

Keywords—securities, cryptocurrency, stock market, artificial intelligence, machine learning, probabilistic modelling, classification models, artificial neural network, random forests, naïve bayes

I. Introduction

Stock market prediction, in general, is about predicting, at time t, the stock movement (up or down) in the next k time steps. If profitability is the main prediction objective, formalizing the problem as classification suits it more than regression. There are two main hypotheses in stock market prediction as discussed by Ican in [1]. The first hypothesis suggests that stock prices move in a stochastic manner and cannot be predicted (*Efficient Market Hypothesis*). On the other hand, the other hypothesis claims that stock markets are predictable, at least to a certain degree, and prediction turns into long-term profits. As a matter of fact, the literature is rich with articles that support the latter hypothesis. Specifically, the research is ongoing to investigate the feasibility of probabilistic classification models in terms of profits generation capabilities. One of the main challenges related to modelling stock markets for prediction purposes is the selection of features set for training, and which models optimally fit with this set. Features and model selection philosophy is concerned about the method with which the features were selected, that is whether they are selected based on performing any type of analysis or arbitrary selection. Proper features selection does not only contribute to building more accurate and reliable AI models, but it is also vital for the application of stock market prediction. Professional charts analysts and traders have numerous, if not theoretically infinite, options of technical indicators that they can implement in their trading strategy to generate profits. After selecting the features (technical indicators), exploratory data analysis (EDA) that is concerned with visualizing the features should also be performed for developing a reliable classification model. It is also important to understand the statistical distribution of the classes and the features before attempting modelling, otherwise the models may result in adverse performance as stated by Kuhn [2]. Besides features and models selection, developing and training practical models that learn data patterns related to maximizing the returns rather than ones that optimize classification accuracy is of significant importance. Although that there are works in the literature present their models' results based on profits, the ongoing research related to using supervised machine learning with adjusted loss function that match the purpose of trading is remarkably limited, at least at best of our knowledge.

The novel contribution of this work is proposing a methodology for AI trading with systematic approach of developing and training the models. This mainly constitutes of setting and testing two hypotheses claiming that performing EDA and introducing new reward loss result in higher returns.

II. Literature Review

A. Data and Model Selection Methodology

In the literature of stock market prediction, there are limited number of works that performed some type of data analysis for features and models selection. The work of He in [4] is remarkable as he discussed various complex features selection methods that can be implemented for the application of stock market prediction such as Genetic Algorithm (GA), Principal Component Analysis (PCA) and Sequential Features Selection (SFS); however, the work was not proceeded in testing their efficiency in the application (e.g., applying them on models to test which one is best). Basak in [5] used Gini Impurity technique for features selection as a wrapper method, and the non-linear classifiers Random Forest and eXtreme Gradient Boost were used based on the linearity tests results. In [6], Di used Random Forest for features selection after the creation of 84 features of technical indicators generated at different time horizons, and then they were used to train Support Vector Machine (SVM) model without going through the crucial step of EDA. Cheng in [7] used Pearson Correlation Coefficient (PCC) for supervised filtering of features, that is measuring numerical to categorical data relevancy, although it is well-know that PCC is only useful to detect linear relevancy and is applicable for numerical type of data. It can be concluded that performing data exploratory analysis is a clear gap in the literature of stock market prediction although that it

978-0-7381-3177-1/21/$31.00 ©2021 IEEE

is one of the most crucial steps in developing a classification model with an optimized performance.

B. AI Models' Practicality

AI models' practicality is essential in case they are developed for practical purposes, such as prices forecasting or trading. Since this work focuses on the trading side of it, the practicality of the models discussed in the literature will be reviewed accordingly. There is considerable number of works that evaluate the performance of the model based on its accuracy rather than the profits it generates when it is back tested with out-of-sample dataset. Generally speaking, this evaluation criteria is not practical neither for forecasting purposes nor for trading. To exemplify this, the proposed work by Basak in [5], although that high accuracy for 90-day trading window (>90%) was reported, there is no calculation for the profits which makes it impractical. Note that holding a stock for long period of time in case of leveraged trading with Contracts For Difference (CFD) and short selling, which was assumed in the paper, will cause brokers to apply overnight funding and this definitely affects the returns. Similarly, the models reported in [6], [8] and [11] were not evaluated according to generated returns resulted from unseen data. Noteworthy, there is significant number of works that consider the profits as a performance measure. This, however, is not generally combined with back testing environment that mimics the actual trading environment as much as possible. Although that Gerlein in [9] used the profits as a measure and he discussed the importance of trading costs, such as spread and slippage, these associated costs were not considered in profit calculations, which leaves the model practicality unproved. This is also the case in the work presented by Carapuço in [10] where only the spread cost was considered while neglecting any associated transactions commissions. Despite that Li in [12] considered both transaction and spread costs in the return calculations, the considered transaction cost is relatively low (0.05%) for the leverage trading that was implemented in the trading strategy.

III. Objective, Hypotheses Description And Methodology

A. Objective

The main objective of this work is to develop a probabilistic classification model for trading applications, i.e., the probability of the class, either the asset will go higher or lower in price at t+1, given the features (input) at t, and it is given in the following general form:

$$p(C_k|\boldsymbol{S}) = f_k(\boldsymbol{W}, \boldsymbol{S})$$

Where C_k is the class for k = 1, 2, $\boldsymbol{S}$ is the features space, $\boldsymbol{W}$ is the set of parameters to be estimated during training and f is the mapping function.

