A Channeled Multilayer Perceptron as Multi-Modal Approach for Two Time-Frames Algo-Trading Strategy

Noussair Fikri, Khalid Moussaid, Mohamed Rida, Amina El Omri, Noureddine Abghour
LIMSAD Labs, Faculty of Sciences, Hassan II University of Casablanca
Km 8 Route d'El Jadida,Casablanca, Morocco

***Abstract*—FOREX (Foreign Exchanges) is a 24H open market with an enormous daily volume. Most of the used Trading strategies, used individually, are not providing accurate signals. In this paper, we are proposing an automated trading strategy that fits random market behaviors. It is based on neural networks applying triple exponential weighted moving average (EMA) as a trend indicator, Bollinger bands as a volatility indicator, and stochastic RSI as a momentum reversal indicator to prevent false indications in a short time frame. This approach is based on trend, volatility, and momentum reversal patterns combined with a market adaptive and a distributed multi-layer perceptron (MLP). It is called channeled multi-layer perceptron (CMLP) that is a neural network using channels and routines trained by previous profit/loss earned by triple EMA crossover, Bollinger Bands, and Stochastic RSI signals. Instead of using classic computations and Back-propagation for adjusting MLP parameters, we established a channeled multi-layer perceptron inspired by a multi-modal learning approach where each group of modalities (Channel) has its K_c That stands for a dynamic channel coefficient to produce a multi-processed feed-forward neural network that prevents uncertain trading signals depending on trend-volatility-momentum random patterns. CMLP has been compared to Multi-Modal GARCH-ARIMA and has proven its efficiency in unstable markets.**

Keywords*—FOREX; neural networks; EMA; Bollinger band; stochastic RSI; momentum reversal; MLP; back-propagation; feed-forward

I. Introduction

Financial markets are where securities trades occur, including the forex market, stock market, bond market, and derivatives market. Financial markets are the primary source of liquidity for businesses. Our paper focused on the most common and liquid financial market, which is the foreign exchange market. F.X., Forex, or Foreign Exchange trades one currency for another, for example, GB Pound vs. U.S. Dollar, known as GBP/USD instrument[1]–[3]. Forex has no physical location; it's an electronic market, technically an extensive network of financial institutions (banks and brokers) and individual traders that operate through brokers or banks. There is a significant amount of data generated by market prices movements. From the big data view, it represents a rich source of features for predictions[3]. Algorithmic trading implements manually used strategies to predict when prices go down and prices go up, then place buy or sell trades with significant consideration of taking profit and stopping loss parameters for the best portfolio management. Machine learning is the best way to build a robust algorithmic trading platform by retrieving historical market data, selecting efficient features then training machines for future market movement prediction[1], [4]–[6]. To predict market prices, traders use several technics to have an accurate expectation on future trending. The most used technics are based on detecting trading signals from trends [5], [7]–[9], Volatility, or Momentum behavior[10]–[14]. For trend, the well-known technic is moving average; it has three famous alternatives; the first is Simple Moving Average[1], [15], [16], it uses the last N values average. An enhanced version of moving average is exponential weighted moving average[17], [18]; instead of assigning the same weight to all values, EMA gives more weight to last values. This technique has evolved, and traders now use a triple EMA with different selective periods to predict market prices accurately. Especially when there is a crossover between the 3 EMA lines, it's interpreted as a signal to place a trade that is not consistently profitable, and it's called a false signal. False or True profitable signals follow a different pattern that distinguishes their behavior. For volatility, the well-known method is the Bollinger Band which uses the Simple Moving Average as anchor line and the standard deviation as an indicator of aggressive markets movement[12], [19], [20]. Then finally momentum that has a famous method, the Stochastic Relative Strength Index that shows the market position reversal[21]–[23].

To discover the winning patterns based on cited indicators, we need to parse historical market data and train a neural network as the chosen algorithm of the machine learning branch. There are a lot of neural network types. The convolutional neural network, which belongs to the deep learning class, is used for visual imagery[24]–[26], and the recurrent neural network, which performs in natural language processing[18], [27]–[29]. In our case, we are dealing with a dataset of time series data (Financial market prices) processing and produce a binary decision: Place a profitable trade or Place none profitable trade[30], [31]. We choose a MultiLayer perceptron as a feed-forward neural network by using a back-propagation algorithm[24], [32]. On our approach, the features are calculated previously from historical data. We combined 3 EMAs and their growth rates as trend features, Bollinger Bands as volatility features, and Stochastic RSI as momentum features to enhance the accuracy of predicted decisions from current patterns and follow an adaptive model based on

different criteria (Trend- Momentum - Volatility). Let's go back to the existing state of the art. Several methods offer almost the same deep learning architecture but ignore the leading indicators of the instability of a specific market. Among these methods, we will cite the way that predicts price changes using LSTM [33]. Then a compound method consists of producing features based on GARCH to frame the data that describe volatility and their link to price variation and then inject them as input to an LSTM like N.N. to predict the subsequent variations. We will also cite a mixed method to prepare trend and seasonality data using the ARIMA approach and then inject data on an LSTM to predict the next trend. And finally, couple the last two methods quoted on a Multi-modal architecture, where the first modal is the LSTM-GARCH, and the 2nd modal is the LSTM-ARIMA. That will allow us to create the perfect hybrid model to benchmark our Channeled Multilayer Perceptron approach.

II. Related Work

A. Basic Trading Strategies

Day trading is an activity that involves entering winnings of less than ten pips per transaction on windows with 1, 5, and 15 minutes; this technique is called scalping. The simple moving average is the most popular among the statistical methods used to predict future prices, and it's an essential technique to describe a specific market trend [1], [15]. The first method we will study is the Simple Moving Average (SMA) which is the average of the previous N last prices, as shown on the following statement:

$$SMA = \frac{\sum_{i=0}^{N} Price_i}{N}$$

Then we have an Exponential moving average (EMA) that works the same as a simple moving average, except it places greater weight on the more recent closing prices[13], [15], [17].

$$M = \frac{2}{N+1}$$

$$EMA = M \times (ClosingPrice - EMA(Previous\ Day)) + EMA(Previous\ Day)$$

The difference between the simple and exponential moving average is that the EMA is smoother than the SMA; it is safer to use the EMA as a trend indicator. However, there is a more accurate technique to predict the trend of a specific financial instrument. It is called EMA crossover. It can be used with 2 or 3 different periods and detect their curve crossing. In the case of 2 respective curves with two periods, if the shortest EMA period crosses the more extended EMA period from bottom to top, it is considered an uptrend signal. If it passes from top to bottom, it is regarded as a downtrend. In the case of 3 periods, if the shortest EMA period crosses the long EMA period from bottom to top and both are above the longer EMA period; this is considered a more accurate up-trend signal. And if it goes from top to bottom and both are below the longer EMA period, this is considered a more accurate down-trend trend signal.

We also have a Moving Average Divergence and Convergence (MACD) trading strategy that uses 3 AMS with fixed windows of 26 EMA periods, 12 EMA periods, and 9 EMA periods overlapping [21] [22]. It reads positive and negative movements based on a zero line. The buy signal is triggered when the MACD passes over the zero line from the bottom, considered an uptrend, and sells where it crosses the signal line below, considered a downtrend. The take profit is triggered when the MACD falls below the signal line if a purchase order is placed. And if a sell order is placed, the profit-taking signal is triggered when the MACD rises above the signal line. The momentum of time series assumes that a market that behaves poorly with losses or profits over specific periods will continue at the same rate. It is based on the average log of N periods and its measure of performance above or below a red line on which the position is defined as positive or negative.

For the measurement of volatility, the chosen method is the Bollinger band which uses the simple moving average as an anchor line and the standard deviation as an indicator of aggressive market movement. It has been developed and protected by the copyright of John Bollinger, a trading expert, to provide meaningful insights into the specific volatility of the market [12], [19]–[21]. Thus, the first step in the calculation of the Bollinger band is to obtain the typical prices of the chosen period, then the standard deviation over the same period, and finally to calculate them with the simple moving average of the upper and lower bands, as follows:

$$BBUpper = SMA(TP, n) + \left(m \times \sigma(TP, n)\right)$$

$$BBLower = SMA(TP, n) + \left(m \times \sigma(TP, n)\right)$$

With:

- BBUpper: Upper Bollinger Band
- BBLower: Lower Bollinger Band
- SMA: Simple Moving average
- TP (typical price): (High+Low+Close) ÷3
- N: Number of days in smoothing period (typically 20)
- m: Number of standard deviations (typically 2)
- σ[T.P.,n]Standard Deviation over last n periods of T.P.

Fig. 1 shows the produced Bollinger lower and upper bands using 20 as a number of smoothing days and two as standard deviation:

Fig. 1. Plot of Bollinger Band using as 20 Smoothing Periods and 2-Period Standard Deviation.

