

Basrah Journal of Science

Volume 31 , No. 2 , 2013

Editor – In Chief

Prof. Dr. Kadhim Saleh Hassan

Editorial Secretary

Prof. Dr. Karima Majed Zaidan

Publish By

**College of Science, University of Basrah
Ministry of Higher Education and Scientific Research**

Basrah Journal of Science

Editor – In Chief
Prof. Dr. Kadhim Saleh Hassan

Editorial Secretary
Prof. Dr. Kareema Majed Zaidan

Editorial Committee
1 – Prof. Dr. Badir N. Albadran
2 – Prof. Dr. Hadi S. Al-Lami
3 – Prof. Dr. Nasir Abd-Ali Halify
4 – Prof.Ass.Dr.Saeed Abdul – ameer Taban
5– Prof.Ass.Dr. Abdul hussein Mohsin Abdullah

Publish By
College of Science, University of Basrah
Ministry of Higher Education and Scientific Research

Subject Index	page
Measuring full and partial geometrical similarity of Map objects <i>Mahdi JASIM and Tawfiq Al ASADI</i>	1 - 5
Estimating MPEG-4 Video Transmission over Gilbert Wireless Channels <i>Ghaida. A. AL-Suhail and Sarah A. Subber</i>	6 - 19
Computer Network Routing Using Fuzzy Neural Networks <i>Khlood Ahmad Nasar ; Turki Y. Abdalla Abdulkareem Y. Abdalla</i>	20 - 35
M OUT OF N ERROR CORRECTING CODE USING DIGITAL FILTERS <i>Rawnaq A. Habeeb</i>	36 - 43
Currency exchange rate prediction using wavelet network <i>Adalla Mehdi</i>	44 - 53
Satellite image enhancement using histogram hyperbolization <i>Hussain Zaydan Ali</i>	54 - 63
Financial Prediction using Inductive Models <i>Turki Y. Abdalla; Abdulkareem. Y. Abdalla; Lamya A. Dexon</i>	64- 72
New Three Methods for Improving Initialization of <i>k</i>-Means Clustering <i>Abbas H. Hassin Alasadi, Moslem Mohsinn Khudhair</i>	73 - 85
A Field Bus Network With CAN Protocol And a Fuzzy Neural Petri Net Controller <i>Ali A. Abed, AbdulAdhem A. Ali, Ali F. Marhoon, Nauman Aslam</i>	86-102

Subject Index	page
The Predator-Prey Model Simulation <i>Taleb A. S. Obaid</i>	103-109
Classification of GIS Image using GLCM and Neural Network <i>Tawfiq A. Alasadi ; Wadhah R. Baiee</i>	110- 119
Enhancing Intrusion Detection System Using Distributed Environment Safaa O. Almamory, Wesam S. Bhaya, Anees M. Hadi	120- 131
توليد شفرة السمات الدلالية للأسماء آليا في المعجم المحوسب 132- 143 سلمى محمود عبد النبي , يسرى مالك ضمد	132- 143

auther Index	page
<i>Abbas H. Hassin Alasadi , Moslem Mohsinn Khudhair</i>	73 - 85
<i>Adalla Mehdi</i>	44 - 53
<i>Ali A. Abed, AbdulAdhem A. Ali, Ali F. Marhoon, Nauman Aslam</i>	86-102
<i>Ghaida. A. AL-Suhail and Sarah A. Subber</i>	6 - 19
<i>Hussain Zaydan Ali</i>	54 - 63
<i>Khlood Ahmad Nasar ; Turki Y. Abdalla Abdulkareem Y. Abdalla</i>	20 - 35
<i>Mahdi JASIM and Tawfiq Al ASADI</i>	1 - 5
<i>Rawnaq A. Habeeb</i>	36 - 43
<i>Safaa O. Almamory, Wesam S. Bhaya,Anees M. Hadi</i>	120- 131
<i>Taleb A. S. Obaid</i>	103-109
<i>Tawfiq A. Alasadi ;Wadhah R. Baiee</i>	110- 119
<i>Turki Y. Abdalla; Abdulkareem. Y. Abdalla; Lamya A. Dexon</i>	64- 72

Measuring full and partial geometrical similarity of Map objects

Mahdi JASIM

*College of Computer Technologies
Univ. of Babylon-Iraq
mnjisd@yahoo.com*

cell. 009647813184069

Tawfiq Al ASADI

*College of Computer technologies
Univ. of Babylon-Iraq
tawfiqasadi63@yahoo.com*

cell. 009647801463183

Abstract:

Geographic Information System(GIS) are usually classified into raster, vector, and raster – vector systems. The research deals with proposing new algorithm called **GIS-Span** to measure the partial and full geometrical similarity between map objects. The algorithm is used for clustering the vector features of GIS. The vector data are usually stored in data files called shape files. These files contains the (point, lines, polygons,...,etc). The extracted data is then stored in a dataset to be processed by the proposed algorithm to discover the full and partial similarities among map objects to assist the clustering and analysis of map data. It deals with clustering the polylines and polygonal data according to their geometrical similarity.

Keywords :

Graph clustering algorithms, Graph mining, GIS data analysis, Mining GIS data ,Spatial data mining, geometrical similarity.

Introduction

During the study of related researches to Spatial clustering which deals with spatial data that is generally organized in the form of a set of *points* or *polygons*[Deep et.al 04] we have found that mining of GIS spatial data is fertile for more research to be more convenient to GIS applications especially the vector data that

deals with GIS Graph data which deals with vector map elements like points, lines, polygons,..., etc , and the huge amount of GIS-based applications make it essential to improve graph mining techniques to extract the spatial knowledge embedded in that data to support GIS applications to be installed in new data analysis fields to support geographers and GIS

users to investigate the similarity among the spaghetti of map objects, and therefore can cluster them

The proposed algorithm is intended to extract spatial knowledge based on the similarity among map objects to cluster them accordingly. The research is consisting from the following main steps. Extracting the graph features included in GIS-Shape files, design and implement dataset to store the extracted data, propose a new algorithm called **GIS-Span** to discover the fully and partially similar vector map objects, cluster the resultants similar graph features, re-visualize the discovered object on the real map giving them a distancing effects.

1. Related works

Recently the GIS graph mining techniques attracts more researching efforts in both raster and vector data mining, but we are interested in the mining of graph data which concerned with knowledge discovery in GIS-vector data, the extracted data are intended to enhance the GIS-query system to be more responding to user queries.

In 2009 Paul Van Dooren Catherine Fraikin [C. FRA 02] studied the Similarity matrices for coloured graphs but only between nodes of two graphs, to the case of coloured graphs, where the colouring is either on the nodes or on the edges of both graphs. The proposed method tries to find the optimal matching between the nodes or edges of both graphs but only performs the comparison when their colours are the same.

In 2010 Akshara Pande et. al.

designed an information system to work with data referenced by spatial geographical coordinates. They are detecting design patterns so that it can be used as a conceptual tool to cope with recurrent problems appearing in the GIS domain [aksh 01].

In 2011[Deep 04] used districting which is the process of dividing a geographic space or region of spatial units often represented as polygons into smaller sub-regions or districts. As such, it can be viewed as a set partitioning problem, i.e. the problem is to cluster the entire set of spatial polygons into groups according to the spatial properties involved, redistricting is like to spatial clustering which can be done using spatially-flavoured constraints such as spatial contiguity and compactness which are spatial features in GIS data.

In 2010 a study of clustering algorithms of area geographical entities based on geometric shape similarity was made and the researchers presented a similarity criterion of line segments shape and a criterion of area geographical entities comprehensively utilizing distance and geometric shape similarity. Clustering algorithms based on these criterions are more suitable for clustering analysis of area geographical entities[Chen et. al 03]. The extracted knowledge can be used to explore deeper level knowledge combined with other data mining methods and to improve the efficiency and quality of data mining [Ilar et.al 06]. We studied clustering algorithms of area geographical entities based on geometry

similarity. They also presented a similarity criterion of line segments shape and a criterion of area entities similarity comprehensive utilizing distance and geometry similarity. Clustering algorithms based on these criteria are more suitable for clustering analysis of area geographical entities [Chen et. al 03] .

2. Graph Mining Algorithms

Graph mining algorithms deal with discovering the knowledge embedded GIS vector data which are mostly stored in GIS shape files. These GIS data structures are stored as one geographic feature per shape file like point feature layer, multipoint feature layer, and polygon feature layer [Kang 07] [saya 08].

We used the ESRI shape file technical report to implement a *software module* to extract the *polyline* and *polygon* features of ESRI-Based maps and designed a dataset to contain the extracted data [ESRI 05]. Fig.2.1 shows the structure of ESRI shape file, it consists of three files for each feature layer (*.shp, *.shx, and *.dbf) the first one containing the geographical coordinates, the second is an index between each feature and its description in the database file, and the last containing a description on each polygon feature.

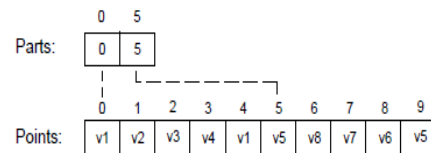


Table 1
Polygon Record Contents

Position	Field	Value	Type	Number	Byte Order
Byte 0	Shape Type	5	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little

Note: $X = 44 + 4 * \text{NumParts}$

Fig.1 the shape file structure for Polygon feature.