B. Hypotheses Description

***First Hypothesis*:** Performing extensive numerical and exploratory (visual) data analysis on features and classes help in selecting a proper model structure for the data; as a result, more practical models in terms of accuracy and profitability will be obtained.

***Second Hypothesis*:** As long as profitability is considered as the main objective, a loss function that maximizes the models' generated profits during training results in higher returns compared to a loss that only considers the accuracy.

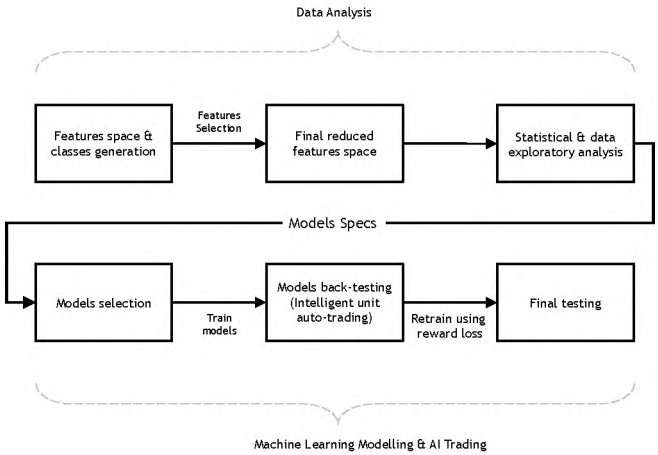

Fig. 1 Overview of the proposed methodology

TABLE 1 TIME PERIODS OF DATASET

Asset	Daily Prices Period	Hourly Prices Period
AAPL	Jan 2007 – Jan 2018	Jan 2017 – Dec 2020
IBM	Jan 2007 – Jan 2018	Jan 2017 – Dec 2020
OIL	--	Mar 2019 – Dec 2020
BTC	--	Mar 2019 – Dec 2020

TABLE 2 FINAL SELECTED FEATURES FOR AAPL AND IBM

	AAPL	IBM
Indicator (i)	***Period (n)***	***Period (n)***
RSI	21, 22	8, 27
Fast %K	21, 49	12, 59
Slow %K	3, 5	35, 58
Slow %D	5, 19	34, 59
ADX	25, 43	10, 51
CCI	5, 18	3, 41
FRL	22, 49	12, 59

C. Models Development Methodology

In order to meet the objectives of the work and to perform efficient testing for the hypotheses, the following steps (depicted in Fig.1) are considered for the development of the probabilistic models:

1. Retrieving the original data; that is the assets opening, closing, high and low prices over the considered time period for model development.
2. Man-crafted features extraction (technical indicators).
3. Numerical data analysis by utilizing features selection methods.
4. Exploratory data analysis (visualization) to assess the relationship between selected features and classes, data statistical distributions and time dependency of the data.
5. Proposing and training appropriate probabilistic models.
6. Assessing the selected trained models in terms of accuracy and profits (test of first hypothesis).
7. Retrain models with a reward loss function that has more practical sense for stock prediction applications.

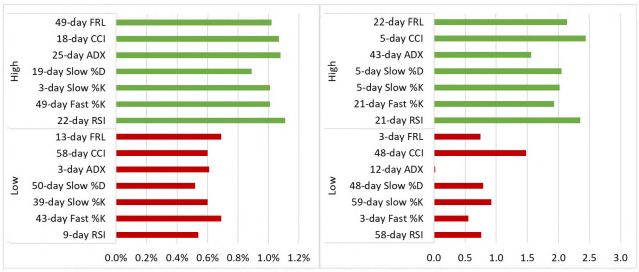

Fig. 2 AAPL SU (left) and t-Stat (right) tests results

8. Reassessing the models with using the reward loss (test of second hypothesis) and building auto-trading strategy that serves as an intelligent trading unit (AI).

IV. Data Analysis

A. Features Extraction and Selection

1) Raw Data Retrieval – Prices Quotes

The retrieved price quotes will be for different assets including shares, commodities and cryptocurrencies. These specifically are AAPL and IBM as shares, Brent Crude as commodity and Bitcoin as crypto. Furthermore, both daily and intra-day (hourly) prices are retrieved as detailed in Table 1. The raw data are the time step prices including closing, opening, low and high. The daily prices were retrieved from *finance.yahoo.com* while the intra-day data were retrieved from *www.dukascopy.com*, and the historical stock-split price adjustment is also considered.

