Journal of Contemporary Issues in Business and Government Vol. 27, No. 3, 2021
https://cibg.org.au/

P-ISSN: 2204-1990; E-ISSN: 1323-6903
DOI: 10.47750/cibg.2021.27.03.263

Realtime stock prediction using Transformations and Modeling

DR Y.V.S.SAI PRAGATHI[1], M.V.S PHANI NARASIMHAM[2]

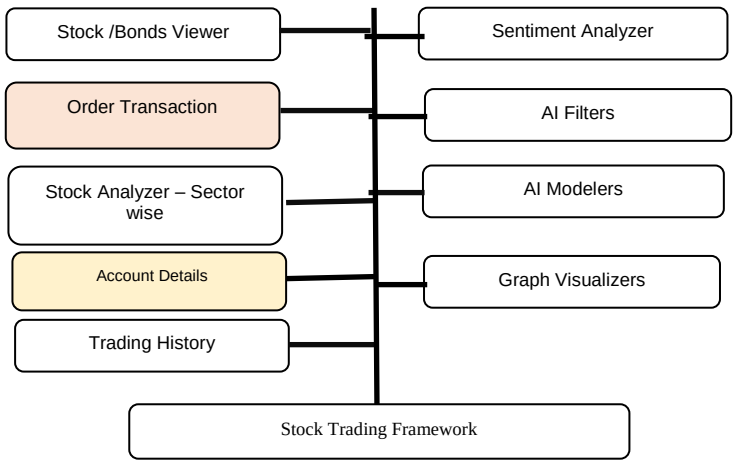
[1]Professor, Stanley college of engineering and technology for women, Hyderabad, Telangana,
[2]Senior Architect, Wipro Technologies, Alumni IISC Bangalore.

Abstract: Real time stock prediction is growing demand with dynamic changing markets. Especially sentiments of the market are going to change the stock price dynamically, giving fund managers challenging scenario when they are investing. This paper proposes stock prediction model aiding stock agents to predict closing prices and current price of the stock. The algorithm models the stock tick and balance data parameters using KPCA Trans-formation and Artificial neural network model. Our stock predictor is cost optimized and end to end delay optimized to achieve real time stock prediction. Stock predictor is architected using Clean architecture and GRPC based microservices which gives the advantage of real time modifications, pluggability with minimum validation costs. Secure and cost-effective data backup features of the cloud will give additional cost advantage to the companies implementing the new stock predictor services.

Keywords: Stock, KPCA, ANFIS, Microsoft Azure, KPCA, RSI, EMA10, EMA20, ADO,Kubernetes, GRPC, Miro service, Clean Architecture, dockers, Load Balancer, Cloud Simulators

INTRODUCTION:

Typical stock application framework, consisting of key modules is shown in Fig.1. It is used by bank securities, stock agents for stock booking and reducing risk of daily trading. Stock viewer module consists of reading real time stock tick data parameters like daily-high, daily-low, daily-range, ask, bid, last-trade-amount, ebitda, market-capitalization, pe-ratio, volume, etc. and display the data in user customized grid. Order Transaction module consists of account details (id, balance, daily-profit-loss, gross-profit-Loss), current price per share, transaction type for taking and placing the order. Stock analyzer module consists of different types of charts of stock price, sales volume and financial data of the stock market-cap, pe, roe, div-yield, price-to-book, price-to-free-cashflow, net-profit-margin, etc. to display the data in user customized charts. Stock analyzer module also consists of heat maps of different industry sectors – technology, financial, healthcare, consumer goods, telecom services. Sentiment Analyzer integrates parses the stock news and integrates the effect of Facebook and twitter into stock predictions., AI Filters and AI Modelers will filter the data and will have real time AI models using logistic regression, boosted Bayesian trees, support vector machine (SVM), artificial neural network (ANN), anfis and deep learning techniques. There is very high demand for real time stock predictors especially when investing money by fund managers. Real time stock predictor service solves the problems of predicting stock price accurately masking neighborhood stock fluctuations, news fluxes and market movements. The rest of the paper is organized as follows: Related work is discussed in section 2, cloud reference architecture using GRPC micro services is illustrated in section 3, ANN cloud mathematical model is summarized in section 4, Modeling parameters are discussed in section 5, results are discussed in section 6.