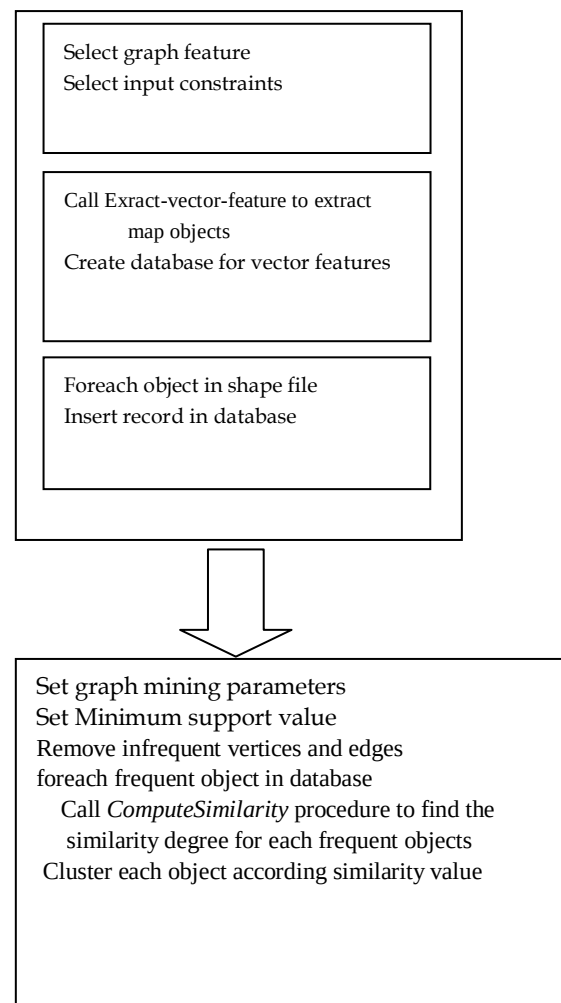


Fig2. The block diagram of the main steps of the proposal.

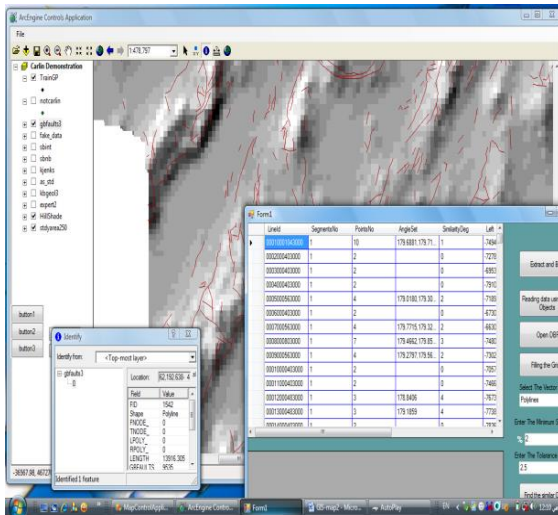


Fig.3. The dataset extracted from the shape file

Fig.3 shows part of the dataset which is built to hold the spatial-vector data that extracted from the GIS shape file by our module *Extract-vector-feature*. This data will be passed to the data mining algorithm **GIS-span** to extract the similarities between polygons and polylines map-objects.

Fig.4 shows part of the output of our Graph-mining algorithm *GIS-span* which is applied on ESRI map layer for the selected map objects features.

When we applied the proposed algorithm on the map depicted in Fig.4 we have found that it consists of 1674 objects and the is clustered to 74 clusters after excluding the objects that are not respond to the preprocessing constraints.

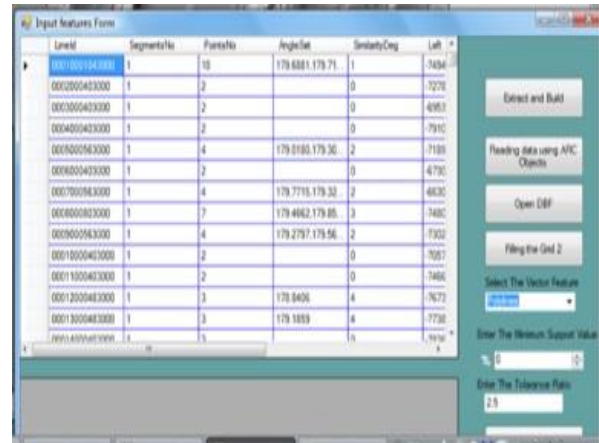


Fig.4 The dataset with similarity degrees

Conclusions

In this paper we present a new algorithm for mining the GIS vector data(polylines, polygons, and ,etc).The algorithm clusters map objects based on the degree of geometrical similarity among map objects. The results of applying this algorithm on actual map data gives good clustering results with less computing costs because of the simple geometrical similarity measure.

The algorithm may need some enhancement to deal with mining all vector types of map objects. The algorithm is also capable to deal with vector data stored in shape files and not related to GIS system like auto CAD and other drawing tools to cluster their map objects as a type of knowledge discovery to help designers analysing their maps.

References

- 1- Akshara Pande Manjari Gupta A.K. Tripathi, Design Pattern Mining for GIS Application Using Graph Matching Techniques 978-1-4244-5540-9/10,2010 IEEE.

- 2- C. Fraikin and P. Van Dooren, *Graph matching with type constraints*, ECC 07, Greece, July 2-5, 2007.
- 3- Chen Guang-xue, LI Xiao-zhou, Chen Qi-feng, LI Xiao-zhou, *Clustering Algorithms for Area Geographical Entities in Spatial Data Mining*, 978-1-4244-5934-6/10/2010 IEEE, (FSKD 2010).
- 4- Deepti Joshi, Leen-Kiat Soh *Member, IEEE*, and Ashok Samal, *Member, IEEE*, Redistricting using Constrained Polygonal Clustering. Digital Object Identifier 10.1109/TKDE.2011.140.
- 5-ESRI Shapefile Technical Description An ESRI White Paper—July 1998 Copyright © 1997, 1998 Environmental Systems Research Institute, Inc.
- 6- Ilaria Bordino, Debora Donato, and Aristides Gionis Mining large networks with subgraph counting, 2008 Eighth IEEE International Conference on Data Mining
- 7- Kang-Tsung Chang, *Programming ArcObjects with VBA Second Edition*, Chapman & Hall/CRC Taylor & Francis Group 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, FL 33487-2742 © 2008 by Taylor & Francis Group, LLC.
- 8-Sayan Ranu and Ambuj K. Singh. Graphsig: A scalable approach to mining significant subgraphs in large graph databases. In *ICDE*, 2009.

Estimating MPEG-4 Video Transmission over Gilbert Wireless Channels

Ghaida. A. AL-Suhail

Computer Engineering Department, University of Basrah, Basrah, Iraq

0780 60 12 354

ghaida_alsuhail@yahoo.com

Sarah A. Subber

0770 277 26 40

0770 277 26 40

sarah_subberidsco@hotmail.com

Abstract

In this paper, the aim is to propose an analytical packet loss models to estimate the video quality for MPEG-4 video transmission over error-prone correlated wireless channels. Two packet loss models are considered at the packet level in the network layer using TCP-Friendly protocol: (i) Gilbert-Elliott Model (GEM) and (ii) Extended Gilbert Model (EGM). The resultant video quality in terms of playable frame rate (PFR) is estimated at the end client over wireless cellular network. The obtained results point out the effect of packet correlation factor on the perceived video quality under different conditions of packet loss probability. It is found that the proposed EGM model outperforms in estimating the perceived video quality.

Keywords— Gilbert-Elliott, Extended Gilbert, MPEG-4 Video, TCP-Friendly, Wireless Network

Introduction

The IP packet-based wireless cellular systems, like Universal Mobile Telecommunications Systems (UMTS) and High-Speed Uplink/Downlink Packet Access (HSUPA/HSDPA) technologies are recently developed for high data rate transmission [1][2][3]. However, a packet-loss process in such networks is still a crucial issue in estimating the probability of

correctly received video packets at the clients.

The real-time video transmission over the internet and wireless channel in heterogeneous packet networks is subject to various errors types. In wired networks (such as the Internet), packets are lost mainly due to buffer overflows (congestion) at the routers, meanwhile in the wireless hop, packets are often lost due to random bit errors caused by channel variations

like noise, co-channel interference, mobility, and multi-path fading effects. Thus maintaining video quality in wireless environment is very challenging and it requires more strategic planning and robust video transmission which is adaptable to the network and channel conditions [4][5].

More specifically, understanding the packet-loss process in IP-based packet networks and providing an accurate mathematical model to describe it are of great importance for the design and performance analysis of network applications (real-time applications). One way of describing the packet-loss process is to concentrate on the observed packet loss by itself without considering explicitly the causes that lead to such process [5] [6]. Thus, Markov process (chain) based models are classified in this category. In k -th order Markov chain model characterizing a loss process, the loss of every packet is assumed to be dependent only on the loss status of the previous k packet transmissions. The simplest Markov chain model is the case $k=0$, known as the Bernoulli model. In this model, the packet loss probabilities are independent of each other [7][8]. Another simple Markov chain model is the case $k=1$, known as the Gilbert model [9], which has been widely used to model end-to-end packet loss processes for real-time applications [6]- [8]. The former model has also been extended to Gilbert-Elliott model [10].