2) Technical Indicators and Targets Generation

According to Katz [13], there are three categories for trading models with respect to the utilized technical indicators. These are Break-Out models (BOM), Moving Average Models (MAM) and Oscillator-Based Models (OBM). For this work, the models will be based on oscillatory indicators and they are Relative Strength Index (RSI), Commodity Channel Index (CCI), fast and slow Stochastics (%K and %D) and Fibonacci Retracement Level (FRL). Hence, the total number of used indicators is seven (7). Each indicator is generated more than once at different past time periods. The used periods are from 3-day to 60-day. In general, an indicator i with past time period n is indicated by a column vector $\boldsymbol{s}_{i,n}$ with length N (number of samples), where i = 1, 2, ….,7 represents a specific indicator and n = 3, 4, 5,..., 60 represents the indicator period and is extracted using the price quotes from t down to $t - n - 1$. For example, a 3-day technical indicator, say RSI, that uses the prices from t down to $t - 2$ is indicated by $\boldsymbol{s}_{1,3}$. Each indicator, including its all periods, will be represented by a subspace of features given by $\boldsymbol{S}_i$ where $\boldsymbol{s}_{i,n} \subset \boldsymbol{S}_i \; \forall \, n$. The final space of features $\boldsymbol{S}$ will constitute of two features only from each subspace $\boldsymbol{S}_i$, bringing the dimensionality of the final features space to 14 with $\boldsymbol{s}_j$ for j = 1, 2, 3, …, 14 represents a feature in the final space $\boldsymbol{S}$. The two features from each $\boldsymbol{S}_i$ will be selected based on statistical analysis that measures the relevancy between each feature in the subspace, $\boldsymbol{s}_{i,n} \subset \boldsymbol{S}_i \; \forall \, n$, with the classes, as discussed and detailed in the next two subsections. The main advantage of this general methodology is the ability of using all oscillatory technical indicators discussed by Katz [13] and at the same time only the most relevant ones are used for models' training. Noteworthy, for data analysis purposes only, the features were discretized in order to ease the process of the analysis. All features $\boldsymbol{s}_{i,n} \subset \boldsymbol{S}_i \; \forall \, n, i$ were discretized as follows:

$$\boldsymbol{s}_{i,n} : \; 0 \leq s_{i,n,t} \leq 100 \rightarrow \{0, 1, 2, 3, \ldots\ldots, 100\}$$

Where $s_{i,n,t}$ represents an observation of a technical indicator i calculated at the period of n at time t. Note that the Commodity Channel Index is not bounded between 0 and 100, and therefore it was rescaled before the discretization process.

Since the problem is identified as a binary classification problem, either the price will increase or decrease in the next time step, the target at time t for the dataset was defined as follows:

$$T_t = \begin{cases} C_1 = 1; & close_{t+1} - close_t > 0 \\ C_2 = 0; & close_{t+1} - close_t \leq 0 \end{cases}$$

Where $T_t \in \boldsymbol{T}$ and $\boldsymbol{T}$ is the target vector with size N.

3) Symmetric Uncertainty (SU)

The SU between a feature and the target (two random variables) is the normalized mutual information between the two and thus can be given by:

$$SU(\boldsymbol{s}_{i,n}, \boldsymbol{T}) = 2 \times \frac{MI(\boldsymbol{s}_{i,n}, \boldsymbol{T})}{H(\boldsymbol{s}_{i,n}) + H(\boldsymbol{T})} \tag{1}$$

Where MI and H are the mutual information and entropy respectively.

4) t-Statistics

The usefulness of t-Stat method is that it calculates the normalized difference between the mean of the class conditional probabilities i.e. $p(\boldsymbol{s}_{i,n}|C_1)$ and $p(\boldsymbol{s}_{i,n}|C_2)$, and it is given by:

$$TS = \frac{\sqrt{N}\left(\overline{\boldsymbol{s}_{i,n,C_1}} - \overline{\boldsymbol{s}_{i,n,C_2}}\right)}{\sqrt{\sigma^2_{\boldsymbol{s}_{i,n,C_1}} + \sigma^2_{\boldsymbol{s}_{i,n,C_2}}}} \tag{2}$$

Where:

$\overline{\boldsymbol{s}_{i,n,C_1}}, \overline{\boldsymbol{s}_{i,n,C_2}}$: the means of the first and second classes conditional probabilities, respectively

$\sigma^2_{\boldsymbol{s}_{i,n,C_1}}, \sigma^2_{\boldsymbol{s}_{i,n,C_2}}$: the variances of the first and second classes conditional probabilities, respectively

N: total number of observations (samples)

The results of *SU* and t-stat tests for AAPL, daily data, are illustrated in the bar charts in Fig.2. For each indicator space $\boldsymbol{S}_i$, only vectors $\boldsymbol{s}_{i,n}$ with the highest and lowest values are shown. The green bars are associated with *max* and the red are with *min.* Based on this result and for illustration purposes, the final selected features, for both AAPL and IBM daily data, are shown in Table 2.

B. Exploratory Data Analysis

The main objective of this analysis is to select a proper classification model that can optimize the use of the data. Accordingly, the features with the classes are analysed in the following sense:

- Data linearity: that is if it is possible to separate them by a hyperplane.
- Assessing the statistical distribution of the data; that is the probability mass function of $p(\boldsymbol{s}_j|C_k) \forall j, k$.

TABLE 3 CHARACTERISTICS OF PROPOSED CLASSIFICATION MODELS

	NBC	ANN	RF
Non-Linear		•	•
Distribution Assumptions	•		
Loss error weight	•	•	•

- Assessing the distribution of the classes' prior probabilities i.e. $p(C_k)\forall k$.
- Assessing the time-dependency to capture any time relationship between the features and the classes (applicability of sequential data models).

Note that for illustration in this section, only AAPL and IBM daily data are discussed although that same process was applied on other assets.