And finally, The Stochastic RSI (Stochastic Relative Strength Index) is an application of a stochastic oscillator on a set of RSI[22], [23]. It varies between 0 and 1 or 0 and 100 percent to indicate an overbought or oversold of a specific market, in other words, a momentum position reversal. It is calculated on N-period by using two steps as follows:

$$RSI = 100 - \left[\frac{100}{1 + \frac{Previous\ average\ gain * N + Current\ gain}{Average\ average\ loss * N + Current\ lossPrevious}}\right]$$

Then:

$$StochRSI = \frac{RSI - MIN(RSI, N)}{MAX(RSI, N) - MIN(RSI, N)}$$

Fig. 2 shows the produced 80-Overbought and 20-Oversold lines of a 14-period stochastic RSI:

B. An Overview

1) Existing approach for prediction: There are several machine learning techniques, ranked according to their objectives, such as fault detection for unusual data points using One-class SVM [16], [33], [34], which can be a good solution if there are more than 100 features, with aggressive boundaries. We can also use the detection of anomalies by PCA in case we need fast training [13], [35], [36], or Clustering to discover the structure using the algorithm K-means if we want to label data that belong to a specific category or group [14]. But this cannot be used in our study because any market behavior is not necessarily an anomaly; the appropriate indicator must consider any aberrant data. Otherwise, for the prediction of continuous values, the best approach is linear regression if we follow a linear model [4], [14], [37], otherwise, for a two-class or multiclass binomial prediction, the classic approach is logistic regression, but we prefer to use a neural network.

Fig. 2. Plot of 14-Period Stochastic-RSI on Short-Time Frame.

For the classification to two classes and multiclass [38], [39], we can mention the decision tree, which has the ability of interpretation and categorization by following a binary or multiclass configuration [40], [41]. It can even use random forests and gradient-boosted tree algorithms as sets of trees to optimize classification and regression processes [42], [43]. There is also the Support Vector Machine or Networks (SVM) [16], [34], [44], which is a combination of supervised learning models and trained learning algorithms. It depends on partitions of spaces of high or infinite dimensions defined by parallel lines called hyperplanes, or maybe a set of hyperplanes with an operating margin quantified by distance to learning data points; the smaller the margin, the greater the generalization error of the classifier. Other methods, such as the One-vs-Rest classifier and naive Bayes, do not correspond to our case study.

2) Convolution neural network: In complex images, for example, face recognition, pictures have pixel dependencies where CNN is more accurate in prediction [24], [26]. Therefore, the prominent role of the convolutional network is reducing the complexity of images input features without losing accuracy. As shown in Fig. 3, a convolutional neural network is composed of, Convolution Layer, Pooling Layer, Flatten Layer, Fully connected layer, and SoftMax Layer [24], [25], [26], [30]:

3) Recurrent neural network: RNN is a feed-forward neural network that remembers the past. It takes what is learned from previous training iterations in its memory and uses it for the next predictions. RNN accepts one or more input vectors then produces one or more output vectors. It's not limited to associated weights but also hidden vectors which are considered as its memory. For each X input with an associated weight W_x there is a hidden node H with an associated weight W_y and an output Y with associated weight Y. A recursive process is applied on hidden states, accurately on H weights and remembering H and W_h values. The exact process is used on deep RNNs, weights are recursively updated in each iteration, and then more iterations are running, we go deep in learning, the reason we call it deep RNNs[18], [27]–[29].

4) Basic multilayer perceptron: MLP applies activation function where the primary is introducing the non-linearity to neural network outputs. Adding bias to this function provides additional trainable weight to a fixed value (Constant) that shifts output curves depending on the inputs group[30]. These are three known activation functions defined by the following separated expressions:

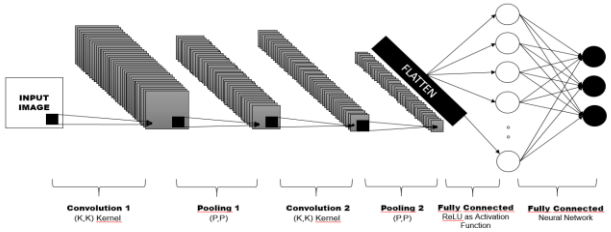

Fig. 3. Full Representation of Convolutional Neural Network.

$$Z = \sum_{i=0}^{n} W_i x_i + B$$

The Sigmoid function that computes the real value to range it between 0 and 1 as follows:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

The tanH function that computes the real value to range it between -1 and 1 as follows:

$$\tanh(z) = 2\sigma(2z) - 1$$

The ReLU, which means Rectified Linear Unit, computes real-values to replace negative values by 0 as follows:

$$F(z) = \max(0, z)$$

Feed-forward NN is the simplest form of ANNs [6], [23], [28]. It's composed of multiple neurons or nodes organized as layers with connections or edges between them containing weights of each node. After the first iteration comes the back-propagation algorithm to adjust the random weights associated with N.N. training ignition. By knowing the targeted value and the outputted value of the activation function, we get the difference between them as an error value to use back-propagation to calculate the gradients and adjust weights as optimal as possible by using a derivative function [6], [18], [26], [32], until error is reduced as shown on the following Fig. 4:

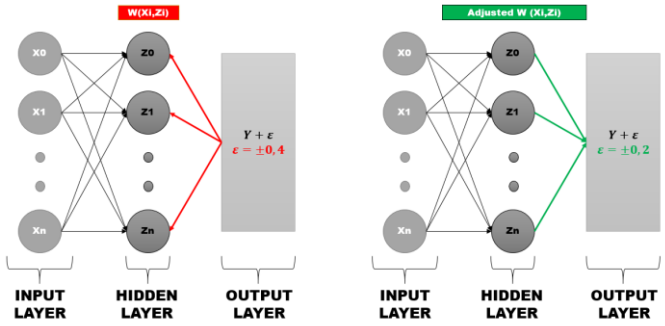

Fig. 4. A Preview of the Back-Propagation Process.

C. Available also-Trading Approaches

Several proposed approaches combine machine learning and forex market prices prediction, especially neural networks, to predict a trend of a specific instrument (e.g., GBP/USD). Most of them choose multi-layer perceptron, which seems to have the necessary ability and accuracy to recognize an effective trending pattern. Zhai, Hsu, and Halgamuge (2007) [8] choose a combination of market-related news and technical indicators delivered as features to a Support Vector Machine (SVM) based prediction model. Leslie Tiong Ching Ow [4] found a technic of trend prediction based on considering linear regression line (LRL) of the prices of timeline and trend line, in other words, the distance between the two lines, as a pattern of a possible up-trend or down-trend. Then use these patterns to train a Multilayer Perceptron for predicting future market trend movement. Gene I. Sher [32] based his method on chart pattern analysis as the 2D picture by submitting picture data as features to a feed-forward neural network and training it to recognize the winning geometric patterns. Svitlana Galeshchuk [34] used macroeconomic factors such as (GDP growth, unemployment, wages), current and capital account(current account balance, openness as ratio of total import and export to GDP), public and private foreign debt, capital flows, and the ratio of international reserves to 3 months import, global variables (interest rates and price ratios), as input features to a deep neural network. This method is focused mainly on a novel feature based on currency clusters, but it's also useful for emerging currencies markets. This approach is based on Autoregressive Integrated Moving Average (ARIMA) models and Exponential Smoothing (ETS) models as times series models for its analysis and Stacked Long Short-term Memory for features processing and future stock prices prediction. Iman. Zabbah [35] enhanced an automated strategy based on EMA crossover, with 2 EMA lines of 2 different periods in a short time frame. This strategy is also based on Multilayer perceptron with linear scaling between the minimum and maximum values, as input features, to avoid false signal that triggers losing orders and distinguish winning EMA crossovers.

Nathan D'Lima [36] also chooses the same approach as Svitlana Galeshchuk [33] by basing their analysis on the hypothesis that time-series behavior is influenced by economic, political, and psychological factors. And he proposed a solution that combines a Multi-layer perceptron (MLP) with back-propagation and Adaptive Neuro-Fuzzy inference System (ANFIS). First, the MLP takes as inputs the previous day closing prices, simple moving average (SMA), exponential moving average (EMA), and Rate of Change (ROC) as momentum oscillators. And the ANFIS takes the previous day's closing prices and SMA. Then Mean Square Error (MSE) and Mean Absolute Error (MAE) for performance metrics.

On the other hand, Zhixi Li and Vincent Tam [10] found research on momentum and reversal effects unconvincing. This is because they cannot be applied as a trading strategy on a real-world investment portfolio, and the reason was the lack of predictive ability. So, they make a deep comparison between Decision Tree (D.T.), Support Vector Machine (SVM), Multilayer Perceptron Neural Network (MLP), and Long Short-Term Memory Neural Network. For them, SVM can be a source of profitable trading strategies.

D. Neural Network-based Approaches

In the same radius, we can mention Svetlana Borovkova and Ioannis Tsiamas proposed a group of Neural Networks of Long-Short-Term Memory (LSTM) for intraday trading based on technical pre-scan analysis [37]. This approach is quite good, but it does not consider the functional parameters: Momentum reversal cases, massive trend following, and volatility of the chosen market, which is the basis of our motivation. This technique works by weighting individual models in proportion to their past performance, permitting them to deal with possible non-stationarities by using a new approach. The sector then measures model performance under the operational characteristics of the receiver. Finally, they evaluate the predictive power of their model on several high-

value U.S. stock prices-cap and compare it to Lasso and Ridge. The proposed model was better than either the reference model or the equally weighted reference model sets. But this will not stop us from comparing the approach to the LSTM-based methods.

Seyed Taghi Akhavan Niaki and Saeid Hoseinzade study show the prediction of the daily direction of (S & P 500) index using an ANN. The main purpose is to select the most influential (features) characteristics of the proposed ANN that affect the day-to-day direction of S& P 500 (response). Experiments were conducted to find correct parts among 27 potential variables, projected as input nodes of the trained ANN. Using this methodology, the results of this study, using the most accurate features, can predict the daily S&P 500, which is much better than the existing logic model. In addition, the experimental results from the use of studied ANN on trial trading could significantly increase trading benefits by using a buy-and-hold strategy [38]. This has the same empiric context as our approach but does not fit our model for one reason: market behavior. This technique can be effective on trend following or momentum reversal but is not adaptable to volatile markets; the S&P 500 is a long-term trading index.