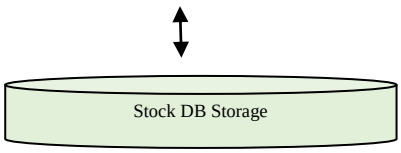


Fig.1 Stock Predictor Components

LITERATURE SURVEY

M.Th Kotouzal in their paper "A dockerized framework for hierarchical frequency-based document clustering on cloud computing infrastructures" presented dockized hierarchical clustering framework using the usage of topics in the document and multi metrics [1].

Aparna N. et. al in their paper, "Prediction Models for Indian Stock Market", have com- bined prediction models with sentiment analysis. Sentiment of the company using twitter and news combined with open price, close price and volume traded per day are used in the predicted model. Decision boosted Tree was found to give better performance than and Logistic Regression [2].

Kunal P. et. al in their paper, "Stock Market Analysis using Supervised Machine Learning" have used SVM to predict the future stock price [3].

Xi Zhang et.al in their paper, "Stock Market Prediction via Multi-Source Multiple Instance Learning", have used stock data, social media data, web news data and developed multi source multiple instance model (M-MI). SVM is applied on the M-MI model to predict the stock price [4].

K. Hiba Sadia et.al in their paper, "Stock Market Prediction using Machine Learning Algorithms", have preprocessed the stock dataset and applied random forest and SVM for predicting of stock price [5].

Osman H et.al in their paper "A machine learning model for stock market prediction", integrates particle swarm optimization (PSO) algorithm and Least Square(LS)-SVM for stock price prediction using financial technical indicators relative strength index, money flow index, exponential moving average, stochastic oscillator and moving average convergence/divergence [6].

Ekram G et.al in their paper, "Using artificial neural network models in stock market index prediction" have analyzed different ANN models like multi-layer perceptron(MLP), dynamic artificial neural network (DAN2) and hybrid neural networks which use generalized autoregressive conditional heterodasticity (GARCH). They found MLP model outperforms over DAN2 and GARCH in terms of stock market index prediction [7].

Amin H et.al in their paper, "Stock market index prediction using artificial neural network" have applied OSS training method and TANGSIG transfer function in a network with 20-40-20 neurons in hidden layers resulted in an R2 value of 0.9408 for validation 'NASDAQ dataset [8]

Weiwei Jiang in their paper, "Application of deep learning in stock market prediction: recent progress" have analyzed different linear and machine learning tools that can be used for stock prediction. Different deep learning models and their reproducibility to baseline stock prediction is accomplished in this paper [9].

Joanathan L. T in his paper, "A Bayesian regularized artificial neural network for stock market forecasting" have used Bayesian regularized artificial neural network for predicting the future closing price of individual stocks. The probabilistic nature of the network weights allows the investors to safely expand the network without increasing the risk of overfitting with the use of effective number of parameters [10].

Dharmaraja S et. al in their paper, "Indian stock market prediction using artificial neural networks on tick data" have compared different neural networks learning algorithms Levenberg-Marquardt, Scaled Conjugate Gradient and Bayesian Regularization for stock market prediction based on tick data and found Bayesian regularization gives best results for stock prediction [11].

Nirbhay N in his paper, "Stock market prediction using Machine Learning and Cloud Computing", have proposed using cloud computing and machine learning algorithms to perform stock prediction. In this paper SVM, Linear Regression, random Forest, XG Boost, LSTM for deep learning are used to perform stock prediction [12].

Jyh-Shing, Roger Jang in their paper "ANFIS: Adaptive-Network-Based Fuzzy Inference System" presented the architecture and learning procedure underlying ANFIS, a fuzzy inference system implemented in the framework of adaptive networks [13]. Briefly, ANNs are designed based on simulation of the human brain with the purpose of determining the relationship between outputs and inputs of a system. An ANN is trained with the available experimental data throughout the training step and is employed for estimating the unknown data. Neural networks include, simple synchronous processing components that are known as nodes or neurons located throughout layers. Usually, an artificial neural network has three layers: an output layer, a hidden layer, and an input layer. ANFIS modeling is used to solve the non-linear problems. An adaptive neuro-fuzzy inference system (ANFIS) is a kind of artificial neural network that is based on Takagi–Sugeno fuzzy inference system. It integrates both neural networks and fuzzy logic principles and captures the benefits of both in a single framework [13].