1) Linear Separability Test

A Hard Margin SVM classifier is utilized to test for data separability. The test procedure is explained as follows:

- The Kernel function used for the test is linear in order to test if a hyperplane can separate the data or not.
- The cost parameter in the loss function was set theoretically to infinity to force "Hard-Margin" classification of the SVM model and hence to overfit it.
- After training, the accuracy of the model is monitored.
- If the accuracy is 100%, then the data is linearly separable; it is not otherwise.

The results of this classifier for both AAPL and IBM are shown in Fig.4. Since the accuracy results of the linear SVM model is not even close to 100%, it can be concluded that the data are not linearly separable.

2) Scatter Plot Matrix

The scatter plot matrix is a very helpfully visual tool for data analysis. The off-diagonal elements of the matrix show the scatter plot for two features while each diagonal element shows the class-conditional histograms, i.e. $p(\boldsymbol{s_j}|C_k)\forall k$. This tool can help to see if the data are generally separable or not, whether linearly or non-linearly. The other advantage of this plot is determining the overlapping severity of the class-conditional distributions as this is highly related with the capability of classification models to perform well with the data [14]. The resultant scatter plot matrices for AAPL and IBM are shown in Fig.3.

3) Parallel Coordinates Plots

Parallel Coordinate Plot was used to assess time-dependency of features to classes. Noteworthy, the plot is originally used as multi-variate visual tool to visualize the usefulness of the selected features in a certain classification task. In this case, the technique was used to plot the indicators/features time series from *t*-7 up to *t*. This will illustrate the applicability of sequential data models in this particular application. A single plot for AAPL with RSI indicator, to exemplify, is shown in Fig.5.

4) Prior Class Distributions

The prior class distributions are important to be assessed as part of the data analysis. In case of imbalance in their distributions, then a special treatment shall be carried out during model development in order to prevent the adverse impact they may cause. Imbalance could make the model, when it is given out-of-sample data for testing, always predict the class with the higher prior probability. One of the methods suggested by Kuhn in [2] is to introduce an observation weights vector into the model loss and adjust it according to the imbalance found in the data. Fig.6 shows the distributions for both AAPL and IBM.

5) Data Analysis Discussion

From the SVM linearity test, it can be observed that the data are non-linearly separable. Also, the scatter plot matrix shows that the statistical distributions do not always follow a Gaussian or one of the exponential family distributions. This can be clearly seen in the stochastics oscillators, CCI and Fibonacci indicators. Remarkably, the distributions are also found with severe overlapping that makes the classification task more complex and challenging. For the prior class probabilities, the distributions of the classes are imbalanced, especially for AAPL, and proper measures to mitigate the impact of such imbalance shall be taken into consideration in the modelling phase. Parallel coordinates plots indicate that there is no obvious pattern that can be recognized between past time steps of features and the targets. In summary, the performed data analysis proves that the classification in stock market prediction, with using technical indicators as features, is complex and as a result demands a model that can deal with such complexity.

V. PROPOSED MACHINE LEARNING MODELS

Based on the results of the data analysis in the previous section, the models that are capable to recognize a pattern in the data should have the following properties:

- Nonlinear models.
- Capable of estimating/and or approximating complex statistical distributions.
- Have a loss function with error weight to handle the imbalanced distribution of class priors.

The selected models based on the data analysis with their associated properties are shown in Table 3. One linear parametric model, Naïve Bayes Classifier, that has significant restrictions in terms of used data distributions, is examined to test its accuracy against two statistical non-parametric models.

A. Naïve Bayes Classifier

Naïve Bayes Classifier (NBC) is a generative model in nature where the posterior probabilities are computed from the estimate of the class conditional probabilities $p(\boldsymbol{S}|C_k)$, and then Bayes probability theorem is applied to find the posterior $p(C_k|\boldsymbol{S})$. Thus, the general form of the model is given by:

$$p(C_k|\boldsymbol{S}) = p(C_k)p(\boldsymbol{S}|C_k) \quad (3)$$

To compute the posterior, a prior assumption of the class conditional probabilities shall be given and this will restrict the flexibility of the model. Once an assumption is given, the parameters of the statistical distribution assumed for $p(\boldsymbol{S}|C_k)$ are estimated. Consequently, the final posterior probabilities for binary classification, using total probability law and assuming independency between the features in $\boldsymbol{S}$, are given by Hastie [15]:

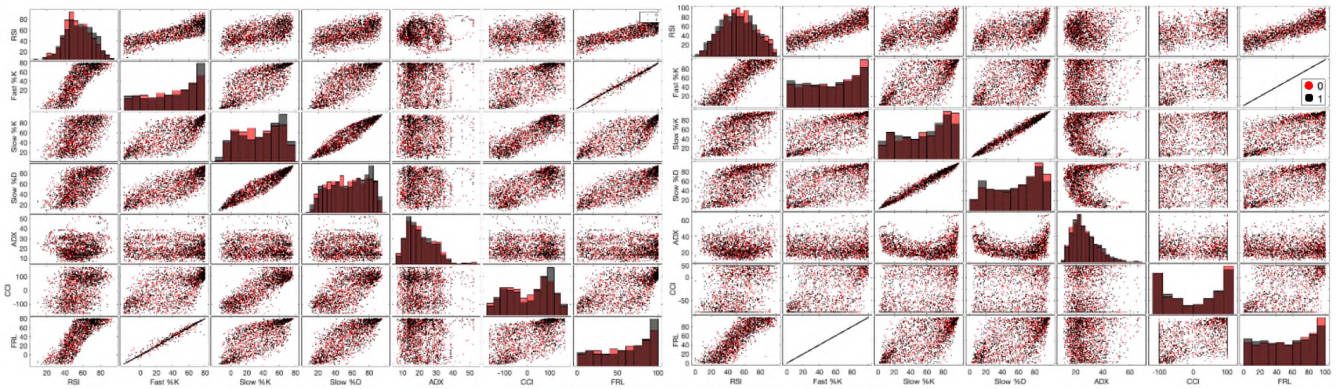

Fig. 3 Scatter Plot Matrix, AAPL (right) IBM (left)

Fig. 4 Linear separability test results for AAPL (left) and IBM (right)

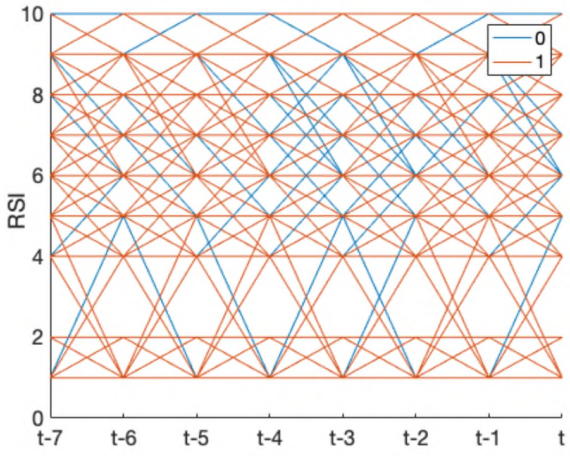

Fig. 5 Parallel Coordinates Plot for AAPL, RSI

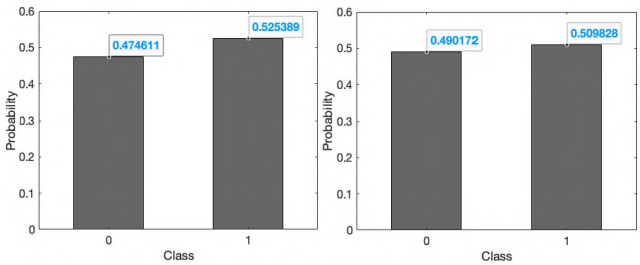

Fig. 6 Prior classes distribution for AAPL (left) and IBM (right)

$$p(C_k|\boldsymbol{S}) = \frac{p(C_k)\prod_{j=1}^{D} p(\boldsymbol{s}_j|C_k)}{\sum_{i=1}^{2} p(C_i)\prod_{j=1}^{D} p(\boldsymbol{s}_j|C_k)} \quad (4)$$

Where $D = 14$, which is $\boldsymbol{S}$'s dimension. The NBC model converges while optimizing the following loss:

$$\frac{1}{N}\sum_{i=1}^{N} I\{\hat{T}_i \neq T_i\} \quad (5)$$

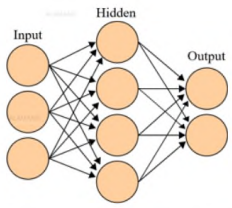

Fig. 7 Typical architecture of ANN model

Where $\hat{T}_i$ is the prediction at sample i, and I is the indicator function.

B. *Ensemble Decision Trees – Random Forest*

The Random Forest (RF) model composes of multi single decision trees and hence the term "Forest". Each tree has a unique number of splits M and leaf size L. In this model, the final decision depends on the most votes, one vote for each tree, for a certain class. The RF is discriminative in nature. Each decision tree with M splits can be evaluated by the Gini impurity given by Basak [5]:

$$G(M) = 1 - \sum_{k=1}^{2} p(C_k) \quad (6)$$

The loss function of this model is described in (5).

C. *Artificial Neural Network*

ANN models are considered non-parametric and discriminative models. The neurons in the first layer form the inputs (or features) into the network. In this layer, no transfer function is applied on the neurons' inputs. Unlike the input layer, hidden and output layers' neurons perform mathematical processing on their inputs. As shown in Fig.7, every link between two neurons in successive layers has a parameter called "*weight*". In addition, there are "*bias*" parameters in the network; one at each node in a hidden layer and two *bias* terms associated with input and output layers. *Weights* and *biases* terms form the mathematical model parameters that has to be estimated during the network training phase. For hidden layers' neurons, every neuron sums all preceding layer's neurons outputs and apply an *activation function,* which could by hyper tan, RELU or others depending on the problem. For classification problems, the output layer activation function is given by the famous *softmax* stated in [3] by Bishop:

$$p(C_k|\boldsymbol{S}) = \frac{p[\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{S})|C_k]}{\sum_j p[\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{S})|C_j]p(C_j)} \quad (7)$$