Although these low-complexity Markov chain models have been frequently used to model such processes [7][8], the accuracy of these Markovian models in capturing the correlation characteristics of real-world packet-loss processes needs to be investigated. Sanneck et al. [11] develop a different Markov chain model, called the extended Gilbert model. There are two categories of extended Gilbert models; those which describe reception run-lengths (RRL) and those which describe loss run-lengths (LRL). In our approach, we concentrate on RRL extended Gilbert models which is derived by Wu and Radha [12] to be applied for MPEG-4 video packet loss process over wireless channel.

On the other hand, many researches [13]-[18] have devoted on video streaming based on TCP/UDP protocols in different wired/wireless IP-packet networks. Specifically, TCP has become popular protocol because of its easy handling and deployment. TCP features in order delivery and reliable end-to-end transport, which makes additional tools like error concealment, unnecessary at the clients. In low-latency networks, TCP introduces good throughput performance and low end-to-end delays, which makes TCP-based interactive services possible. In [18], for example, they propose a client-driven video transmission scheme by utilizing multiple HTTP/TCP streams. In [3], they evaluate video streaming

based on TCP-Friendly Rate Control (TFRC) using Bernoulli packet loss model for a single and Multiple TFRC flows over a 1xRTT CDMA in UMTS network.

In this paper, we focus on video traffic controlled by TCP-Friendly protocol over cellular

network for the advantages of this protocol: (i) it does not cause network instability and avoid congestion collapse. (ii) it is fair to TCP flows, which is the dominant source of traffic on the internet, and finally (iii) this protocol has lower fluctuation compared to TCP. That makes TCP more appropriate for Real time applications, which requires a constant video quality [3]. To our best knowledge, the studies [13]-[18] did not investigate the extended Gilbert model for TCP video transmission over wireless channel. Therefore, we propose an approach to introduce the two packets loss models: Gilbert-Elliot Model (GEM) [7]-[10] and Extended Gilbert Model (EGM) [12]. The models employ TCP-Friendly Rate Control for MPEG-4 video

transmission, and then estimate the resultant video quality in terms of playable frame rate at the client end.

Network Preliminaries

MPEG Standard

In the last years, several video standards have been developed for 3G mobile multimedia communications like H.263, MPEG-4, and H.264/AVC [8]. MPEG video is one of the most commonly used video compression standard which is encoded into three different types of frames- I (Intra-coded), P (Predictive-coded), and B (Bi-directional Predictive coded) frames.

In this standard, I-frame is independently coded, while P-frame is coded based on the prediction of object movement of the previous I or P frame. The B-frame is coded based on the differences between the previous I or P frame and the next I or P frame. Thus, there is a certain dependency relationship between I, P, B frames as shown in Fig. 1 [13][17].

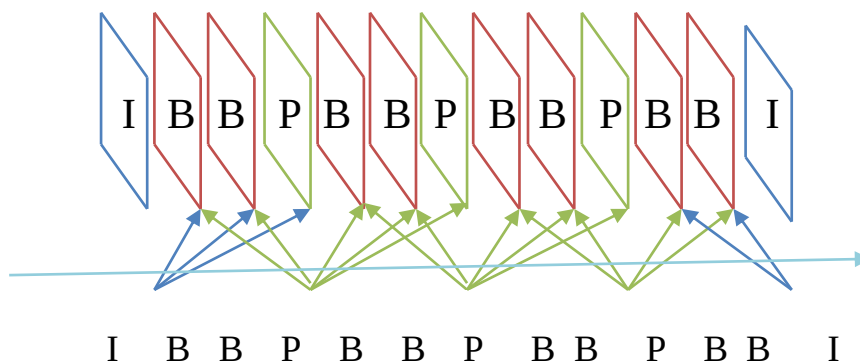


Figure1. Frame dependency

TCP-Friendly Throughput

We consider a video flow in a point-to-point network which is simply composed of one base station (access point) in UMTS and a single user end. This last wireless link is connected to a wired Internet via this base station [3][14]. By adjusting the sending rate to the desirable rate determined by an underlying TCP-Friendly Rate Control (TFRC), one can achieve the required quality of service (QoS) of video applications over a wireless link. Thus the normalised available bandwidth of a TFRC video session with respect to TCP packet size can be expressed as,

$$T = \frac{S}{t_{RTT} \sqrt{\frac{2P}{3}} + t_{RTO} (3\sqrt{\frac{3P}{8}}) P(1+32)} \quad (1)$$

S is the TCP packet size [byte], P stands for the average packet loss probability, i.e., loss event rate due to only the channel bit errors and there is no buffer overflow effect at the base station. The t_{RTT} is the round-trip time [sec], and t_{RTO} is the TCP retransmit time out value [sec].

GOP Rate

Since (1) provides an expression for T as TFRC throughput, then GOP rate can be written as;

$$G = \frac{T/S}{(S_I + S_{IF}) + N_P(S_P + S_{PF}) + N_B(S_B + S_{BF})} \quad (2)$$

where S_I is the size of I-frame, S_P is the size of P-frame, S_B is the size of B-frame, N_P represents the number of P-frame in a GOP, N_B represent the number of B-frame in a GOP [13].

Playable Frame Rate

The total PFR by Wu et al.'s, technique is given by [13,14];

$$PFR = G \cdot P_{SI} \cdot (1 + \frac{P_{SP} - P_{SP}^{N_P+1}}{1 - P_{SP}}) + N_{BP} \cdot P_{SB} \cdot (\frac{P_{SP} - P_{SP}^{N_P+1}}{1 - P_{SP}} + P_{SI} \cdot P_{SP}^{N_P}) \quad (3)$$

Where P_{SI} , P_{SP} and P_{SB} are the probabilities of success transmission of I, P, or B frame. The detail of the derivation of this equation is outlined in.

The Proposed Approach

GEM-TFRC Video Model

Recently many studies used the measurements of burst error over a wireless channel by the well known Gilbert-Elliot model (GEM) [7][10]. In the present work, we further assume GEM as virtual channel with two nodes, to improve the video quality by estimating the playable frame rate. The state diagram of the model is shown in Figure 2, the model represents two states; the good state (good packet P_0) and the bad state (bad packet P_1).

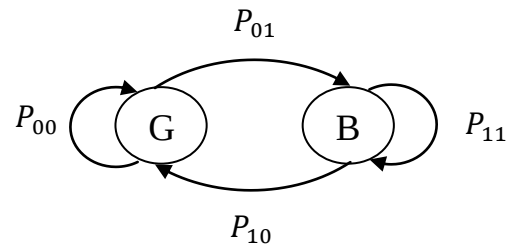


Figure 2. Gilbert Elliot state diagram for packet level

The P_{01} is the state probability for the transition from good to bad, and P_{10} is the state probability from the transition from bad to good. The π_1 and π_0 study state probability being in state P_1 and P_0 and $\pi_1 = \frac{P_{01}}{P_{01} + P_{10}}$, $\pi_0 = \frac{P_{10}}{P_{01} + P_{10}}$ respectively. The average packet loss rate product by the GE model is [7]-[10];

$$P_{avg} = P_0\pi_0 + P_1\pi_1 \quad (4)$$

The GEM is memory less model, where Packet error is produced by a sequence of independent trial. Each packet has P_{avg} being flipped and $1 - P_{avg}$ being successfully transmitted, P_{avg} is then the packet loss probability for the wireless channel. The network throughput becomes;

$$T = \frac{S}{t_{RTT} \sqrt{\frac{2P_{avg}}{3}} + t_{RTO} (3 \sqrt{\frac{3P_{avg}}{3}}) P_{avg} (1+32)} \quad (5)$$

Giving N packet being transmitted over a Gilbert Elliott channel, with K packet received, the probability of successful transmission is;

$$P_{suc} = \sum_{i=K}^N \binom{N}{i} (1 - P_{avg})^i P_{avg}^{N-i} \quad (6)$$

With a specific I, P, and B frame sizes, Equation (5) can be used to compute the probabilities of successful transmission for each frame type as following;

$$GEMP_{S_I} = \sum_{i=K}^{S_I} \binom{S_I}{i} (1 - P_{avg})^i P_{avg}^{S_I-i} \quad (7)$$

$$GEMP_{S_P} = \sum_{i=K}^{S_P} \binom{S_P}{i} (1 - P_{avg})^i P_{avg}^{S_P-i} \quad (8)$$

$$GEMP_{S_B} = \sum_{i=K}^{S_B} \binom{S_B}{i} (1 - P_{avg})^i P_{avg}^{S_B-i} \quad (9)$$

The total PFR using GEM-TFRC is expressed as

$$\begin{aligned} PFR_{GEM-TFRC} = & G * GEMP_{S_I} \cdot \left(1 + \frac{GEMP_{S_P} - GEMP_{S_P}^{N_P+1}}{1 - GEMP_{S_P}}\right) \\ & + N_{BP} * GEMP_{S_B} \cdot \left(\frac{GEMP_{S_P} - GEMP_{S_P}^{N_P+1}}{1 - GEMP_{S_P}} + GEMP_{S_I} * GEMP_{S_P}^{N_P}\right) \end{aligned} \quad (10)$$

EGM-TFRC Video Model

The Second proposed model for packet loss probability is a model which was derived by Wu and Radha [12] have been used, where the

authors extends the two-state Gilbert model using the number of correctly received packets as a indexes for Gilbert states.