Ghofrane Rehaiem, Hamza Gharsellaoui, Samir Ben Ahmed in their paper "A Neural Networks Based

Approach for the Real-Time Scheduling of Reconfigurable Embedded Systems with Minimization of Power Consumption" have used ANN based back propagation technique to model the real time scheduling of tasks in embedded systems to achieve minimum cost [14]. ANN with proper number of input, output and hidden neurons is able to identify and classify complex regions.
S.Agatanovic-kustrin, R. Beresford, "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research" has applied ANN to find optimal dosage of the drug and analyzing chromatographic data [15]
M.V.S Phani Narasimham et. al in their paper, "Development of realistic models of oil well by modeling porosity using modified ANFIS technique" have used ANN based neutron porosity model will give reliable static reservoir models for oil well simulation frame- 3 works[16].
Alberto Nunez et.al in their paper "iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator" propose cloud simulator to model cloud based large simulations by varying number and type of virtual machines [17].

GRPC Based Simulator Pods:
Existing stock predictor application is modified using Clean Architecture [18] and GRPC based microservices which will be containerized as docker containers. The GRPC PODs will communicate using mes- saging services bus queue to communicate between the PODs. The docker containers are deployed to pods which are managed by the Kubernetes cluster. The optimal number of containers and pods to achieve real time response will be determined based on optimized energy and end to end delay schemes. This approach has added advantage as it replaces the heavy weight VMs with docker containers giving more granularity to cloud scheduling to achieve the real time elastic requirement of modern cloud. Stock Viewer and Chart Visualizer pods are implemented using Angular JS and communicate with GRPC APIs. AI Filters and AI models are implemented using C# and Python flask frameworks. Intermediate stock data model required for AI analysis is saved in mongo db for quick and easy access.

ANN BASED REAL TIME CLOUD ARCHITECTURE:
ANN Supervised learning model with various algorithms back propagation, hybrid algorithms are explored to develop optimal ANN based cloud reference model. ANN perceptron network with input layer, hidden layer and output layer classifies input stock predictor tasks. Input batch of simulator tasks of a real time scenario are classified based on the task features obtained from the training batches. Stock predictor characteristics in json file are fed to the ANN modeler. Fig 2. Gives the clean architecture using PODS.

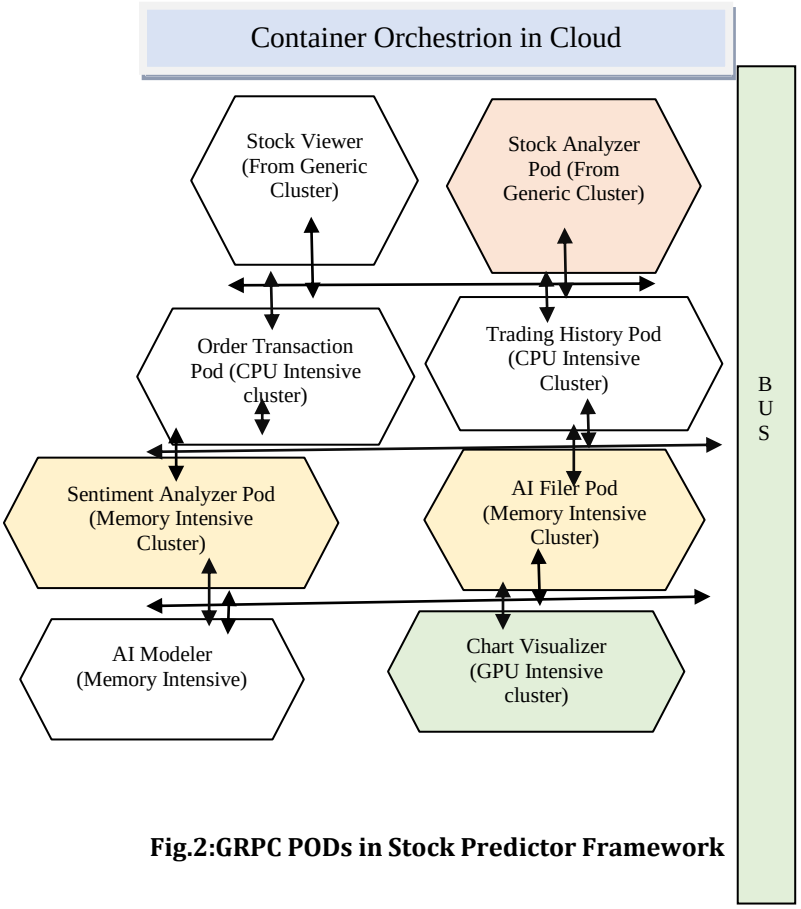


Fig.2:GRPC PODs in Stock Predictor Framework

Training batch measures the following VM container characteristics CPU Speed, GPU, Memory Intensive and Generic. The ANN classifier will identify the most suitable containers for execution Class 1 containers are CPU intensive. Class 2 containers are GPU intensive. Class 3 containers are memory intensive. Class 4 Generic containers. Ant colony optimizer will find the optimal schedule for execution of the tasks on the containers that will have the minimum timespan and cost for execution. Reducing the cost reduces the power usage and this will be determined by the energy consumptions of the Green cloud simulator.