In their paper, Bang Xiang Yong, Mohd Rozaini Abdul Rahim, and Ahmad Shahidan Abdullah have proposed a trading system using a deep neural network (DNN) to predict the Singapore Stock Exchange Index (SESI) prices. Using the FTSE Straits Time Index (STI) in the coming days, test it through market simulations on historical daily prices [39]. The DNN has 40 nodes as the input layer, the last ten days, opening, closing, minimum and maximum price, and consists of 3 hidden layers with ten neurons per layer. It is trained by using stochastic gradient descent with backward propagation performed by multicore processing. They evaluated this approach using RME and MAPE and found that the profit factor was up 70%. It is not recommended to compare this method with ours because the latter is somewhat long-term, whereas our approach is intraday and mainly looks at the volatility phenomena. But we are going to be inspired by this method because it is based on deep learning.

Shuanglong Liu chooses the combination (CNN-LSTM) convolutional neural network and long-term memory, Chao Zhang, Jinwen Ma, to produce a model and analyze quantitative selection in stock markets [40]. First, the CNN method is used to establish the quantitative strategy of stock selection and then the quantitative approach of synchronization to improve profits using the LSTM. Their experiments show that the CNN neural network model can be successfully applied in developing a quantitative strategy and producing better yields than the benchmark. This approach is quite far from our study and cannot be compared with our practice due to its use of quantitative selection, but still one of the approaches that strands among state-of-art.

In their study, Wing Ki Liu and Mike K. P. incorporated the GARCH method on a neural network recurrent to model the price index's volatility on the S&P 500 financial market [41]. They produced a hybrid LSTM-GARCH model because it can consider the potential non-linearity of volatility at any time (t) and then predict the future volatility of the target index. This model is very similar to one of the channels of our CMLP, which led us to integrate it as a modal on an N.N. architecture for comparison.

On another radius, precisely on the study of the trend, G. Peter Zhang has set up a hybrid model ARIMA-LSTM, which will train an RNN based on three main features, the trend $\boldsymbol{m_t}$, the seasonal component $\boldsymbol{s_t}$ and the noise $\boldsymbol{\epsilon_t}$. This model represented the 2nd modal on the N.N. architecture of comparison [42].

E. Similar N.N. Architecture: MultiModal Learning based

Our case study is near Multi-Modal Deep Learning-based studies, which involve data from different input models by combining distinct modalities or groups of information with evident quantitative influence over the prediction output.

Our world considers several modalities, and a modality is a dataset about a lived experience in a specific context in real life. Most research focuses on sensory modalities representing the classic case, mainly based on natural language, visual, and vocal signals. It's the best approach to help artificial intelligence understand the world around us and interpret multi-modal messages from each modality. The classic case, which uses Multi-modal learning, has five main challenges:

- Representation: Summarization of multi-modal data (Heterogeneity)
- Translation: Mapping and Relationships between modalities (Open-Ended or Subjective)
- Alignment: Mapping and Relationships between sub-elements
- Fusion: Joining two or more modalities
- Co-learning: Knowledge transfer between modalities

There are two categories of multi-modal representations: joint and coordinated [43]. The collaborative approach consolidates unimodal signals into one unique model, shown in Fig. 5, and following this equation:

$$x_m = \mathrm{f}(x_1, \ldots, x_n)$$

Whereas coordinated handle unimodal signals separately, as shown in Fig. 6 and the following equation:

$$\mathrm{f}(x_1) \sim \mathrm{g}(x_2)$$

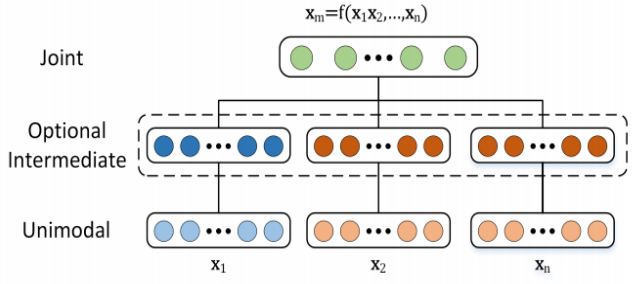

Fig. 5. Multi-Modal Joint Representation.

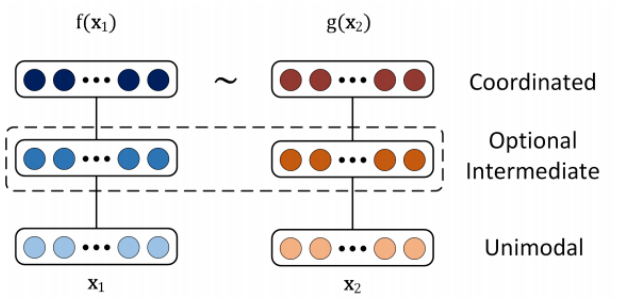

Fig. 6. Multi-Modal Coordinated Representation.

The generalized autoregressive conditional heteroskedasticity (GARCH) approach is used in financial markets volatility estimation. This technic is combined with intrinsic mode functions derived from empirical mode decomposition as a hybrid neural network [44]. They proposed an intelligent system that has the same properties as a multi-modal learning approach.

F. CSP, Share Memory by Communicating and Channeling

Communication is an exchange of specific data or explicit messages between specific N processes (explicit messages are used in distributed systems against shared memory systems). Messages are sent in two different ways, the asynchronous way that allows synchronization between emitter and receiver. The emitter process must wait for its ACK (acknowledgment) message to be received [45], [46]. And asynchronous way where the emitter does not wait for ACK and hand over to next process through shared variables which each process had public access to (used by standard memory systems) or by RPC (Remote Procedure Call). Synchronization links one process to another process and performs the exchange of control information between processes. In other words, if a process needs data produced by another process and this specific data is not available, the process must wait until it's available. The big challenge in process interleaving is their concurrency on data is that we must use an excellent technic to keep our system runnable. An interleaving of two sequences, s, and t, is a constructed U sequence from the events of s and t so that the events of s preserve their order in U and those of t. Hence, the birth of CSP is a method of communication and synchronization by using shared memory in multiprocessing. Instead of communicating by sharing memory (the traditional threading model used by famous languages like C and C++), based on sharing their data structures in memory, then locking used resources for a thread-safe mode-based communication between threads and mainly the processes. We will focus on the model-based approach used by Golang, which is sharing memory by communication. This concept is based on CSP [46] for its implementation using goroutines and channels as programming primitives. Goroutines are based on dynamic stack size, the reason why we called them "Green thread," the size of each dedicated segment on processes stack grows as needed depending on the data used by current tasks. Instead of the classic approach that uses a fixed stack size. This new technic allows us to create an infinite number of concurrent tasks. Goroutines start quickly instead of threads start-up and come with integrated primitives that allow them to communicate safely: Channels. There is Multi-Core Parallel Programming in Go demonstration [45], [47], [48] based on parallel integration for P.I. calculation and shows its performance by using a different number of parallel cores. We will build our trading architecture in charge of data retrieving, features extraction, and decision adaptation using a Goroutine-based implementation that fits our Channeling approach.

III. METHOD

Our approach is based on channeled multi-layer perceptron, in other words, building a feed-forward neural network that has a channel for each features group as follows:

A. Trend Features Group for T-Channel

Based on 3 EMAs (Exponential Moving Averages) on what we call an anchor time frame, in our case, a 1 Hour as long-term time frame. The main reason behind this is that there is less volatility on an H1 time frame and gives a safe deduction on market trend if it's in an uptrend or a downtrend. For trend determination, we use 3 EMA of different periods. Of course, we can use another period in our case (8, 13, 21), but those stay optimal for our approach. We also use the crossover to define the current hour trend, and the best method for this is 3 EMA crossover as cited in related works. After producing the 3 AMS data, we must select the qualified characteristics to be submitted to the trend channel. In this case, the growth rate of each EMA and the status ({1} for above - {-1} for below - {0} on the same line) between the first EMA (8 as a period) and the second (13 periods) the status between the second and third (21 periods), as shown in Table I:

TABLE I. SAMPLE OF TREND FEATURES GROUP FOR T-CHANNEL

Row	time	ema_1_gr	ema_2_gr	ema_3_gr	ema_1_2	ema_2_3
0	05:00	-1.23445	-0.7715	-0.54004	0	0
1	06:00	0.92575	0.46288	0.30858	-1	-1
2	07:00	0.69427	0.46286	0.30857	0	-1
3	08:00	3.16177	2.08242	1.31126	1	1
4	09:00	2.39004	1.6965	1.15686	1	1

B. Volatility Features Group for V-Channel

Based on the Bollinger Bands discovered by computing a 2-standard deviation of 20 periods SMA (Simple Moving Average) on an active trading frame, a 1 minute as a short time frame, where the volatility is considerably high. The standard deviation gives us a safe view of specific financial instrument volatility if it's stable or ultra-active. Technically, the standard deviation produces two bands, an upper and a lower band. If the upper band and lower band are far from the SMA, this means two things, the chosen financial instrument

is too volatile, and any crossing between the closing price and the upper or lower must be taken as trend reversal. After computing the 20-Period SMA and 2-Standard-Deviation, we choose the qualified features for the volatility channel. The distance between closing price and up or down band, and the distance between SMA and the up or down band, as shown on Table II:

TABLE II. SAMPLE OF VOLATILITY FEATURES GROUP FOR V-CHANNEL

Row	time	price_up_bb	price_low_bb	sma_up_bb
19	12:26	0.00129	0.00315	0.00222
20	12:27	0.001	0.00268	0.00184
21	12:28	0.00086	0.00226	0.00156
22	12:29	0.00065	0.00187	0.00126
23	12:30	0.00084	0.00124	0.00104

C. Momentum Features Group for M-Channel

Based on Stochastic RSI (Relative Strength Index) as cited in related works, a stochastic oscillator is applied on a set of RSI. The main goal behind using Stoch RSI is to detect the position reversal of a specific financial instrument by observing the crosses between RSI lines and oversold or overbought lines. We applied this on an active trading frame, in our case, a 1 minute as a short time frame. After computing the Stock RSI with {14, 14, 3, 3} as parameters, which are defined for our case. The distinguishing features for M-channel are the distance between RSI line 1 and oversold line, the distance between RSI line 1 and overbought line, the distance between RSI line 2 and oversold line and finally between RSI line 2 and overbought line, as shown on Table III:

TABLE III. SAMPLE OF MOMENTUM FEATURES GROUP FOR M-CHANNEL

Row	time	L14	H14	K_80	K_20	D_80	D_20
15	15:06	1.30374	1.30476	17.0	77.0	8.33	68.33
16	15:07	1.30378	1.30476	2.0	62.0	7.33	67.33
17	15:08	1.30378	1.30476	-3.0	57.0	5.33	65.33
18	15:09	1.30388	1.30476	5.0	65.0	1.33	61.33
19	15:10	1.30388	1.30476	12.0	72.0	4.67	64.67

D. Channeled MLP for TVM (Trend – Volatility – Momentum) Features Groups

As shown in Fig. 7, the main goal behind channeling a Multilayer Perceptron is to add a functional weight to each feature group and then parallelize their execution to enhance the speed of training and execution. In our case, we have three groups of features as follows:

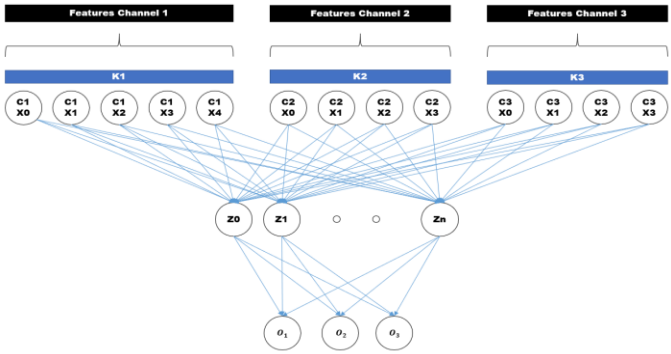

Fig. 7. Channeled Multi-Layer Perceptron Architecture Applied on Three Specific Channels.

- Trend group (Channel 1): composed of 5 features ema_1_gr, ema_2_gr, and ema_3_gr, which are the growing rate of EMAs then ema_1_2 and ema_2_3 which are respectively the EMA 1 / EMA 2 and EMA 2 / EMA 3 crossing status. We assign K1 to this channel as the initial weight.
- Volatility group (Channel 2): composed of 4 features price_up_bb, price_low_bb, sma_up_bb, and sma_low_bb, which are respectively the distance between price and upper band, the distance between price and lower band, the distance between SMA and upper band, and the distance between SMA and lower band. We assign K2 to this channel as the initial weight.
- Momentum group (Channel 3): composed of 4 features K_80, K_20 which are RSI line 1 and oversold line, the distance between RSI line 1 and overbought line, then D_80 and D_20, which are the distance between RSI line 2 and oversold line and finally between RSI line 2 and overbought line. We assign K3 to this channel as the initial weight.

The expected outputs are O1, O2, and Q3, respectively Buy signal, Sell signal and Wait for Signal. These outputs result from expected historical profits in the past. If the max profit is greater than 1, we consider it as a buy signal which has {1} as an expected feature. If the min profit is less than -1, we consider it as a sell signal which has {2} as the expected value. In the case where profit is between -1 and 1, the expected value is the wait signal which has {3} as the expected value.

E. Channeled MLP Execution (1 Iteration of Training)

Fig. 8 shows the execution of a Multi-layer perceptron using our approach, channeling a feed-forward neural network to parallelize each node's computation from the input layer to the output layer. It is considered that we have two channels which are composed of $(X_0, X_1, X_2, \dots, X_n)$ features, and are linked to each node of the hidden layer with $(W_0, W_1, W_2, \dots, W_n)$ weights. There are 2 (Depending on the number of channels) concurrent routines in charge of summation for each hidden layer node by respecting each channel weight. When summation of each channel is completed, a signal is triggered to start submission of sums to

activation function to produce (In the first iteration of the MLP training) Y + $\boldsymbol{\varepsilon}$ with Y as output and $\boldsymbol{\varepsilon}$ as an error. In our case, the used activation function is Sigmoid, and the first iteration is computed by using a random parameter for each node.

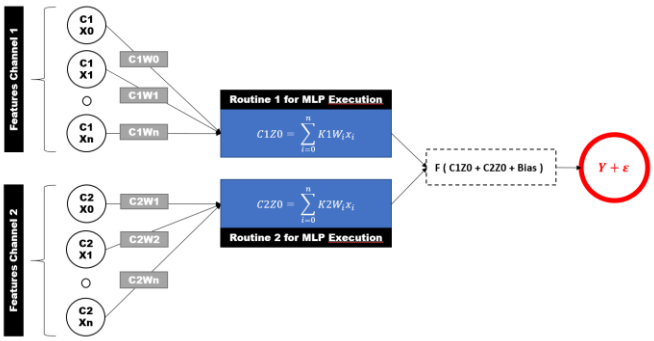

Fig. 8. Channeled MLP Executed by Channels Dedicated Routines.

F. Channelled MLP Back Propagation (Sigmoid Derivative Function)

In Fig. 9, we will see how back-propagation of produced error is done using our approach, which is based on channeling the propagation of error and applying it on the delta of each node. For each output node, there is a set of routines dedicated to each node of the hidden layer to compute the delta by using the derivative of the sigmoid action function, as shown in the Routine H_0 for Z_0 delta calculation. Then for each hidden layer node, there is a set of routines which is respectively dedicated to each node of the input layer to compute the delta by using the derivative of the sigmoid action function, as shown in the Routine I_0 and I_n for respectively X_0 and X_n Delta calculation. The process can be reproduced if there is more than one layer in the hidden layer. For concurrency synchronization, the computation of the previous layer must wait for the current layer computations to be completed.

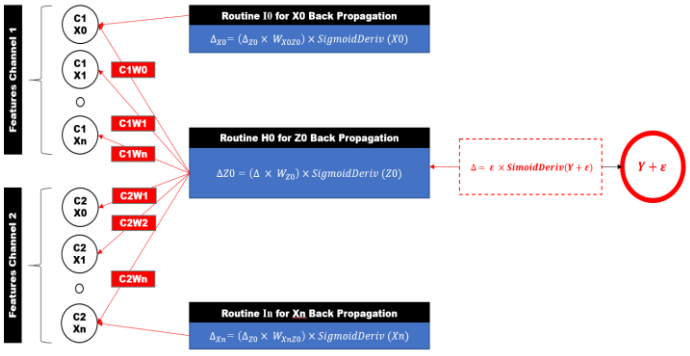

Fig. 9. Channeled Back-Propagation using Layer Set of Routines.

G. Channeled MLP Weights Adjustment (using Learning Rate)

Fig. 10 shows how weights adjustment is made for each node using our approach, based on channeling this process for each iteration. For each output node, there is a set of routines dedicated to nodes of the hidden layer to compute the learning rate, the delta, and the value of each node. We can see this in the Routine H_0 for Z_0 new weight calculation. Then for each input node, there is a set of routines that are dedicated to each node of the input layer to compute the learning rate, the delta, and the value of each node, it's represented by routine I_0 and I_n which are respectively in charge of W_{x_0} and W_{x_n} New weight calculation. The process can be reproduced if there is more than one layer in the hidden layer. However, for concurrency synchronization, the computation of the previous layer must wait for the current layer computations to be completed.

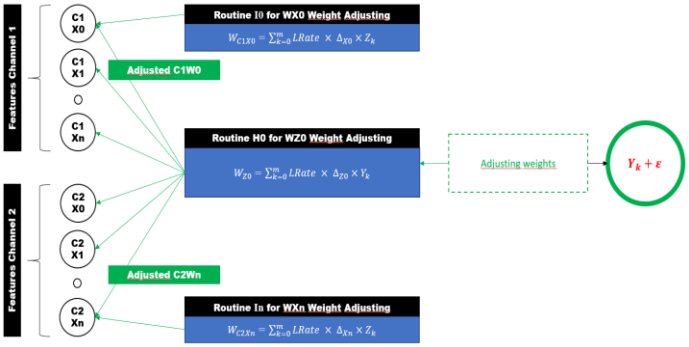

Fig. 10. Channeled Weights Adjustment.

H. Targeted Comparison Model: LSTM, GARCH, and ARIMA

To compare our model, as shown in Fig. 11, we will use a multi-modal neural network based on two different modals:

- The first modal is an LSTM ANN that will have input data from the GARCH approach, representing the study of volatility considering the historical returns of the preceding periods.
- The second modal is also an LSTM that will inject data from the ARIMA study that represents the trend of the target market.