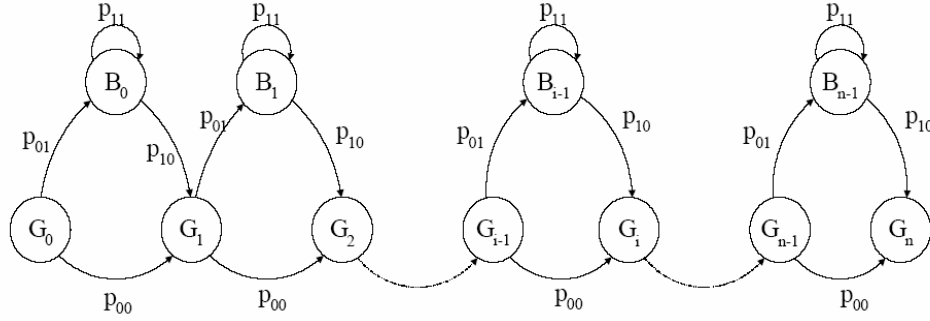


Figure 3. The state transition diagram of Extended Gilbert Model (EGM) [12].

The Extended Gilbert model is build on the probability of receiving correctly K packets among N packets transmitted over Gilbert is giving by [12];

$$\phi(N, K) = \pi_0(\phi_{G_0 G_K}(N) + \phi_{G_0 B_K}(N)) + \pi_1(\phi_{B_0 G_K}(N) + \phi_{B_0 B_K}(N)) \quad (11)$$

Where

$$\phi_{G_0 G_K}(N) = \begin{cases} \sum_{m=1}^K \binom{K}{m} \binom{N-K-1}{m-1} p_{01}^m p_{10}^m p_{00}^{K-m} p_{11}^{N-K-m} & 0 < K < N \\ 0 & K=0 \\ p_{00}^N & K=N \end{cases} \quad (12)$$

$$\phi_{G_0 B_K}(N) = \begin{cases} \sum_{m=0}^K \binom{K}{m} \binom{N-K-1}{m-1} p_{01}^{m+1} p_{10}^m p_{00}^{K-m} p_{11}^{N-K-m-1} & 0 < K < N \\ 0 & K=N \end{cases} \quad (13)$$

$$\phi_{B_0 G_K}(N) = \begin{cases} \sum_{m=1}^{K-1} \binom{K-1}{m} \binom{N-K}{m} p_{01}^m p_{10}^{m+1} p_{00}^{K-m-1} p_{11}^{N-K-m-1} & 0 < K < N \\ 0 & K=0 \end{cases} \quad (14)$$

$$\phi_{B_0 G_K}(N) = \begin{cases} \sum_{m=0}^{K-1} \binom{K-1}{m} \binom{N-K}{m+1} P_{01}^{m+1} P_{10}^{m+1} P_{00}^{K-m-1} P_{11}^{N-K-m-1} & 0 < K < N \\ P_{11}^N & K = \\ 0 & K = \end{cases} \quad (15)$$

The state probability P_{01} and P_{01} (or $P_{00} = 1 - P_{01}$ and $P_{11} = 1 - P_{10}$) are the effective parameters on the evaluation of $\phi(N, K)$. More useful parameters can be involved to improve the performance. In [12], the author used the average loss rate P , and the packet correlation

ρ , to give another perception to the state transition probabilities, where packet correlation is the correlation between two errors symbols P_{00} and P_{11} , which can be expressed as;

$$P_{01} = P(1 - \rho) \quad (16)$$

$$P_{10} = (1 - P)(1 - \rho) \quad (17)$$

The computation of the successful transmission probabilities for a specific I , P , and B frame sizes can be written as;

$$EGMP_{S_I} = \phi(S_I, K) \quad (18)$$

$$EGMP_{S_P} = \phi(S_P, K) \quad (19)$$

$$EGMP_{S_B} = \phi(S_B, K) \quad (20)$$

Then the total PFR using EGM-TFRC can be written as;

$$PFR_{EGM-TFRC} = G.EGMP_{S_I} \cdot \left(1 + \frac{EGMP_{S_P} - EGMP_{S_P}^{N_P+1}}{1 - EGMP_{S_P}}\right) + N_{BP} \cdot EGMP_{S_B} \cdot \left(\frac{EGMP_{S_P} - EGMP_{S_P}^{N_P+1}}{1 - EGMP_{S_P}} + EGMP_{S_I} \cdot EGMP_{S_P}^{N_P}\right) \quad (21)$$

Simulation Results

Methodology

Based on the above assumption, we develop the following illustrative steps to find the optimal

playable frame rate based on the given models GEM-TFRC video model, and EGM-TFRC video model) mentioned in Section II.

1. Giving a specified state probability one can determine the from equation 3. For each set of other system variables, compute the PFR using equations 4 and 12. For EGM-TFRC, assign specific average loss rate p , and packet correlation, determine PFR using (4) and (21). We characterize the variation of the PFR in both Models in term of two dimensions; Packet loss Probability and PFR dimension, to find the optimal PFR.
2. Used the four different cases of transmitted and received packet with a certain probability of packet loss, to find the optimal PFR.
3. Giving a combination of packet correlation, and a set of an average loss probability, compute the PFR using (21). Characterize the variation of the PFR in term of two dimensions: packet correlation and PFR. To investigate the effect of packet correlation on the PFR, in four different cases transmitted and received packet.

System Settings

Table 1 describes network characteristic of many typical network connection.[2] in

which we used in our simulation. Encoded video is transmitted as a repeating sequence of GOP (N_P, N_{BP}) pattern, where $N_P=3$ and $N_{BP}=2$. The P_{00}, P_{11} and P_0 are set at 0.96, 0.94 and 0.001 respectively. The packet error rate is set between 0.001 to 0.1 with 0.001 intervals. Also, the packet correlation is set between 0.1 to 0.9 with 0.1 intervals.[13][12][17]

Table 1: Network Settings [3][13][12]

Parameter	Value
t_{RTT}	168ms
t_{RTO}	$t_{RTT} \times 4$
S_I	24.64KB ~ 25 packet
S_P	7.25KB ~ 8 packet
S_B	2.45KB ~ 3 packet

Table 2 MPEG Sending, Receiving Frames Size

	Sending	Receiving
	(N_I, N_P, N_B)	(K_I, K_P, K_B)
1	(25,8,3)	(25,8,3)
2	(25,8,3)	(20,6,1)
3	(25,8,3)	(25,6,1)
4	(25,8,3)	(24,6,1)

Performance Evaluation

This section describe the PFR of an MPEG4 Video streaming over wireless channel using two packets lost models; the GEM-TFRC video model and EGM-TFRC video

model. The performance evaluation is carried out by considering the parameters in Tables 1, 2 and 3 for the fact that maximum frame rate allowed over internet is 30fps. Result was conducted using Matlab programming.

Figure 4 shows the PFR over the effect of packet error rate, using proposed models.

The PFR clearly indicates better performance in the range from 0.02 to 0.1 packet loss probability. Meanwhile, the GEM-TFRC becomes closer to the EGM-TFRC at packet loss probabilities below 0.02. However, the special case of the GEM-TFRC for which the correlation parameter had been added, gives us a poor performance in comparison with the other.

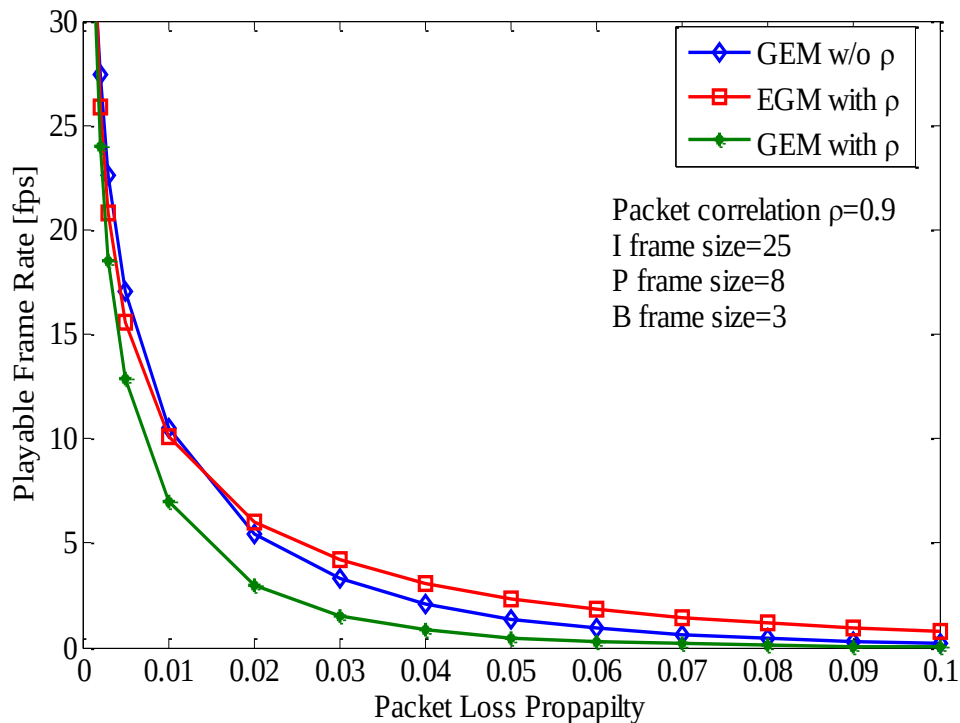


Figure 4 The Playable Frame Rate comparison using GEM-TFRC without memory, GEM-TFRC with memory, and EGM-TFRC models