Where $\boldsymbol{\phi}(\boldsymbol{S})$ is the hidden layer activation functions vector (assuming single-hidden layer network). The loss function is the cross-entropy and given by Bishop [3]:

$$V_{ANN} = -\sum_{j=1}^{N}\sum_{k=1}^{2} T_{jk} log\ (\hat{T}_{jk}) \tag{8}$$

D. Models' Custom Reward Loss

In order to achieve maximum practicality for the proposed models, and also to test the second hypothesis, a reward loss function will be used to retrain the model with the highest accuracy in the first hypothesis. The reward loss should consider the generated returns by the model and maximize it, instead of minimizing the classification error. Therefore, this function can be described as follows:

$$V_c = \frac{1}{R_a}\sum_{t=1}^{N}\left(\prod_{i=1}^{t}[1+R_{a,i}] - \prod_{i=1}^{t}[1+R_{p,i}]\right)^2 \tag{9}$$

Where:

$$R_{p,i}: \begin{cases} R_{a,i}; & \hat{T}_i = T_i \\ -R_{a,i}; & otherwise \end{cases}$$

R_a: The total actual accumulative returns over the whole period composed of N samples.

$R_{a,i}$: The actual price change at time step i.

The summation and division are for normalization purposes, preventing the loss from exploding with the continuously increased difference between actual and resulted returns from prediction. Noteworthy, in this way, it can be seen that past performance will be **relevant** to future performance of the model at each time step during training.

VI. Hypotheses Testing

A. Test of First Hypothesis

1) Models' Architectures – Training and Validation

The models developed here are for AAPL and IBM with the daily prices, hence the time step t is days henceforth in this section. This should be enough for the purpose of testing the usefulness of EDA in the process of developing a probabilistic model for stock market prediction. Also, the data used for training are from Jan 2007 – Dec 2016, this includes both training and validation. The out-of-sample (testing) period is from Jan 2017 – Jan 2018. Table 4 shows the final hyperparameters that result in best training/validation performance for all models, for both AAPL and IBM.

2) Back Testing Procedure (Intelligent unit strategy)

After training, the models were back tested in terms of the returns each share will generate over a period of 13 months, from Jan 2017 up to Jan 2018. The trading strategy during testing was as follows:

1. At time t, the trained model is fed with the features (input) to make the prediction for the next time step $t + 1$.
2. If the prediction is to increase, a long position is opened; otherwise, the asset will be shorted.
3. At time $t + 1$, another prediction is done for the next time step $t + 2$.

TABLE 4 HYPERPARAMETERS OF TESTED MODELS IN FIRST HYPOTHESIS

		AAPL	IBM
NBC	***Data Distribution***	Uniform Kernel	Gaussian Kernel
RF	***Trees***	296	300
	Leaves	37	50
	Features/split	1	1
ANN	***Hidden Layers***	2 (tanh)	2 (tanh)
	Neurons	8, 6	10, 8

TABLE 5 TEST RESULTS OF FIRST HYPOTHESIS (OUT-OF-SAMPLE DATA)

		Accuracy	Returns	# Transactions	Return/ Transaction*
ANN	*AAPL*	**55%**	**40%**	**33**	**1.51%**
NBC		51%	7%	49	0.44%
RF		54%	25%	41	0.91%
ANN	*IBM*	**50%**	**10%**	**33**	**0.60%**
NBC		47%	-25%	93	0.03%
RF		51%	4%	23	0.47%

* *Calculated before applying the transaction cost of 0.3%.*

TABLE 6 ANN ARCHITECTURES FOR SECOND HYPOTHESIS TESTING

	AAPL	IBM	OIL	BTC
Hidden Layers	6 (RELU)			
Neurons	30	30	50	50
Output Function	softmax			
Loss Function*	Cross-entropy (8) and Custom Reward Loss (9)			

* *Used separately to produce two models for each asset, for comparison purposes*

TABLE 7 RESULTS OF SECOND HYPOTHESIS TEST

		Accuracy	Returns	# Transactions	Return/ Transaction*
Cross-entropy	*AAPL*	53%	30%	704	0.34%
Reward Loss		**53%**	**178%**	**549**	**0.62%**
B&H		--	54%	--	--
Cross-entropy	*IBM*	46%	28%	514	0.35%
Reward Loss		**46%**	**128%**	**432**	**0.60%**
B&H		--	-8%	--	--
Cross-entropy	*OIL*	43%	-57%	413	0.16%
Reward Loss		**52%**	**-48%**	**1,194**	**0.26%**
B&H		--	123%	--	--
Cross-entropy	*BTC*	57%	130%	1,400	0.39%
Reward Loss		**58%**	**331%**	**1,400**	**0.53%**
B&H		--	111%	--	--

* *Calculated before applying the transaction cost of 0.3%.*

4. If the prediction is similar to the previous one, the position will be held, if not, the position is reversed from long to short or short to long.
5. This process continues until the last time step in the testing period.