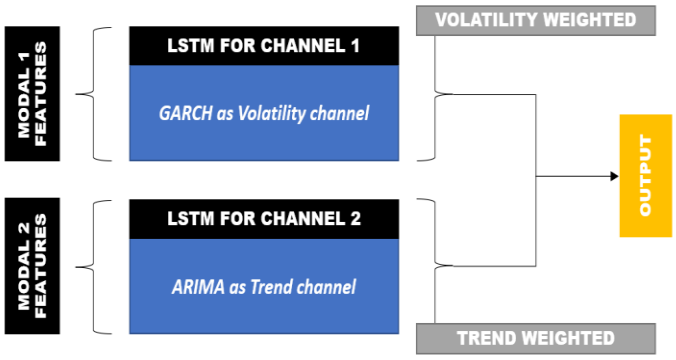

Fig. 11. GARCH-ARIMA MultiModal based on LSTM.

About the GARCH modal, this study follows a hybrid model: LSTM model and GARCH framework. The model first defines that the return $\boldsymbol{p_t}$ follows a standardized t distribution with conditional variance $\boldsymbol{h_t}$.. The conditional variance $\boldsymbol{h_t}$ is then modeled by LSTM. The objective function for the estimation of the parameter is the negative

loglikelihood. The GARCH model is conventionally used for profit modeling because it takes into account the dynamics nature of the conditional variance of performance. The GARCH(1,1) model can be expressed as:

$$p_t = \epsilon_t \sqrt{h_t} \quad | \quad \epsilon_t \sim N(0,1)$$

$$h_t = \alpha_0 + \alpha_1 h_{t-1} + \beta_1 p_{t-1}^2$$

$$\alpha_0 > 0 \quad | \quad \alpha_0, \alpha\beta_0 \geq 0$$

The stationary condition of the GARCH model (1,1) is:

$$\alpha_1 + \beta_1 < 1$$

In 2 first formula, the profit at time t, $\boldsymbol{p_t}$, follows a normal distribution with a zero mean and an $\boldsymbol{h_t}$variance, given the information up to time t-1. The variance at time t is the linear combination of conditional and square returns at time t1. The architecture of $\boldsymbol{h_t}$ is illustrated in Fig. 12. In the model, we will use the respective values of the previous **m** periods (x_{t-m}, ..., x_{t-1}) to predict the conditional variance of current h_t Returns.

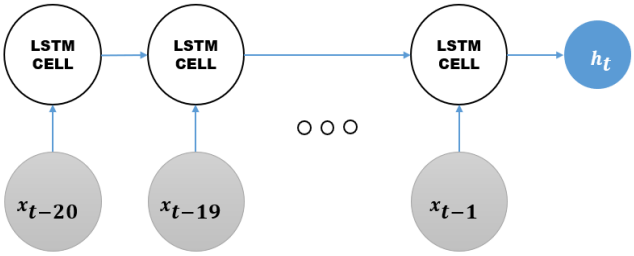

Fig. 12. GARCH based LSTM for Volatility.

Next comes the modal ARIMA/LSTM, a hybrid model where the study focuses mainly on the trend. This will be based on the variations of the triplet (m_t, s_t, ϵ_t) to predict m_t, this is a fairly simple model to consider the trend of the target market, specifically on forex, where the behavior does not always follow a uniform trend. The idea is to transform the original y_t Time-series into a smooth function by isolating it from its seasonal sound. To do this, we proceed to the first step of the Box and Jenkins method, which is $y_t = m_t + s_t + \epsilon_t$ Decomposition. Three functions are obtained: trend m_t, seasonal component and ϵ_t Noise. s_t and ϵ_t are kept to reproduce seasonality and put the forecast values into the confidence intervals using ϵ_t. From now on, m_t is smooth because it is free of fluctuation. Therefore, we can approach it through a network of neurons to know its dynamics and thus predict its future. Moreover, as it is a time series, the observations are sequential, so we use RNN of LSTM.

IV. Experiment

A. OANDA Broker – GBP/USD Instrument – REST API

For experimentation, we used OANDA as a broker due to available features for programmers. The most important feature for us is his REST API which provides access to historical data, price streaming, and orders management. The financial instrument used in this experiment is GBS/USD (Great Britain Pound against United States Dollar). The used OANDA API calls in our case are:

```
body=$(cat << EOF
{
"order": {
"units": {0},
"instrument": {1},
"timeInForce": {2},
"type": {3},
"positionFill": "DEFAULT"
}
} EOF
)
curl \
-X POST \
-H "Content-Type: application/json" \
-H "Authorization: Bearer <AUTHENTICATION TOKEN>" \
-d "$body" \
https://api-fxtrade.oanda.com/v3/accounts/<ACCOUNT>/orders
```

Where:

- {0}: Number of units to buy (Positive number) or sell (Negative number)
- {1}: Used financial instrument in our case GBP/USD
- {2}: In our case, we will use FOK (Fill Or Kill)
- {3}: The initial position which is set to default

B. Historical Data Retrieving then Features and Expected Values Production

This stage is dedicated to TVM (Trend-Volatility-Momentum) winning and losing patterns production. As a requests wrapper, we used a Golang based program, which is in charge of retrieving historical prices and computing them into usable features and expected values. This program takes the prepared features as a flat set of indicators for the first test, which is a normal MLP training, then taken as three groups (TVM) of features in the second test, which is the implementation of our approach, the channeled MLP training.

In the first stage, we must produce the T-CHANNEL (Trend Channel). In Table IV, we can see the growth rate and the status of each of the 3 EMA periods. Then profit max and min generated by each model. If the maximum profit is greater than {1.0}, we consider it as a buy signal that has {1} as an expected feature. Otherwise, if the minimum profit is less than -1, we consider it as a sales signal that has {2} as expected value:

TABLE IV. MAX AND MIN PROFIT THEN SIGNAL GENERATED FROM T-CHANNEL FEATURES

ema_1_gr	ema_2_gr	ema_3_gr	ema_1_2	ema_2_3	max_pr	Min_pr	Signal
-1.23445	-0.7715	-0.54004	-1	-1	1	-2	2
0.92575	0.46288	0.30858	-1	-1	0,3	-5	2
0.69427	0.46286	0.30857	1	-1	0,1	-1	2
3.16177	2.08242	1.31126	1	1	2	-0,1	1
2.39004	1.6965	1.15686	1	1	4	-0,1	1

In the second stage, we must produce the V-CHANNEL (Volatility Channel). In Table V, we see the calculated distance between prices or SMA up and low bands (Bollinger bands), then the max profit and the min profit generated in the case of trading on each pattern, if the max profit is greater than 1, we consider it as buy signal which has [1] as expected feature, if the min profit is less than -1, we consider it as sell signal which has [2] as the expected value. In the case where profit is between -1 and 1, the expected value is the wait signal which has [3] as the expected value.

In the third stage, we must produce the M-CHANNEL (Momentum Channel). In Table VI, we can see the calculated distance between K-Stochastic-RSI and D-Stochastic-RSI lines, and respectively 80% and 20% lines. Next, the max profit and min profit generated in the case of trading on each model. If the max profit is greater than 1, we consider it as a buy signal that has [1] as expected characteristic, if the min profit is less than - 1, we consider it as a sell signal that has [2] as expected value. Finally, if the profit is between -1 and 1, the expected value is the waiting signal that has [3] as expected value.

TABLE V. MAX AND MIN PROFIT THEN SIGNAL GENERATED FROM V-CHANNEL FEATURES

price_up_bb	price_low_bb	sma_up_bb	sma_low_bb	max_pr	min_pr	Signal
0.00129	0.00315	0.00222	0.00222	0,1	-0,1	3
0.001	0.00268	0.00184	0.00184	1	-0,4	1
0.00086	0.00226	0.00156	0.00156	0,1	-0,7	3
0.00065	0.00187	0.00126	0.00126	0,2	-0,3	3
0.00084	0.00124	0.00104	0.00104	0,7	-1	2

TABLE VI. MAX AND MIN PROFIT THEN SIGNAL GENERATED FROM M-CHANNEL FEATURES

K_80	K_20	D_80	D_20	max_pr	Min_pr	Signal
17.0	77.0	8.33	68.33	0,2	-3	2
2.0	62.0	7.33	67.33	0,1	-2	2
3.0	57.0	5.33	65.33	0,9	-0,5	3
5.0	65.0	1.33	61.33	0,7	-0,2	3
12.0	72.0	4.67	64.67	0,3	-0,01	3

1) Normal Multi-layer perceptron training

We used the current parameters for training our MLP:

- Global parameters (Using K-Fold training):
  - *a)* Learning Rate: 0.01
  - *b)* Accuracy percentage: 0.67
  - *c)* Shuffle: True
  - *d)* Epochs: 100/200/500
  - *e)* Folds: 2/3/5
- For trend features:
  - *a)* Granularity: H1 (1 hour)
  - *b)* Number of candle patterns: 500
  - *c)* Max/Min profit: 1 / -1
- For volatility features:
  - *a)* Granularity: M1 (1 minute)
  - *b)* Number of candle patterns: 500
  - *c)* Max/Min profit: 1 / -1
- For momentum features:
  - *a)* Granularity: M1 (1 minute)
  - *b)* Number of candle patterns: 500
  - *c)* Max/Min profit: 1 / -1

Table VII shows the list of features in the input layer:

TABLE VII. THE COMPLETE LIST OF FEATURES SUBMITTED TO NORMAL MLP

Feature name	Associated name	Node notation
Short Period EMA (EMA 1) growing rate	ema_1_gr	X0
Medium Period EMA (EMA 2) growing rate	ema_2_gr	X1
Long Period EMA (EMA 3) growing rate	ema_3_gr	X2
Common status between EMA 1 / EMA 2	ema_1_2	X3
Common status between EMA 2 / EMA 3	ema_2_3	X4
Distance between price and upper Bollinger band	price_up_bb	X5
Distance between price and lower Bollinger band	price_low_bb	X6
Distance between SMA and upper Bollinger band	sma_up_bb	X7
Distance between SMA and lower Bollinger band	sma_low_bb	X8
Distance between K-Stochastic-RSI line and 80%	K_80	X9
Distance between D-Stochastic-RSI line and 80%	D_80	X10
Distance between K-Stochastic-RSI line and 20%	K_20	X11
Distance between D-Stochastic-RSI line and 20%	D_20	X12

Table VIII shows the list of expected values (Classes) in the output layer:

TABLE VIII. EXPECTED VALUES ON THE OUTPUT LAYER

Expected output	Name of Expected output	Description
1	Buy signal	Triggered buy signal on expected max profit obtained
2	Sell signal	Triggered sell signal on expected max profit obtained
3	Wait for signal	Triggered wait signal on expected max profit not obtained

2) Channeled Multi-layer perceptron training:

We used the same parameters in normal MLP training, but we added another parameter which is the associated weight for each channel:

- For trend features: K1 as channel associated weight
- For volatility features: K2 as channel associated weight
- For momentum features: K3 as channel associated weight

The channeled features are distributed in Table IX, as follows:

TABLE IX. LIST OF THE FEATURES SUBMITTED TO CHANNELED MLP

Feature name	Associated name	Node notation	Channel	Initial weight
Short Period EMA (EMA 1) growing rate	ema_1_gr	C1X0	C1	K1
Medium Period EMA (EMA 2) growing rate	ema_2_gr	C1X1	C1	K1
Long Period EMA (EMA 3) growing rate	ema_3_gr	C1X2	C1	K1
Common status between EMA 1 / EMA 2	ema_1_2	C1X3	C1	K1
Common status between EMA 2 / EMA 3	ema_2_3	C1X4	C1	K1
Distance between price and upper Bollinger band	price_up_bb	C2X1	C2	K2
Distance between price and lower Bollinger band	price_low_bb	C2X2	C2	K2
Distance between SMA and upper Bollinger band	sma_up_bb	C2X3	C2	K2
Distance between SMA and lower Bollinger band	sma_low_bb	C2X4	C2	K2
Distance between K-Stochastic-RSI line and 80%	K_80	C3X0	C3	K3
Distance between D-Stochastic-RSI line and 80%	D_80	C3X1	C3	K3
Distance between K-Stochastic-RSI line and 20%	K_20	C3X2	C3	K3
Distance between D-Stochastic-RSI line and 20%	D_20	C3X3	C3	K3

3) Multi-Modal learning based on GARCH and ARIMA: As shown in Section 2, we decided to build a multimodal N.N. system based on generalized autoregressive conditional heteroscedasticity (GARCH) and integrated autoregressive moving average (ARIMA). This hybrid approach is already used in volatility prediction using the ARMA(R,M) and GARCH(p,q) models[48]. The first group or modality is the variance group representing volatility that belongs to GARCH, based on Gaussian pseudo-random numbers. This modality is composed of an average model, a volatility process and a distribution. The second modality that represents the trend group that represents the ARIMA model.

The most important part is the study on volatility. For the first GARCH-oriented LSTM intended to receive volatility-related data

- The features chosen as input are the elements having a studied correlation and which constitute the vector

$$x_t = \{o_t, g_t, l_t, c_t\}$$

where o, g, l, and c are in the same order open, high, low, and the close price at the time (t), then the parameter p_t^2 that will allow each LSTM node to model the h_t, based on conditional variance using the objective loglikelihood function.

- The exits of the LSTM are the following:

$$p_t = P_t - P_{t-1}$$

Where:

p_t is the profit at **t,** and

P_t is the price at **t**

4) Classical hybrid approach: Then combine Bollinger Band with Stochastic RSI and use it as a trading strategy on GBP/USD couple. Intraday traders widely use this technic to make an exhaustive comparison to our strategy and see the benefits of using a Channeled MLP as a validation layer on a switching pattern scenario to be adaptive to any kind of markets behavior.

V. RESULTS

The back-testing execution is done on a machine with Ubuntu 18.04 as an operating system and Docker containers for worker nodes to provide a simulated aspect of distributed computing. All of this is running on a physical machine with an Intel i5 2.2 GHz 4 CORE processor and 8 G.B. of RAM.

For our test case, we have retrieved GBP/USD data of 2-time frames, 1-hour and 1-minute prices, then calculated their 3 EMA crossovers, Bollinger Bands, Stochastic RSI, GARCH, and ARIMA as shown in the method section, to produce the five needed datasets. Then came the normalization process to ensure the quality and performance of training. We then randomly split data into two datasets, 70% for training against 30% for test.

In Fig. 13, we see the scores reached by implementing normal Multilayer Perceptron, the multi-modal GARCH-ARIMA, and channeled MLP. It shows the scores for 2, 3, and 5 folds for 100, 200, and 500 epochs as training iterations. Due to dedicated channel weights, the Channeled Multilayer Perceptron has better scores than normal MLP and MM GARCH ARIMA.

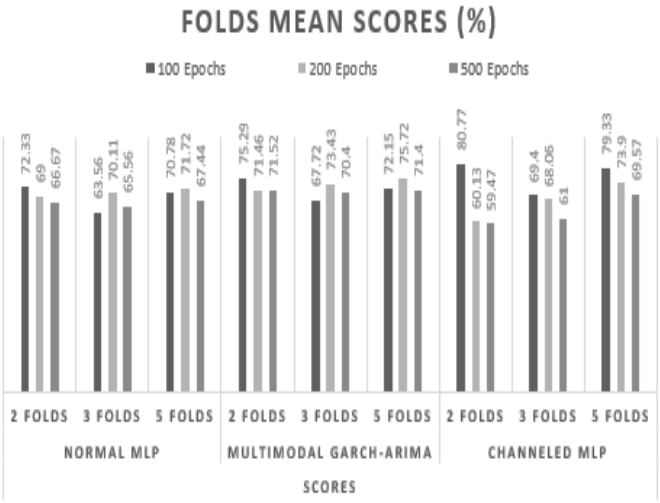

Fig. 13. Back-Testing Scores reached Comparison between Normal MLP, MM GARCH-ARIMA and our Channeled MLP.

In Fig. 14, we see the training time taken by implementing normal Multilayer Perceptron and channeled MLP. It shows the scores for 2, 3, and 5 folds for 100, 200, and 500 epochs as training iterations. Due to parallelized and concurrent node calculations, the Channeled Multilayer Perceptron has been trained faster than normal MLP and MM GARCH ARIMA.

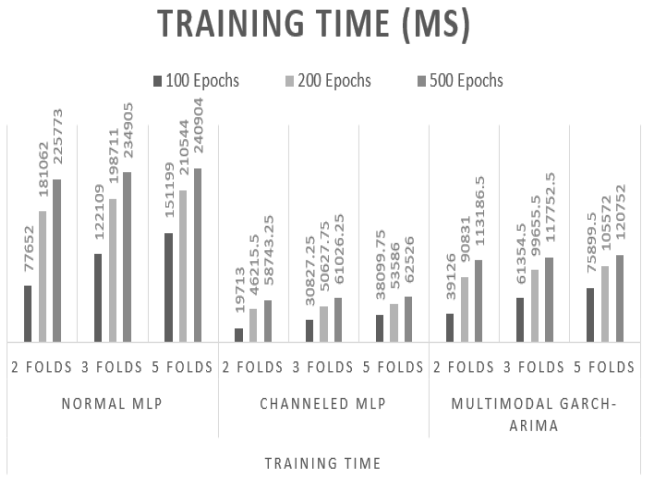

Fig. 14. Training Time Comparison between Normal MLP, MM GARCH-ARIMA and our Channeled MLP.

In Fig. 15, we see the execution time taken by implementing normal Multilayer Perceptron and channeled MLP. It shows the scores for 2, 3, and 5 folds for 100, 200, and 500 epochs as training iterations. The Channeled Multilayer Perceptron computes the given pattern and provides the associated output faster than normal MLP and MM GARCH ARIMA due to parallelized and concurrent node calculations.

In Table X, we observe the trades opened by normal Multilayer Perceptron, channeled MLP, and BB/SRSI, in the same day to see the efficiency of Channeled MLP decisions. The Channeled Multilayer Perceptron has opened six trades where 5 of them were positive with 9.7 cumulative earnings and a profit factor that reached 83%. The normal MLP has opened six trades where 3 of them had positive profits up to 2.15 as net profit. Then the MM GARCH-ARIMA opened six trades where 1 of them was positive with fatal negative profit (-14.3). The main problem was the lack of features that stands for momentum reversal in the market which does a lot of brutal reversal in its trend. And finally, BOLLINGER BAND / Stochastic RSI with six open trades where 1 of them was positive having 0.7 profit. Thus, we can say that channeled MLP made good decisions in opening market switching by trend, volatility, and momentum reversal in GBP/USD market.