Table 3 summaries different sending and receiving packets that are applied in the two models for video streaming. It is clearly noticed that there are two effects as follows:

dependency relationship between frames in GOP, and packet correlation factor. The PFR achieves the highest value at $P=1\%$ and in contrary there is a clear degradation in video

quality when packet loss increases to 5%. However, EGM-TFRC model introduces nonzero values of PFR when sending (25,8,3) and receiving (20,6,1); and this result obtained is caused due to the packet correlation factor.

Figure 5 shows the PFR over the effect of correlation factor in EGM-TFRC, when sending (NI=25, NP=8, NB=3) and receiving (KI=25, KP=8, KB=3) packets, when the packet correlation is low, it will produce low PFR; but this is not always the case when the `correlation

increase. In Fig. 5 we consider a case where a packet loss occur, where sending packet sizes are (NI=25, NP=8, NB=3), receiving packets (KI=20, KP=6, KB=1), as shown in Fig. 6 it is clearly shown that when P set to 1% we have got higher PFR at $\rho = 0$. The fluctuation in PFR values is also clearly noticed when ρ increases to 0.9. The PFR has approximately constant value over a combination of P ; as the correlation ρ is strong. Once the process initially starts in bad or good state, it has the inertia to stay at that state.

Table 3: The MPEG4 sending and receiving Frame sizes and PFR for $P=1\%$ and $P=5\%$.

GEM-TFRC			
Sending	Receiving	PFR $P=1\%$	PFR $P=5\%$
(N_I, N_p, N_B)	(K_I, K_p, K_B)		
(25,8,3)	(25,8,3)	10.5119	1.37873
(25,8,3)	(20,6,1)	0.00	0.0005
(25,8,3)	(25,6,1)	1.4404	0.3022
(25,8,3)	(24,6,1)	0.0044	0.0062
EGM-TFRC			
Sending	Receiving	PFR	PFR
(N_I, N_p, N_B)	(K_I, K_p, K_B)	$P=1\%$	$P=5\%$
(25,8,3)	(25,8,3)	10.0929	1.3787
(25,8,3)	(20,6,1)	0.0028	0.0042
(25,8,3)	(25,6,1)	1.2677	0.3022
(25,8,3)	(24,6,1)	0.2366	0.2416

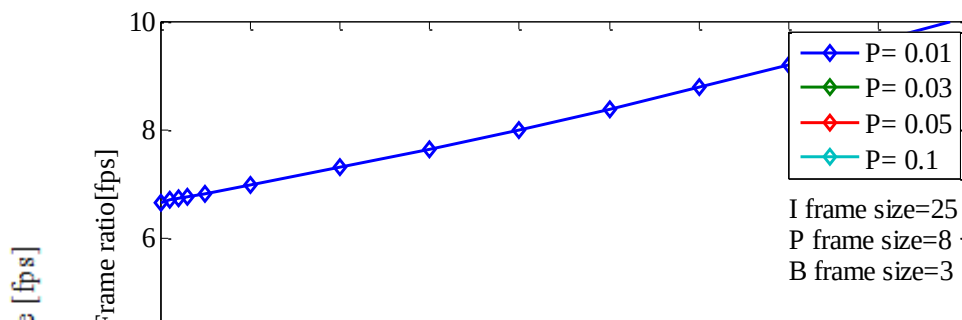


Figure 5. The playable frame rate over the effect of correlation factor in EGM-TFRC video model when sending (NI=25, NP=8, NB=3) and receiving (KI=25, KP=8, KB=3)

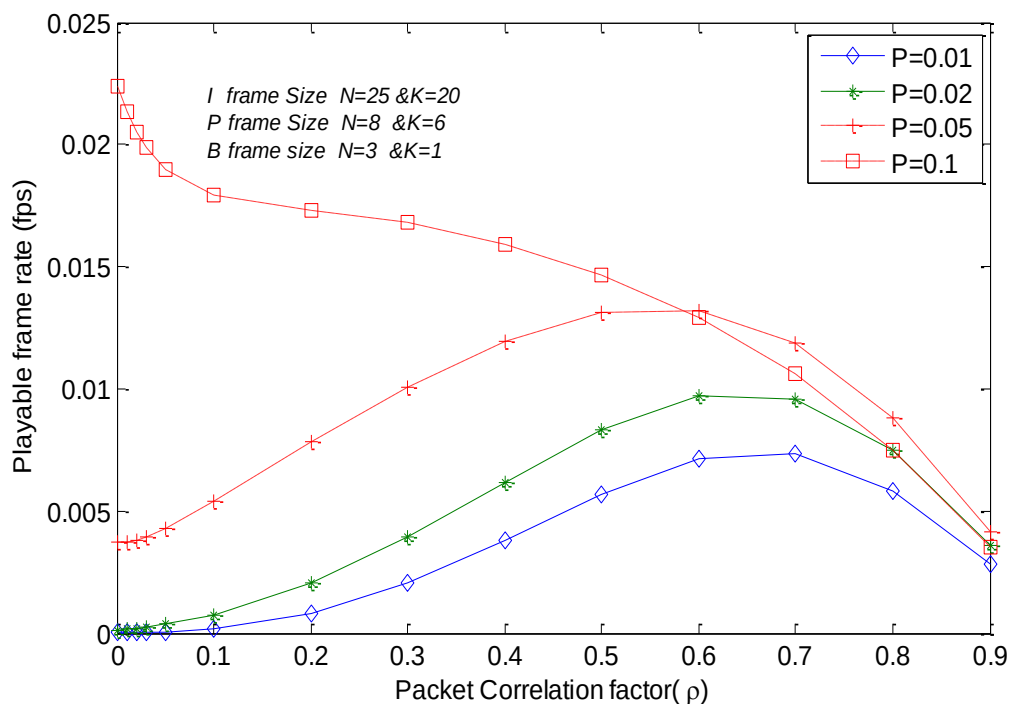


Figure 4.b The playable frame rate over the effect of correlation factor in EGM-TFRC video model when sending (NI=25, NP=8, NB=3) and receiving (KI=20 ,KP=6 ,KB=1)

Conclusion

In this paper, we have proposed analytic packet loss models to estimate video quality over wireless channels with highly correlated errors

using TFRC protocol for MPEG-4 streaming. The models are based on Gilbert Elliot and Extended Gilbert model of packet loss probability. The results of GEM-TFRC video

model show good performance in high packet loss probability; meanwhile GEM and EGM both show reasonable performance as well but in the range of average packet loss rate below 2%. As a result, we conclude that these proposed models can be helpful to predict the video quality using packet loss models over wireless channels. The Future work can be extended to involve the FEC techniques the two to improve the performance of video quality under different conditions of wireless channels.

References

- [1] H. Holma and A. Toskala(2006), HSDPA /HSUPA for UMTS High Speed Radio Access for Mobile Communications, John Wiley, England.
- [2] Q. Liu, S. Zhou, and G. B. Giannahis(2005), "Queuing with Adaptive Modulation and Coding over Wireless Links: Cross-Layer Analysis and Design," IEEE Trans. Wireless Comm., vol. 4, no. 3.
- [3] M. Chen and A. Zakhor(2005), "Rate Control for Streaming Video over Wireless", IEEE Wireless Communication, pp. 32-41.
- [4] A. Khan, L. Sun, E. Ifeachor, J.O. Fajardo, F. Liberal (2010), "Video Quality Prediction Model For H.264 Video over UMTS Networks and their Application in Mobile Video Streaming", IEEE Communication (ICC).
- [5] H. Han(2010), "Performance Improvement of TCP Reno Based on Monitoring the Wireless Packet Loss Rate", Physics & Electronics Engineering Department, Xiangfan University, China.
- [6] X. Yu, W. Modestino and X. Tian(2005), "The Accuracy of Gilberts Models in Predicting Packet-Loss Statistics for Single-Multiplexer Network Model", IEEE International Conference.
- [7] G. Haßlinger and O. Hohlfeld (2008), "The Gilbert-Elliott Model for Packet Loss in Real Time Services on the Internet", Darmstadt University of Technology, Germany.
- [8] Y. Baguda, N. Fisal, S. Syed, S. Yusof, M. Abdullah, and A. Zulkarmawan(2008)" Mobile Streaming of H.264 Video over Gilbert-Elliotts Channel ", World Academy of Science, Engineering an Technology 48.
- [9] E. N. Gilbert(1960), "Capacity of a Burst Noise Channel", Bell Sys. Tech. Journal, vol. 39, p. 1253.
- [10] E. O. Elliott (1963) "Estimates of Error Rates for Codes on Burst-Noise Channels," Bell System Technical Journal, vol. 42, pp. 1977-1997.
- [11] H. Sanneck, G. Carle, and R. Koodli(2000) " A Framework Model for Packet Loss Metrics Based on Loss Run Length," Proceedings of SPIE/ACM SIGMM Multimedia Computing and Networking Conference.
- [12] M. Wu and H. Radha(2005) "Network Embedded FEC for Overlay and P2P Multicast

over Channels with Memory,” Conf. on Information Sciences and Systems, The John Hopkins University,.