ANN Cloud Model

ANN Back Propagation Mathematical Model: Input task characteristic are input to 4 input perceptron layer followed by two output neurons. The perceptron network weights are tuned using back propagation on the cost from the training set. $t = \{t_1, t_2, t_3, t_4\}$; is the expected target value set.

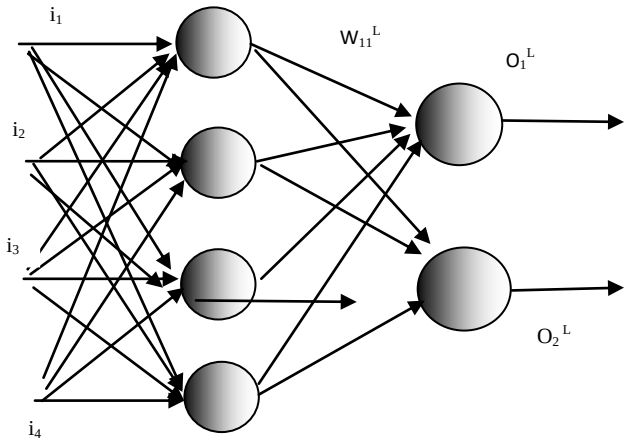


Fig.3 4-input and 4-hidden neurons perceptron network

Eq.1 gives the cost of the network model using mean square error of output and expected target.

$$C = \Sigma_{j=0}^{K=0,n} (o_{kj} - t_{kj})^2 \quad (1)$$

$$z_j^L = w_{j0}^L o_1^L + w_{j1}^L o_2^L + w_{j2}^L o_3^L + w_{j3}^L o_3^L + b^L \quad (2)$$

$$o_j^L = f(z_j^L) \quad (3)$$

Eq.2 calculates the input to the output neuron of the final layer. Eq.3is the activation function output from input z.

$$\frac{\partial c_0}{\partial w_{jk}^L} = \frac{\partial z_j^L}{\partial w_{ik}^L} \times \frac{\partial o_j^L}{\partial z_j^L} \times \frac{\partial c_0}{\partial o_j^L} \quad \text{(Eq. 4)}$$

Eq.4 gives the backpropagation effect of the weights coefficient from the cost which will be tuned using gradient descent of the training set. ACO Mathematical Model In ACO the pheromone component is computed using $(1/ (ETC_{ij} + Cost_j)$, Where ETC_{ij} is the expected time to compute task on the Pod and cost is cost per hour charged for the VMs. The pheromone of ant k mapping task i to container j is given by

$$\tau_{i,j}^k = (1- p) \tau_{i,j} + \Sigma_{k=1}^m \Delta \tau_{i,j}^k \quad (5)$$

The probability of mapping task i to container j is given by

$$\Sigma((\tau_{i,j})^{\alpha} (_{I,j})^{\beta}) \quad (6)$$

β represent the influence factor of heuristic value. Cumulative probability of a task i mapping to the container is calculated. $\{ P_1, P_{1+2}, P_{1+2+3,\ldots} \}$. Random number between 0 to 1 is selected to map the task to container.

Modeling Parameters:

Stock tick data parameters open, high, low, close and volume are modeled using ANN. Relative strength index

(RSI), Estimated moving average EMA10 EMA 20, Accumulation Distribution indicator (ADO) are fed into the ANN model to determine the stock closing price and stock current price in real time. Real time optimized ANN model is developed for each stock in real time to determine the accurate price. Estimated moving average helps the model to detect the strong and sustained uptrend of the markets. Relative strength index (RSI) helps the model to capture the pattern of over-bought when it is over 70 and oversold when it is below 30. Accumulation Distribution indicator (ADO) helps the model to capture divergence between stock price and volume flow. If price is rising but ADO is falling indicates that price decline is going to occur. Price rate of change (PROC) indicator helps capture the momentum of the stock, the indicator moves into a positive territory if price changes are to the upside. Fig 4. Gives the model using 4 input ANN model.