We chose the MultiModal GARCH-ARIMA LSTM model to create a more appropriate and similar comparison context to our approach and thus target several existing methods cited in the state of the art, some of which are not directly comparable since the majority operate in different markets and apply to long-term investment types. In contrast, our approach targets those that affect short-term investment. Furthermore, our CMLP approach considers three indicators and assigns dynamic weights that vary according to the market behavior, which allows it to adapt to so-called aggressive markets. Using the other approaches in more volatile markets, the system triggers trades on necessary signals, hence its strength about the other market, which appears on the results obtained.

Fig. 15. Execution Time Comparison between Normal MLP, MM GARCH-ARIMA and our Channeled MLP.

TABLE X. COMPARISON OF PROFITS EARNED BETWEEN NORMAL MLP, MM GARCH-ARIMA AND OUR CHANNELED MLP

NN TYPE	Type	Market	Units	Profit (USD)	Half Spread Cost	Date
NORMAL MLP	Buy Market	GBP/USD	20	**0**	-1.6	15/06/2021
	Close Trade	GBP/USD	20	**-1.45**	-1.6	15/06/2021
	Buy Market	GBP/USD	20	**0**	-1.6	15/06/2021
	Close Trade	GBP/USD	20	**2.3**	-1.6	15/06/2021
	Buy Market	GBP/USD	20	**0**	-1.6	15/06/2021
	Close Trade	GBP/USD	20	**2.8**	-1.8	15/06/2021
	Buy Market	GBP/USD	20	**0**	-1.6	15/06/2021
	Close Trade	GBP/USD	20	**-2.4**	-1.6	15/06/2021
	Buy Market	GBP/USD	20	**0**	-1.6	15/06/2021
	Close Trade	GBP/USD	20	**-2.3**	-1.6	15/06/2021
	Buy Market	GBP/USD	20	**0**	-1.6	15/06/2021
	Close Trade	GBP/USD	20	**3.2**	-1.7	15/06/2021
CHANNELED MLP	Buy Market	GBP/USD	20	**0**	-1.7	15/06/2021
	Close Trade	GBP/USD	20	**2.6**	-1.7	15/06/2021
	Buy Market	GBP/USD	20	**0**	-1.7	15/06/2021
	Close Trade	GBP/USD	20	**-4.6**	-1.7	15/06/2021
	Sell Market	GBP/USD	20	**0**	-1.9	15/06/2021
	Close Trade	GBP/USD	20	**2.5**	-2	15/06/2021
	Sell Market	GBP/USD	20	**0**	-2	15/06/2021
	Close Trade	GBP/USD	20	**2.4**	-1.9	15/06/2021
	Buy Market	GBP/USD	20	**0**	-1.4	15/06/2021
	Close Trade	GBP/USD	20	**5.6**	-1.4	15/06/2021
	Buy Market	GBP/USD	20	**0**	-1.4	15/06/2021
	Close Trade	GBP/USD	20	**1.2**	-1.8	15/06/2021
MultiModal GARCH-ARIMA	Buy Market	GBP/USD	20	**0**	-1.6	15/06/2021
	Close Trade	GBP/USD	20	**-2.4**	-1.6	15/06/2021
	Buy Market	GBP/USD	20	**0**	-1.6	15/06/2021
	Close Trade	GBP/USD	20	**-4.6**	-1.6	15/06/2021
	Buy Market	GBP/USD	20	**0**	-1.6	15/06/2021
	Close Trade	GBP/USD	20	**4**	-1.5	15/06/2021
	Buy Market	GBP/USD	20	**0**	-1.5	15/06/2021
	Close Trade	GBP/USD	20	**-5**	-1.5	15/06/2021
	Sell Market	GBP/USD	20	**0**	-1.5	15/06/2021
	Close Trade	GBP/USD	20	**-3.1**	-1.6	15/06/2021
	Sell Market	GBP/USD	20	**0**	-1.6	15/06/2021
	Close Trade	GBP/USD	20	**-3.2**	-1.6	15/06/2021
BoLLINGER BAND / StoChastic RSI	Sell Market	GBP/USD	20	**0**	-1.6	15/06/2021
	Close Trade	GBP/USD	20	**10**	-1.7	15/06/2021
	Sell Market	GBP/USD	20	**0**	-1.8	15/06/2021
	Close Trade	GBP/USD	20	**1.8**	-1.6	15/06/2021
	Sell Market	GBP/USD	20	**0**	-1.6	15/06/2021
	Close Trade	GBP/USD	20	**-2**	-1.7	15/06/2021
	Sell Market	GBP/USD	20	**0**	-1.7	15/06/2021
	Close Trade	GBP/USD	20	**-2.3**	-1.8	15/06/2021
	Buy Market	GBP/USD	20	**0**	-1.8	15/06/2021
	Close Trade	GBP/USD	20	**-3**	-1.6	15/06/2021
	Buy Market	GBP/USD	20	**0**	-1.6	15/06/2021
	Close Trade	GBP/USD	20	**-3.8**	-1.8	15/06/2021

VI. Discussion

Most of the methods cited on the state of the art are methods used for algo-trading based on a single indicator and are not based on neural networks, while we target multiple and different type indicators to allow for varying market knowledge. But even we thought it useful to quote them. The only strategy that can make target method topic for our comparison is the neural multi-modal network architecture, only because this technique can take features of several types and group them by observation nature. In our study we shed more light on the approaches that use the multi modal base on 2 LSTM, the first is the LSTM-GARCH which takes into account features oriented towards volatility. And the second is the LSTM-ARIMA which takes into account the market trend, and their weights are static so do not allow to act on markets that are volatile and – or trending. Our approach takes into account another parameter which is the reversal momentum (Stochastic RSI) as oscillator. And puts more weight on the group that characterizes the market has a given period (e.g.: Gives more weight to Group V - volatility if the market is volatile or-and more weight to Group T - trend if the market has a strong trend). The back-testing of our trading strategy (Method) is based on training an NN to recognize a false signal and a real signal or a neutral signal of waiting for opportunity. The scores obtained and displayed in Fig. 11 mean that our CMLP approach is more accurate than the MM LSTM GARCH-ARIMA. The results of our experiments are more focused on trades having positive incomes which represent the main criteria of efficiency. For training / testing we used a proportion of 70% - 30% to back test our approach. We made a focus on sentiment analysis on work. In this paper we bring more light on the statistical aspect for prediction. We have already brought attention on cryptocurrencies that are influenced by communities in Twitter.

VII. Conclusion

The forex market is the more volatile market among financial markets. Trading on it at high frequency is so complex and needs a well-designed algorithm, the reason why we use machine learning, especially the Multi-layer perceptron, which is the best approach for time series. The faced challenges are training speed and execution, more than that, the accuracy of taken decision on open or not open a trade, and if the decision is opening a trade, should the MLP choose to buy, sell or wait. Channeling features helps us a lot with this. However, our approach is still technically crud and needs more ameliorations for some minor details, to be mature and well prepared for aggressive markets, to avoid bad decisions that can make us lose more trades than expected. As future works we are planned to make a focus on sentiment analysis as category of features selection in our current work. We have already brought attention on cryptocurrencies that are influenced by communities in Twitter; this area is also a subject of volatility research.

==========

Declaration of interests

☒ ***The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.***

☐ *The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:*

==========

References

[1] O. Chantarakasemchit, S. Nuchitprasitchai, and Y. Nilsiam, "Forex Rates Prediction on EUR/USD with Simple Moving Average Technique and Financial Factors," in 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Jun. 2020, pp. 771–774. doi: 10.1109/ECTI-CON49241.2020.9157907.

[2] V. Patil, N. Somani, A. Tadvi, and V. Attar, "Algorithmic Forex Trading using Combination of Numeric Time Series and News Analysis," in 2018 4th International Conference for Convergence in Technology (I2CT), Oct. 2018, pp. 1–5. doi: 10.1109/I2CT42659.2018.9058285.

[3] Ji Hoon Jang et al., "Accelerating forex trading system through transaction log compression," in 2014 International SoC Design Conference (ISOCC), Nov. 2014, pp. 74–75. doi: 10.1109/ISOCC.2014.7087602.

[4] T. C. O. Leslie, C. L. N. David, and Y. Lee, "Prediction of Forex trend movement using linear regression line, two-stage of multi-layer perceptron and dynamic time warping algorithms," 2016. doi: 10.32890/jict2016.15.2.6.

[5] A. R. P. Putra, A. E. Permanasari, and S. Fauziati, "I forex trend prediction technique using multiple indicators and multiple pairs correlations DSS: A software design," in 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), Oct. 2016, pp. 1–5. doi: 10.1109/ICITEED.2016.7863248.

[6] A. Kale, O. Khanvilkar, H. Jivani, P. Kumkar, I. Madan, and T. Sarode, "Forecasting Indian Stock Market Using Artificial Neural Networks," in 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Aug. 2018, pp. 1–5. doi: 10.1109/ICCUBEA.2018.8697724.

[7] A. Noertjahyana, A. Noertjahyana, Z. A. Abas, and Z. I. M. Yusoh, "Combination of Candlestick Pattern and Stochastic to Detect Trend Reversal in Forex Market," in 2019 4th Technology Innovation Management and Engineering Science International Conference (TIMES-iCON), Dec. 2019, pp. 1–4. doi: 10.1109/TIMES-iCON47539.2019.9024485.

[8] Y. Zhai, A. Hsu, and S. K. Halgamuge, "Combining News and Technical Indicators in Daily Stock Price Trends Prediction," in Advances in Neural Networks – ISNN 2007, Berlin, Heidelberg, 2007, pp. 1087–1096. doi: 10.1007/978-3-540-72395-0_132.