[13] H. Wu, M. Claypool, R. Kinicki (2003) “A Model for MPEG with Forward Error Correction and TCP-Friendly Bandwidth ”, In Proceedings of Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV),Monterey, California, USA.

[14] Y. Yuan, B. Cokburn, T. Sikora, and M. Mandal(2006) “A GoP Based FEC Technique for Packet Based Video Streaming”, Proceedings of the 10th WSEAS International Conf. on Communications, Greece.

[15] Y. Shan, S. Yi, S. kalyanaraman, and J. Woods(2009) “Two-stage FEC Scheme for

Scalable Video Transmission over wireless Network ”Image comm.24 (9):718-729.

[16] J. R. Yee, E. J. Weldon(1995) “Evaluation of the Performance of Error-Correcting Codes on a Gilbert Channel”, IEEE Trans. on Communications, vol. 43, no. 8, pp. 2316-2323.

[17] C. H. Lin, C. H. Ke, C. K. Shieh(2004)“A Study of MPEG Video Transmission in Lossy Wireless Networks”, Electrical Engineering Department, National Cheng Kung University, Taiwan.

[18] R. Kusching, I. Kofler, and H. Hellwanger(2010) “Improving Internet Video Streaming Performance by Parallel TCP-based Request Response Streams”, Proc. of IEEE CCNC.

تقييم إرسال الصورة المرئية نوع MPEG-4 في قنوات جلبت اللاسلكية

غيداء عبد الرزاق السهيل

سارة عبد الرضا صبر

قسم هندسة الحاسبات كلية الهندسة، جامعة البصرة

الخلاصة

في هذا البحث، الهدف هو اقتراح نماذج تحليلية لفقد الرزمة لغرض مقايسة جودة الصورة المرئية من نوع MPEG4 المرسل عبر القنوات اللاسلكية ذات الاخطاء المترابطة. تم اقتراح نموذجين من فقدان الرزمة عند مستوى الرزمة في طبقة الشبكة باستخدام بروتوكول النقل TCP-Friendly; نموذج جلبت - البوت (2) نموذج جلبت الموسع. يتم تقييم جودة الصورة الناتجة بدلالة معدل الفريم العامل عند نهاية المستخدم في الشبكة الخلوية اللاسلكية. النتائج المستحصلة تبين تأثير معامل ترابط الرزمة على جودة الصورة المرئية المحسوسة تحت ظروف مختلفة من احتمالية فقدان الرزمة. وقد وجد ايضاً ان نموذج جلبت الموسع قد تفوق في مقايسة جودة الصورة المستلمة.

Computer Network Routing Using Fuzzy Neural Networks

Khlood Ahmad Nasar Turki Y. Abdalla Abdulkareem Y. Abdalla
Computer Science Dept. Computer Engineering Dept. Computer Science Dept.
University of Basrah, Basrah, Iraq

Abstract

Fuzzy neural networks can be used efficiently as a solution technique for routing problem in computer networks, to improve the performance. In this paper, two methods are proposed to solve this problem. The fuzzy neural networks are used to make routing decision, and to manage the congestion with solving of the routing problem. The proposed methods are applied for typical examples of computer networks. The fuzzy neural networks are trained. Results of the testing assert their high achievement.

1. Introduction

In a communication network information is transferred from one node to another as data packets. Packet routing is a process of sending a packet from its source node (s) to its destination node (d). On its way, the packet spends some time waiting in the queues of intermediate nodes while they are busy processing the packets that came earlier. Thus the delivery time of the packet, defined as the time it takes for the packet to reach its destination, depends mainly on the total time it has to spend in the queues of the intermediate nodes. Normally, there are multiple routes that packet could take, which means that the choice of the route is crucial to

the delivery time of the packet for any (source, destination) pair [1].

Successful operation of data communication network is critically dependent on the provision of an adequate routing algorithm. Routing algorithms are methods for finding the best way from a node s to another node d. This may be via a large number of other nodes or it may be in the next subnetwork. On a small, simple network the problem is almost trivial, statically allocating routes and defining them by hand, but when dealing with a huge internetwork such as the Internet this is not possible. It heavily interconnected network has many routes from one node to another, and these routes span many different types of link with different

bandwidth and latency characteristics. Calculating the best route through such a complex system is computationally intractable and impossible to do by hand [2, 3, 4].

If part of network becomes overfilled with packets it can become impossible for packets to move. The queues into which they should be accepted are always full. This is called congestion. Routing algorithms strongly interact with congestion [2, 5].

Considerable research has been devoted toward solving the routing problem in the computer networks. Paper of J. Zuo, S. Ng and L. Hanzo [6] propose fuzzy logic based method to specific route having the highest route stability is finally selected for data transmission. L. Lertsuwanakul [7] compare the shortest path routing and probability functions using deterministic or adaptive approach. M. Dehyadegari and S. Mohammadi [8] propose routing method using two multimedia applications and a random traffic profile. A. Haboush [9] investigates three metrics, the mean link bandwidth, queue utilization and the mean link delay to solve routing problem. A. Dana and N. Salehi [10] propose routing algorithm is choosing channel that has more free-slots input buffer beyond adjacent routers and the less number of active requester for a given output port. M. Chelliah and B. Sivaselvan [11] provide an efficient routing method is a fuzzy multi-constraint AODV routing for wireless mesh networks.

2. Routing Using Neural Networks and Fuzzy Logic

In computer networks, adaptive routing methods are recommended in order to efficiently select the transmission path of the packets without causing network congestion. Routing algorithms must be adapted to the traffic and topology changes. The information regarding these changes have inherent uncertainty because it is at least as old as the propagation delay between nodes [12, 13].

The traditional network routing algorithms have not attempted to deal with this uncertainty in the information available at a node. Consequently, these algorithms often make poorer decisions when trying to route to distant nodes. Also, they have incapability to take into account possible node or link failures. Furthermore, their implementation requires complex and lengthy calculations [4, 13].

The parallel, distributed processing structure of the neural networks and their ability to learn are justified to use it as available structure for solving the routing problem [4].

Uncertain notions are described and implemented by fuzzy set theory. Therefore, in fuzzy logic an element can reside in more than one set to different degrees of similarity, and a formalism for implementing expert or heuristic rules is provided as fuzzy systems which are transparent. Then, the fuzzy systems can be integrated in complete routing systems [13].

The main idea of fuzzy neural networks is to combine the advantages of fuzzy systems which are interpretability and use of prior knowledge for initialization with the learning capabilities of neural networks [14]. Therefore, the fuzzy neural networks provide a high level of the performance to make routing decision.

3. Proposed Methods

This paper presents two proposed methods to use fuzzy neural networks for solving the routing problem in the computer networks. Fuzzy neural network is located at each node of the computer network to make a local routing decision. In other method, the

congestion control is combined with solving of the routing problem.

The computer networks which are considered in this paper modeled as graphs. Two examples, the first is a 9- node mesh computer network (CN1) shown in Figure (1). While the second is a random computer network (CN2) shown in Figure (2). The imposing values of cost of the links (packet delay) and queue length of the nodes for these two computer networks are shown in Tables (1) and (2), respectively. The values of the queue lengths in these Tables are located under changing continuously, when the nodes send or receive the packets.