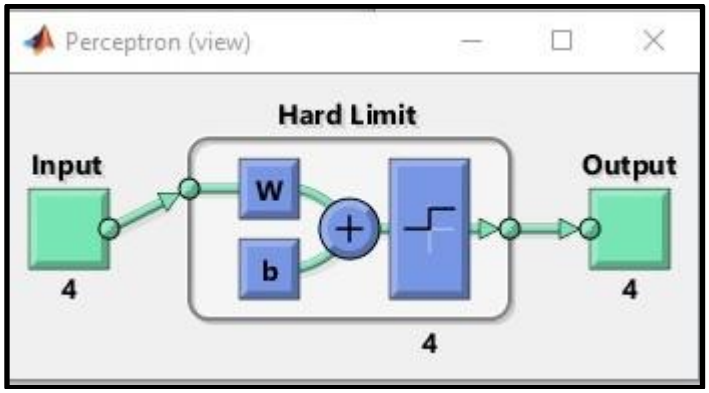


Fig.4: 4-input and 4-hidden neurons perceptron network

RESULTS

Training batch of resource tasks is used to measure the make span of the tasks on CPU intensive, GPU Intensive,

Memory Intensive and Generic container VMs. Relative timespan characteristic of each task is measured using the below formulas.

TCPU = Time Span of the task on CPU intensive VM.

TGPU = Time Span of the task on GPU intensive VM.

TGU = Time Span of the task on Generic VM.

TMI = Time space of the task on Memory Intensive VM.

Relative TimeSpan of CPU

RTCPU = TCPU / TCPU+TGPU+TGU+ TMI

Relative timespan parameters of the tasks are used as inputs to the ANN models of Perceptron networks trained using back propagation. Input parameters of the task characteristic needed for the real time stock predictor service are given as input json file. Input stock predictor training data is used to model the ANN classifier at desired rate of 1e-4 and learning rate of 0.1.

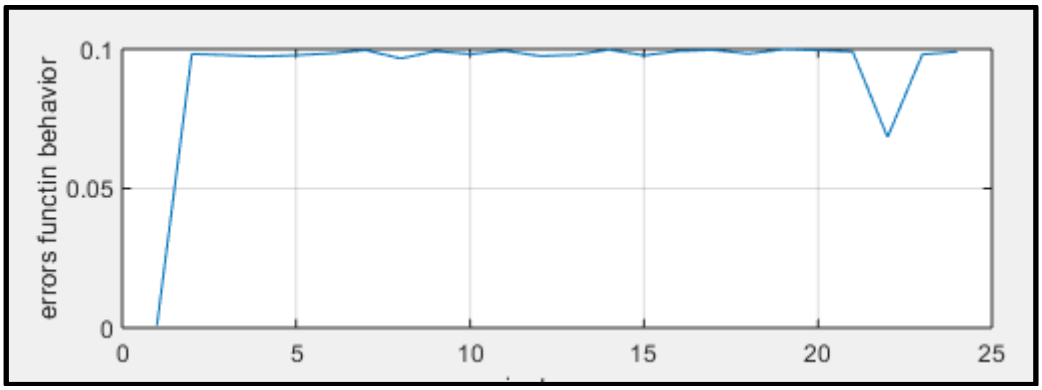


Fig. 5: MSE error of output in the final iteration of ANN.

Docker container images of stock predictor architecture described in section 3 are saved in cloud private registry. Container VMs are deployed on VMs orchestered using kubernetes cluster service. Stock predictor tasks are scheduled on the container pods using the ANN+ACO schedulers.

Table 1: Results of the ANN model used for the prediction of closing of the price of stock in real time for different ANN models. Real time stocks for 100 days are used in the analysis of two different company stocks.

Model	Average absolute error for stock1	Average absolute error for stock1	Average absolute error for stock1
ANN	5.33	11.05	11.02
Transformed ANN	5.66	5.02	8.38

Average Error computation of stock with varying ANN Model and Transformed ANN model

Table 2: Results of the ANN model used for the prediction of closing of the price of stock1 in real time for different ANN hidden layers.

ANN Model Layers	4	6	8	10	12
Average Error	5.3	8.9	8.2	1.4	9.0

CONCLUSION:

Real time ANN model is generated to get optimal stock prediction. Integrating sentiment analysis model, balance sheet parameters into the ANN model will give more accurate results for the fund managers and security investors reducing the risk of bad investments. Multi-dimensional transformation can be further explored for predicting the stock variations more accurately. ANFIS with customized rules and reactive learning models will be explored further to improve the stock prediction. From the mathematical model and results it is concluded that real time reference cloud architecture with the ANN model will give time optimized and cost-effective solution for stock prediction for stock investors. It is observed that sudden transitions are not modeled easily with the KPCA transformed ANN Model. In the next research paper Deep-Q learning with immediate rewards will be explored to model the dynamic stock markets.