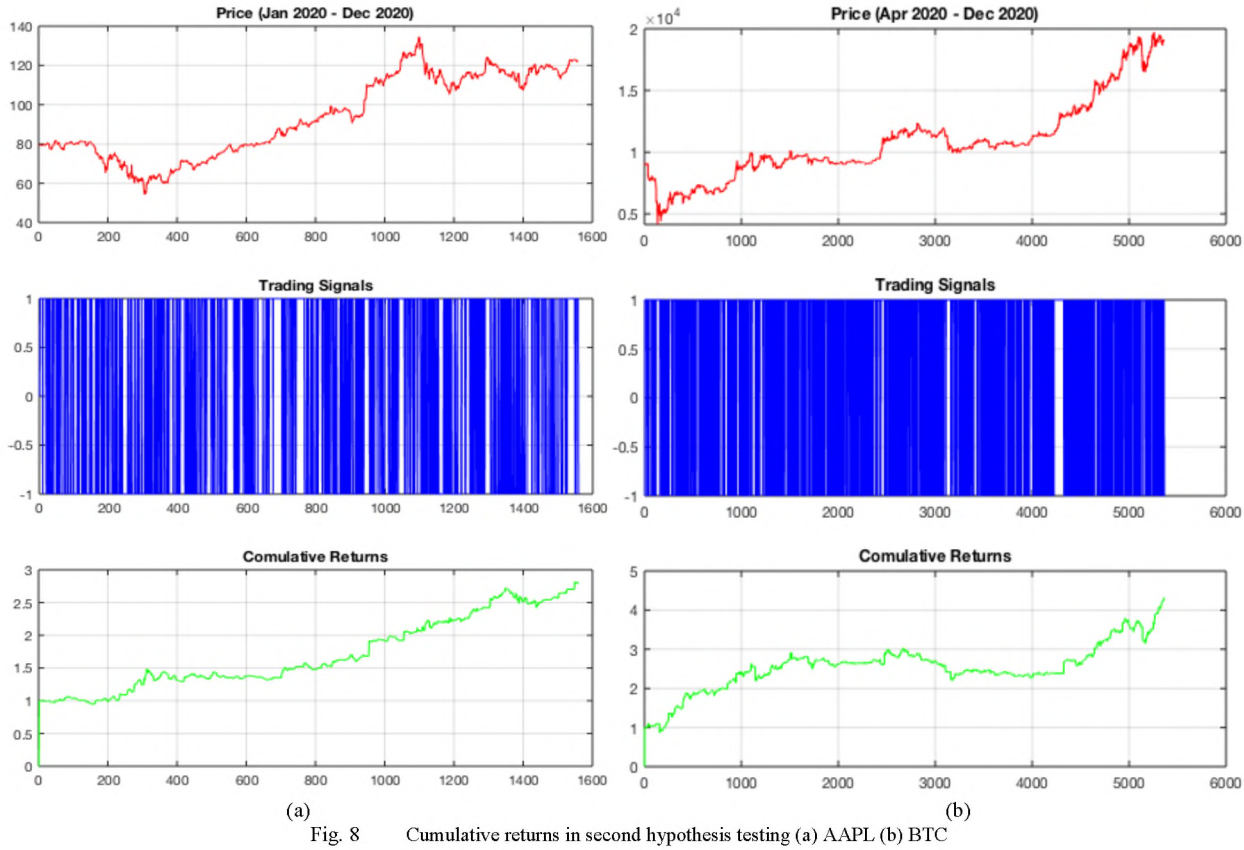

Fig. 8 Cumulative returns in second hypothesis testing (a) AAPL (b) BTC

In order to mimic the real trading environment, relative conservative fees per trade is applied. For example, a transaction cost of 0.3% for each position was considered. More rigorous fees that is accounted for is the spread cost, that is the difference between ask and bid prices. This was adjusted as 5 and 22 cents for AAPL and IBM, respectively.

3) Results and Discussion

The results of the out-of-sample dataset are shown in Table 5. In general, it can be observed that the results support the first hypothesis. The NBC was the lowest in terms of accuracy, for both AAPL and IBM. This is expected since the NBC has restrictive assumptions for the class conditional probabilities and it requires a parametric estimation of the assumed distribution. Moreover, when comparing the results of ANN and RF, they are close in terms of accuracy, but the former outperforms the latter when it comes to returns. Note that in Table 5, the Return/Transaction columns shows the return before deducting the transaction cost of 0.3%. This is a significant measure that should be taken into consideration when performing Trading Cost Analysis (TCA) for an auto-trading model.

B. Test of Second Hypothesis

The models developed here are for AAPL, IBM, OIL and BTC with the hourly prices, hence the time step t is hour henceforth in this section. Also, the data used for training are from Jan 2017 – Dec 2019 for AAPL and IBM, and Mar 2018 – Mar 2020 for OIL and BTC. The out-of-sample (testing) period is Jan – Dec 2020 for AAPL and IBM, and April – Dec 2020 for OIL and BTC.

1) Models' Architectures – Training and Validation

As it was discussed earlier, the model that outperforms other models in the first hypothesis test is the only model that will be examined here. Hence, the selected model is ANN with the custom reward loss described in (9). Note that all features were selected based on the process discussed in section IV.A, and hence the networks have 14 neurons as inputs. Also, the models were trained and optimized using scaled conjugant gradient algorithm with 6 hidden layers having a RELU activation function. The use of RELU here instead of the hyperbolic tangent is due to the risk of having a "vanishing gradient" when using *tanh*, which is a common problem for deep networks that will make the model easily stuck in a local minimum during optimization. The detailed architecture of the models is shown in Table 6. Note that also a separate model for each asset was trained using cross-entropy in (8) for performing the comparison with the models trained by (9).