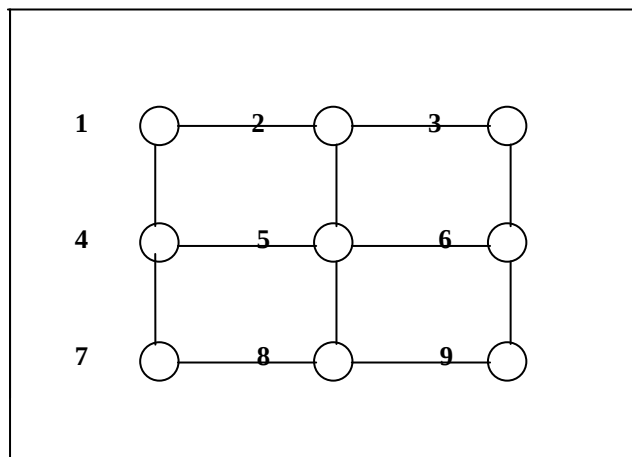


Figure (1) computer network (CN1)

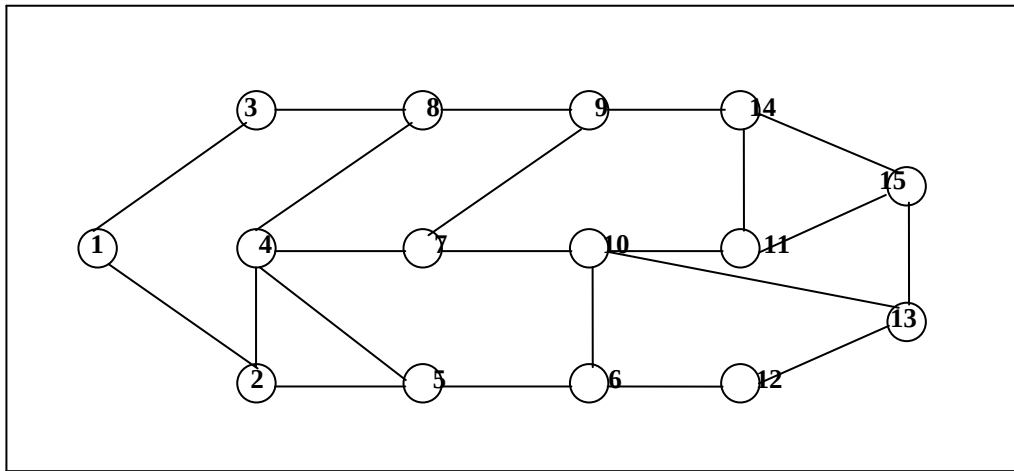


Figure (2) computer network (CN2)

Table (1) link costs and queue lengths of the computer network (CN1)

Links		Nodes	
link	cost (packet delay) (in second)	node	queue length (in packet)
1-2	10	1	6
1-4	2	2	5
2-3	3	3	8
2-5	15	4	4
3-6	8	5	7
4-5	2	6	7
4-7	1.6	7	4
5-6	1.2	8	6
5-8	2.1	9	6
6-9	1.5		
7-8	3		
8-9	1		

Table (2) link costs and queue lengths of the computer network (CN2)

Links		Nodes	
link	cost (packet delay) (in second)	node	queue length (in packet)
1– 2	6.312	1	5
1– 3	6.312	2	5
2– 4	1.544	3	7
2– 5	6.312	4	7
3– 8	6.312	5	4
4– 5	3.352	6	8
4– 7	6.312	7	8
4– 8	3.352	8	5
5– 6	12.624	9	6
6– 10	3.352	10	5
6– 12	6.312	11	4
7– 9	3.152	12	6
7– 10	3.152	13	3
8– 9	6.312	14	4
9– 14	12.624	15	6
10– 11	3.152		
10– 13	3.152		
11– 14	3.152		
11– 15	3.152		
12– 13	6.312		
13– 15	6.312		
14– 15	6.312		

3.1. Routing Decision

A fuzzy neural network (FN1) is designed as a part of system which also consists of central monitor, to solve the routing problem locally. It has three inputs,

they are link time cost, queue time cost and neighbor_ destination path length, and two outputs, they are neighbor time cost and destination path level. Where link time cost is the packet delay on link to the one neighbor

nodes, queue time cost is the waiting time of packet in queue of the one neighbor nodes, neighbor_ destination path length is the distance (number of the nodes) between the neighbor node of the router and the destination, neighbor time cost is the cost of sending packet by one neighbor node and

destination path level is level of evaluating the suitability of the path from the router to the destination by the neighbor node. With h units of product and normalization layers where $h = 48$ is the number of rules, as shown in Figure (3).

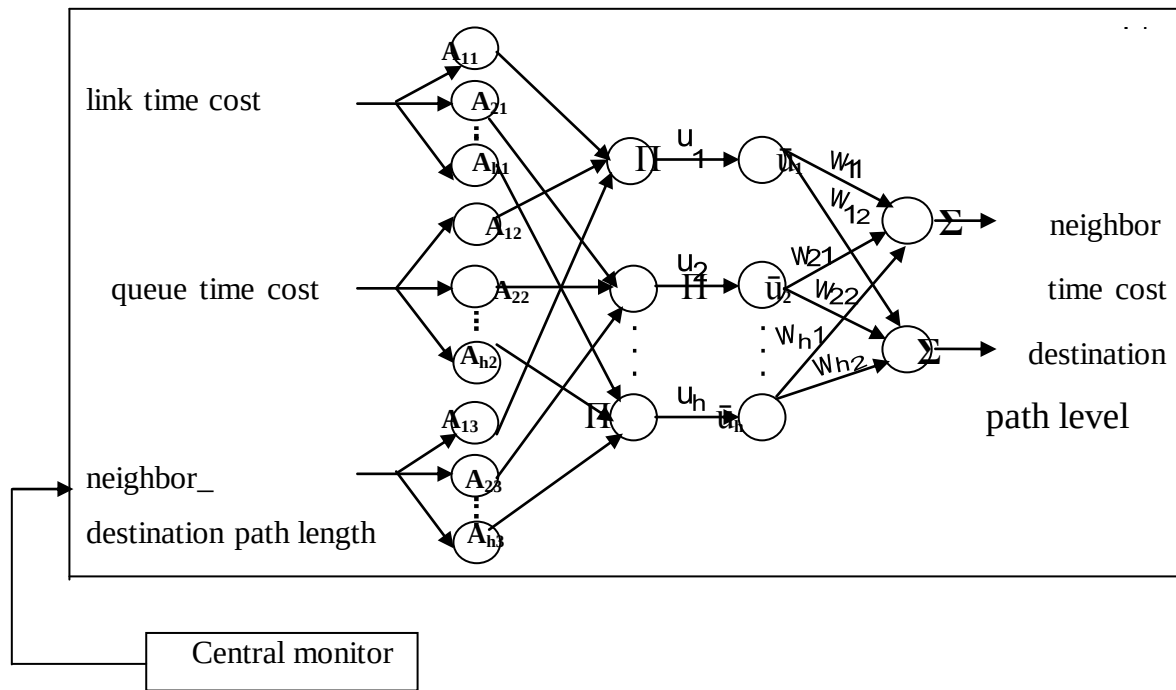


Figure (3) the fuzzy neural network (FN1)

The computer network contains the fuzzy neural network (FN1) at each node. When, node (j) has packet to send to a given destination node, the fuzzy neural network receives time cost of link and time cost of queue of the one neighbor nodes, and also receives length of the path from this neighbor node to the destination node from the central monitor. It decides the time cost of this neighbor node for sending packet by it and level of the path to the destination node. In r_j steps, the node (j) determines the best

neighbor node, where r_j is a number that represents the number of neighbor nodes to node (j). For example, in the computer network (CN1), if node 7 has packet to send to node 5. The fuzzy neural network in node 7 receives time cost of link, time cost of queue of the one neighbor nodes (4, 8) and path length from this neighbor node to destination 5, and determines the time cost of this neighbor node and level of the path to destination 5 by this neighbor node. Then, the routing system of node 7 determines its best

neighbor node to reach destination 5 quickly. The same process is repeated at each node successively till the destination node.

3.2. Routing and Congestion Control

Control

For solving the routing problem with congestion control, the system is designed as a combination of two fuzzy neural networks. The first, fuzzy neural network (FN2) has two

inputs, packet average level and packet variance level, and one output, the congestion prediction level. Where packet average level is level of estimating the average number of packets of a node, packet variance level is level of estimating the variance of packets of a node and congestion prediction level is level of determining the degree of congestion of a node, as shown in Figure (4) (where $h = 16$).

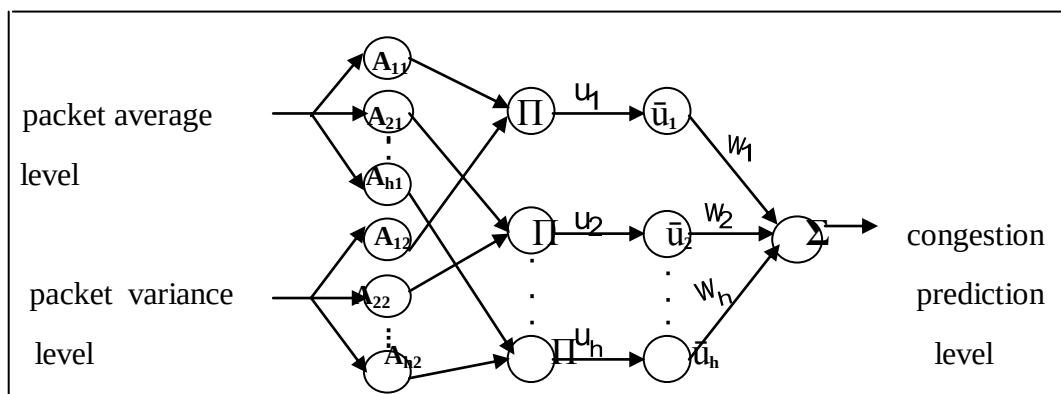


Figure (4) the fuzzy neural network (FN2)

The second, fuzzy neural network (FN3) has two inputs, link time cost and congestion prediction level which is the output of the first part (FN2), and one output, the neighbor

selection level. Where neighbor selection level is level of evaluating the suitability of the selected neighbor node, as shown in Figure (5) (where $h = 16$).