ACKNOWLEDGMENT

Thankful to Stanley college of Engineering & Technology for women, Principal Dr. Satya Prasad Lanka for encouraging to pursue new research areas.

Thankful to Wipro management, Ramakrishna Kashikar, Adarsh Devendraswamy, Praveen Woopalanchi, Wipro Technologies in encouraging and helping with relevant environment for working on this paper.

REFERENCE:

1. M. Th. Kotouza1, F. E. Psomopoulos and P. A. Mitkas1, “A dockerized framework for hierarchical frequency-based document clustering on cloud computing infrastructures”, Journal of cloud computing, 2020, 9:2.
2. N Aparna; M M; Manohara Pai; Radhika M Pai. Prediction Models of Indian Stock Market”, 12th international multi conference on information processing - 2016. *Procedia Computer Science*. **2016**, 89, 441–449.
3. Kunal P, Neha A, “Stock Market Analysis using Supervise Machine Learning”, 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing, India 14th-16th Feb 2019.
4. Xi Zhang, Si Qu, Jieyun H, Binxing F and Philip Y, “Stock Market Prediction via Multi-Source Multiple Instance Learning”, IEEE Access, Aug 27, Oct 8 2018, DOI 10.1109/ACCESS.2018.2869735.
5. K.Hiba Sadia, Adita S, Adaarrsh P, Sarmistha P, Saurav S, “Stock Market Prediction Using Machine Learning Algorithms”, IJEAT, ISSN: 2249-8958, Vol-8, Issue -4, April 2019.
6. Osman H, Omar S. S, and Mustafa A. S, “A Machine learning model for stock market prediction”, International Journal of Computer Science and Telecommunications, Vol-4, Issue -12, Dec 2013, pp 17-23.
7. Erkam G, Gulgan K and Tugrul U.D, “Using artificial neural netowrk models in stock market index prediction”, Expert systems with Applications, DOI 10.1016/j.eswa.2011.02.068.
8. Amin H. M, Moein H M, Morteza E, “Stock market index prediction using artificial neural network”, Journal of Economica, Fiance and Administrative Science, doi 10.1016/j.jefas.2016.07.002.
9. Weiwei Jiang, “Application of deep learning in stock market prediction: recent progress”, Elsevier Journal, Mar 5 2020.
10. Jonathan L. T, “A Bayesian regularized artificial neural network for stock market forecasting”, Elsevier,

Expert systems with Applications, doi 10.1016/j.eswa.2013.04.013
11. Dharmaraj S, Veneet K and Abhishek M, " Indian stock market prediction using artificial neural networks on tick data" Financial Evolution, doi 10.1186/s40854-019-0131-7.
12. Nirbhay N, "Stock Market Prediction using Machine Learning and Cloud Computing", International Journal of Engineering And Computer Science, Vol-8 Issue 9 Sept 2019, pp No. 24847-24850.
13. Jyh-Shing Roger Jang, "ANFIS:Adaptive-Network-Based Fuzzy Inference System", IEEE Transactions on Systems Man and Cybernetics, June 1993.
14. Ghofrane Rehaiem, Hamza Gharsellaoui, Samir Ben Ahmed, "A Neural Networks Based Approach for the Real-Time Scheduling of Reconfigurable Embedded Systems with Minimization of Power Consumption.", ICIS Conference, At Okayama, Japan, June 2016, 10.1109/ICIS.2016.7550777.
15. S. Agantonovic-kustrin, R. Beresford, "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research", Journal of Pharmaceutical and Biomedical Analysis", 22(2000) 717-727.
16. M.V.S Phani Narasimham, Dr. Y.V.S Sai Pragathi, "Development of realistic models of oil well by modeling porosity using modified ANFIS technique", International Journal on Computer Science and Engineering, Vol.11, No.07, July 2019.
17. Alberto Nunez, Jose L.Vázquez-Poletti, Agustin C. Caminero, Gabriel G. Castañé, Jesus Carretero, Ignacio M. Llorente, "iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator", J Grid Computing (2012) 10:185–209.
18. Roland H. Steinegger, P. Giessler, B. Hippchen and Sebastian Abeck, "Overview of a Domain-Driven Design Approach to Build Microservice-Based Applications ", SOFTENG 2017: The third International conference on Advances and Trends in Software , ISBN: 978-1-61208-553-1.