[9] Y. Alperovich, M. Alperovich, and A. Spiro, "Forecasting Price Trends in Financial Markets," in 2018 Eleventh International Conference "Management of large-scale system development" (MLSD, Oct. 2018, pp. 1–3. doi: 10.1109/MLSD.2018.8551778.

[10] Z. Li and V. Tam, "A Machine Learning View on Momentum and Reversal Trading," p. 16, 2018.

[11] T. J. Moskowitz, Y. H. Ooi, and L. H. Pedersen, "Time series momentum," Journal of Financial Economics, vol. 104, no. 2, pp. 228–250, May 2012, doi: 10.1016/j.jfineco.2011.11.003.

[12] "Moving Average Envelopes and Bollinger Bands," in Technical Analysis and Chart Interpretations, John Wiley & Sons, Ltd, 2016, pp. 215–221. doi: 10.1002/9781119204800.ch17.

[13] M. Ghorbani and E. K. P. Chong, "Stock price prediction using principal components," PLOS ONE, vol. 15, no. 3, p. e0230124, Mar. 2020, doi: 10.1371/journal.pone.0230124.

[14] H. He, J. Chen, H. Jin, and S.-H. Chen, "Trading Strategies Based on K-means Clustering and Regression Models," in Computational Intelligence in Economics and Finance: Volume II, S.-H. Chen, P. P. Wang, and T.-W. Kuo, Eds. Berlin, Heidelberg: Springer, 2007, pp. 123–134. doi: 10.1007/978-3-540-72821-4_7.

[15] S. Hansun and M. B. Kristanda, "Performance analysis of conventional moving average methods in forex forecasting," in 2017 International Conference on Smart Cities, Automation Intelligent Computing Systems (ICON-SONICS), Nov. 2017, pp. 11–17. doi: 10.1109/ICON-SONICS.2017.8267814.

[16] C. Xiao, W. Xia, and J. Jiang, "Stock price forecast based on combined model of ARI-MA-LS-SVM," Neural Comput & Applic, vol. 32, no. 10, pp. 5379–5388, May 2020, doi: 10.1007/s00521-019-04698-5.

[17] S. Hansun, "FX forecasting using B-WEMA: Variant of Brown's Double Exponential Smoothing," in 2016 International Conference on Informatics and Computing (ICIC), Oct. 2016, pp. 262–266. doi: 10.1109/IAC.2016.7905726.

[18] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," Physica D: Nonlinear Phenomena, vol. 404, p. 132306, Mar. 2020, doi: 10.1016/j.physd.2019.132306.

[19] S. Lauguico et al., "A Fuzzy Logic-Based Stock Market Trading Algorithm Using Bollinger Bands," in 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Nov. 2019, pp. 1–6. doi: 10.1109/HNICEM48295.2019.9072734.

[20] M. Butler and D. Kazakov, "Particle Swarm Optimization of Bollinger Bands," in Swarm Intelligence, Berlin, Heidelberg, 2010, pp. 504–511. doi: 10.1007/978-3-642-15461-4_50.

[21] T. W. A. Khairi, R. M. Zaki, and W. A. Mahmood, "Stock Price Prediction using Technical, Fundamental and News based Approach," in 2019 2nd Scientific Conference of Computer Sciences (SCCS), Mar. 2019, pp. 177–181. doi: 10.1109/SCCS.2019.8852599.

[22] T. Chong and W.-K. Ng, "Technical analysis and the London stock exchange: Testing the MACD and RSI rules using the FT30," Applied Economics Letters, vol. 15, pp. 1111–1114, Feb. 2008, doi: 10.1080/13504850600993598.

[23] A. Rodríguez-González et al., "Improving Trading Systems Using the RSI Financial Indicator and Neural Networks," in Knowledge Management and Acquisition for Smart Systems and Services, Berlin, Heidelberg, 2010, pp. 27–37. doi: 10.1007/978-3-642-15037-1_3.

[24] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," Insights Imaging, vol. 9, no. 4, Art. no. 4, Aug. 2018, doi: 10.1007/s13244-018-0639-9.

[25] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," Neural Computation, vol. 29, no. 9, pp. 2352–2449, Jun. 2017, doi: 10.1162/neco_a_00990.

[26] W. Wang, Y. Yang, X. Wang, W. Wang, and J. Li, "Development of convolutional neural network and its application in image classification: a survey," OE, vol. 58, no. 4, p. 040901, Apr. 2019, doi: 10.1117/1.OE.58.4.040901.

[27] T. Das and G. Saha, "Addressing big data issues using RNN based techniques," Journal of Information and Optimization Sciences, vol. 40, no. 8, pp. 1773–1785, Nov. 2019, doi: 10.1080/02522667.2019.1703268.

[28] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to Construct Deep Recurrent Neural Networks," arXiv:1312.6026 [cs, stat], Apr. 2014, Accessed: Nov. 23, 2020. [Online]. Available: http://arxiv.org/abs/1312.6026.

[29] R. Bai, J. Zhao, D. Li, X. Lv, Q. Wang, and B. Zhu, "RNN-based demand awareness in smart library using CRFID," China Communications, vol. 17, no. 5, pp. 284–294, May 2020, doi: 10.23919/JCC.2020.05.021.

[30] H. Ramchoun, M. A. J. Idrissi, Y. Ghanou, and M. Ettaouil, "Multilayer Perceptron: Architecture Optimization and training with mixed activation functions," in Proceedings of the 2nd international Conference on Big Data, Cloud and Applications, New York, NY, USA, Mar. 2017, pp. 1–6. doi: 10.1145/3090354.3090427.

[31] M. Aitkin and R. Foxall, "Statistical modelling of artificial neural networks using the multi-layer perceptron," Statistics and Computing, vol. 13, no. 3, pp. 227–239, Aug. 2003, doi: 10.1023/A:1024218716736.

[32] G. I. Sher, "Evolving Chart Pattern Sensitive Neural Network Based Forex Trading Agents," arXiv:1111.5892 [cs], Jan. 2012, Accessed: Nov. 23, 2020. [Online]. Available: http://arxiv.org/abs/1111.5892

[33] O. Fathi, "Time series forecasting using a hybrid ARIMA and LSTM model," p. 7.

[34] S. Galeshchuk, "Neural networks performance in exchange rate prediction," Neurocomputing, vol. 172, pp. 446–452, Jan. 2016, doi: 10.1016/j.neucom.2015.03.100.

[35] I. Zabbah and E. Partovi, "Enhancing an Automated Trading Strategy Using Artificial Neural Networks," p. 4, 2012.

[36] N. D'Lima and S. Khan, "FOREX rate prediction using a Hybrid System," International Journal of Engineering Research and Technology, vol. 3, Nov. 2014.

[37] S. Borovkova and I. Tsiamas, "An ensemble of LSTM neural networks for high-frequency stock market classification," J FORECASTING, vol. 38, no. 6, pp. 600–619, 2019, doi: 10.1002/for.2585.

[38] S. T. A. Niaki and S. Hoseinzade, "Forecasting S&P 500 index using artificial neural networks and design of experiments," J Ind Eng Int, vol. 9, no. 1, p. 1, Feb. 2013, doi: 10.1186/2251-712X-9-1.

[39] B. X. Yong, M. R. Abdul Rahim, and A. S. Abdullah, "A Stock Market Trading System Using Deep Neural Network," in Modeling, Design and Simulation of Systems, Singapore, 2017, pp. 356–364. doi: 10.1007/978-981-10-6463-0_31.

[40] S. Liu, C. Zhang, and J. Ma, "CNN-LSTM Neural Network Model for Quantitative Strategy Analysis in Stock Markets," in Neural Information Processing, Cham, 2017, pp. 198–206. doi: 10.1007/978-3-319-70096-0_21.

[41] W. Liu and M. So, "A GARCH Model with Artificial Neural Networks," Information, vol. 11, p. 489, Oct. 2020, doi: 10.3390/info11100489.

[42] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," Neurocomputing, vol. 50, pp. 159–175, Jan. 2003, doi: 10.1016/S0925-2312(01)00702-0.

[43] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal Machine Learning: A Survey and Taxonomy," arXiv:1705.09406 [cs], Aug. 2017, Accessed: Jul. 02, 2021. [Online]. Available: http://arxiv.org/abs/1705.09406.

[44] H. A. Pathberiya, C. D. Tilakaratne, and L. L. Hansen, "An intelligent system for forex trading: Hybrid ANN with GARCH and intrinsic mode functions," in 2017 Intelligent Systems Conference (IntelliSys), Sep. 2017, pp. 436–445. doi: 10.1109/IntelliSys.2017.8324331.

[45] R. M. Fujimoto and H. Feng, "A shared memory algorithm and proof for the generalized alternative construct in CSP," Int J Parallel Prog, vol. 16, no. 3, pp. 215–241, Jun. 1987, doi: 10.1007/BF01407934.

[46] C. A. R. Hoare, "Communicating sequential processes," Commun. ACM, vol. 21, no. 8, pp. 666–677, Aug. 1978, doi: 10.1145/359576.359585.

[47] N. Fikri, M. Rida, N. Abghour, K. Moussaid, and A. El Omri, "A Near Metal Platform for Intensive Big Data Processing Using A Novel Approach: Persistent Distributed Channels," in Proceedings of the 2020 5th International Conference on Big Data and Computing, New York, NY, USA, May 2020, pp. 1–5. doi: 10.1145/3404687.3404692.

[48] A. Prell and T. Rauber, "Go's Concurrency Constructs on the SCC," p. 7, 2012.