2) Results and Discussoin

The same testing procedure of section VI.A.2 was followed here. The summary of the results is listed in Table 7. Also, the cumulative returns generated by the models of AAPL and BTC along with the trading signals and the prices are shown in Fig.8(a)-(b). In Fig.8, a trading signal of 1 means a long position is opened while –1 indicates shorting the asset. From Table 7, it can be seen that the proposed reward loss outperforms the conventional cross-entropy and the evaluation benchmark B&H significantly, and this is the case for all assets except for OIL. This strongly supports the validity of the second hypothesis. However, note that for OIL, the overall returns are in minus, that is a loss during the testing period. This can be attributed to the low volatility of OIL compared to others; consequently, the profit generated by a position is not enough to breakeven the associated costs. This can be justified by the positive returns per transaction for OIL in Table 7, which indicates that the trading signals are in general profitable before deducing the cost associated with the transactions. Looking at the cumulative returns for AAPL in Fig.8(a), it can be observed that investing 1 price unit (pu) at

the beginning of the period would result in 2.78 pu at the end. When exploring how the model was trading, it is noticed that the returns were generally in an upward trend except for the period from around 1,350 to 1,450, where the returns dropped from about 2.65 to 2.5. The BTC returns on the other hand resulted in 4.31 pu for every 1 pu invested. Note that how the returns started saturating at some point during the testing period; however, it could pick-up again in continuing increasing the profits. This indicates that the model is robust enough to handle different situations in the crypto market. Noticeably, the cumulative returns resulted by both assets' models are approximately following the prices' trend patterns but with observable amplification.

VII. Conclusion

This work proposed a prediction methodology that focuses more in assessing the selected data using EDA techniques in an effort to use more proper models, and also to enhance their practicality by introducing new reward loss function during training. The objective of this work was met by defining and testing two hypotheses related to EDA and introducing new reward loss function. The results of the tests strongly supported the first hypothesis as it was seen that the complexity in the data could not be captured properly with simple models such as NBC, rather they were better represented by ANN and RF in terms of classification accuracy. In proceeding with testing the second hypothesis, a new loss function that maximizes the returns was implemented to retrain the ANN. The subsequent testing results of profits generation showed that this loss superiorly outperformed conventional losses of predictive models such as the well-known cross-entropy.

The results of this work suggest that there should be larger focus on EDA and more practical losses in the research of machine learning modelling for trading applications. Our on-going work is addressing issues related to the use of more advanced EDA techniques related to sequential data analysis to deeply examine if they could achieve better results. Moreover, the decision-making process (trading strategy) is also being examined for potential enhancement through having a model that can predict more than one time step ahead. A regression ANN model with multi-step ahead prediction combined with the reward loss introduced in this work is expected to result in superior performance in terms of profits generation.

References

[1] Ö. Ican and T. B. Çelik, "Stock Market Prediction Performance of Neural Networks: A Literature Review," *International Journal of Economics and Finance*, 9(11), 100., 2017 doi:10.5539/ijef.v9n11p100

[2] M. Kuhn and K. Johnson, Applied Predictive Modelling, Michigan, USA: Springer, 2013, pp. 419 – 445

[3] C. M Bishop, Pattern Recognition and Machine Learning, New York, USA: Springer, 2006, pp. 114 – 227

[4] Y. He, K. Fataliyev, and L. Wang, "Feature selection for stock market analysis," *International conference on neural information processing* (pp. 737-744). Springer, Berlin, Heidelberg. 2013, November

[5] S. Basak, S. Kar, S. Saha, L. Khaidem, and S. R. Dey, "Predicting the direction of stock market prices using tree-based classifiers". *The North American Journal of Economics and Finance*, *47*, 2019, pp.552-567.

[6] X. Di, "Stock trend prediction with technical indicators using SVM," *Independent Work Report, Stanford Univ,* 2014

[7] C. H. Cheng, T. L. Chen, and L. Y. Wei, "A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting," *Information Sciences*, *180*(9), 2010, pp.1610-1629.

[8] S. Dey, Y. Kumar, S. Saha, and S. Basak, "Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting" *PESIT,* Bengaluru, India, Working Paper, 2016

[9] E. A. Gerlein, M. McGinnity, A. Belatreche, and S. Coleman, "Evaluating machine learning classification for financial trading: An empirical approach" *Expert Systems with Applications*, *54*, 2016, pp.193-207.

[10] J. Carapuço, R. Neves, and N. Horta, "Reinforcement learning applied to Forex trading," *Applied Soft Computing*, *73*, 2018, pp.783-794.

[11] M. Qiu, and Y. Song, "Predicting the direction of stock market index movement using an optimized artificial neural network model," *PloS one*, *11*(5), p.e0155133, 2016

[12] Y. Li, W. Zheng, and Z. Zheng, "Deep robust reinforcement learning for practical algorithmic trading," *IEEE Access*, *7*, 2019, pp.108014-108022.

[13] J. O. Katz, and D. L. McCORMICK, The encyclopedia of trading strategies. New York: McGraw-Hill, 2000, pp. 83 – 153

[14] J. A. Sáez, M. Galar, and B. Krawczyk, "Addressing the overlapping data problem in classification using the One-vs-One decomposition strategy," *IEEE Access*, *7*, 2019, pp.83396-83411.

[15] T. Hastie, R. Tibshirani, and J. Friedman, The elements of statistical learning: data mining, inference, and prediction., Springer Science & Business Media, 2009