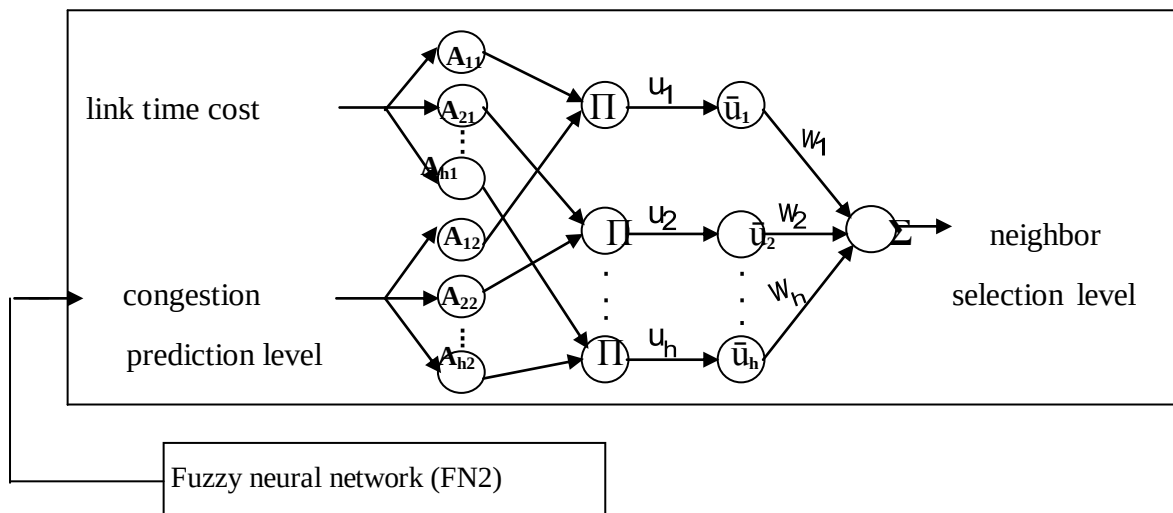


Figure (5) the fuzzy neural network (FN3)

The routing system, that consists of two fuzzy neural networks is included at each node of the computer network. In order for, node (j) to send packet to a given destination node, the congestion predictor (FN2) receives statistics data (average number of packets and variance of packets) of the one neighbor nodes and gives its output about the congestion state of this neighbor node to the other part (FN3) to determine the selection level of this neighbor node. In r_j steps, the system determines the best neighbor node to node (j). For example, in the computer network (CN2), if node 7 has packet to send to node 15, its congestion predictor (FN2) reports the congestion state of the one neighbor nodes (4, 9, 10) to the fuzzy neural network (FN3) which decides the selection level of this neighbor node. So, the selection level of each neighbor node can be determined. Then, the routing system at node 7 selects best neighbor node. Which has less value of congestion prediction with less cost of link. The same process is repeated at each node till the destination node.

3.3. Simulation Result

The simulation has been realized using C++ programming language, in order to test the proposed methods, which use fuzzy neural networks to solve the routing problem. Through that, they are applied for the two computer networks (CN1, CN2).

The proposed fuzzy neural networks have numbers of inputs, outputs, and units of product and normalization layers as described

in Table (3). For setting the parameters (weight, center and width) of these fuzzy neural networks the following method of initialization is used [15].

Step1_ For $k = 1, \dots, h-2$, the training sets $(x_1(k), \dots, x_m(k), y_1(k), \dots, y_p(k))$ which are inputs and desired outputs are presented, where h is the number of rules, m is the number of inputs, p is the number of outputs.

Step2_ For $k = 1, \dots, h-2, i = 1, \dots, m, j = 1, \dots, p$, the center $c_{ki} = x_i(k)$ and the weight $w_{kj} = y_j(k)$. For $k = h-1$, the center $c_{ki} = a_i$ and the weight $w_{kj} = 1$, and for $k = h$, the center $c_{ki} = b_i$ and the weight $w_{kj} = 1$, where a_i is the lower bound and b_i is the upper bound of the universe of discourse $[a_i, b_i]$ of the input x_i .

Step3_ For $k = 1, \dots, h, i = 1, \dots, m$, the width $\sigma_{ki} = \max(|c_{ki} - c_{ri}|, |c_{ki} - c_{li}|) / \sqrt{|\ln \lambda_i|}$, where c_{ri} is the center of membership function at right side for input variable x_i , c_{li} is the center of membership function at left side for input variable x_i , and λ_i is a factor, $0 < \lambda_i < 1$.

The membership function used to perform a fuzzification operation of the proposed fuzzy neural networks is Gaussian function as in equation:

$$A_{ij} = e^{-\frac{1}{2}((x_j - c_{ij}) / \sigma_{ij})^2} \quad (1)$$

where x_j is the input variable, for $(j = 1, \dots, m)$ inputs, c_{ij} and σ_{ij} denote the mean (center) and variance (width) with respect to A_{ij} .

These fuzzy neural networks are trained by backpropagation algorithm, when

the parameters (weight, center and width) are adjusted to minimize the error function. For every one of the computer networks (CN1, CN2), 30 training sets are taken. The training operation is continue, until the minimum value of mean squared error is obtained. For that, some values of learning rates (η_w , η_c , η_σ) and momentum rates (α_w , α_c , α_σ) are employed to get best convergence which are selected by trial and error, they are exposed in Table (4). Results obtained at this stage proof decreasing of the mean squared error with increasing of number of epochs, as shown in Figures (6)–(11).

The testing is performed on the trained sets, and on other test sets for each of the

computer networks (CN1, CN2). Some of the results of testing are exposed in Tables (5)–(9). Results are summarized in Tables (10) and (11) which show the efficiency of the performance of the fuzzy neural networks through the testing. These Tables describe the success rates to test trained sets and test sets.

From the simulation results, the following points are noticed.

The structure of each proposed fuzzy neural network is not related to the size of the computer network. The inputs and the outputs of each proposed fuzzy neural network are information about the one neighbor nodes in the computer network.

Table (3) the number of input, output and units of product and normalization layers of the proposed fuzzy neural networks

Fuzzy neural network	Number of inputs	Number of outputs	Number of units of product and normalization layers
FN1	3	2	48
FN2	2	1	16
FN3	2	1	16

Table (4) learning rates and momentum rates of the proposed fuzzy neural networks

Fuzzy neural network	Learning rate (η)			Momentum rate (α)		
	η_w	η_c	η_σ	α_w	α_c	α_σ
FN1	0.9	0.9	0.9	not used		
FN2	0.9	0.9	0.9	0.6	0.6	0.6
FN3	0.9	0.9	0.9	0.6	0.6	0.6

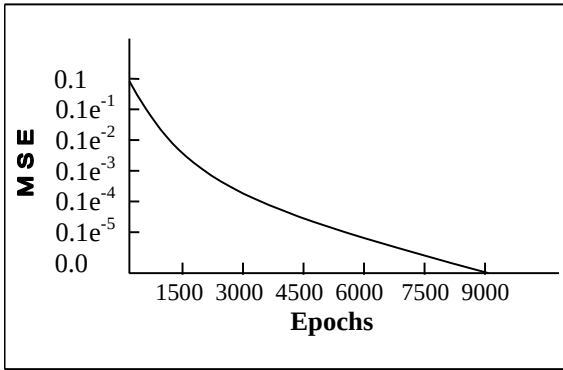


Figure (6) error versus number of epochs of the fuzzy neural network (FN1) for the computer network (CN1)

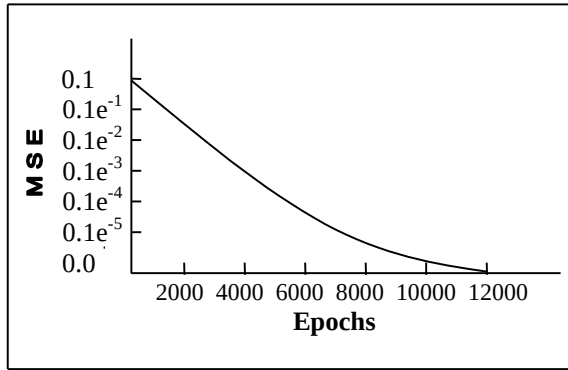


Figure (7) error versus number of epochs of the fuzzy neural network (FN1) for the computer network (CN2)

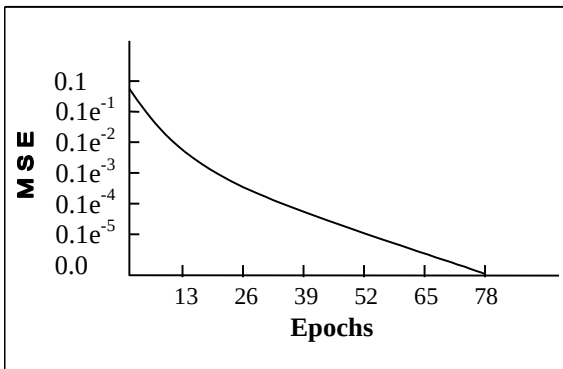


Figure (8) error versus number of epochs of the fuzzy neural network (FN2) for the computer network (CN1)

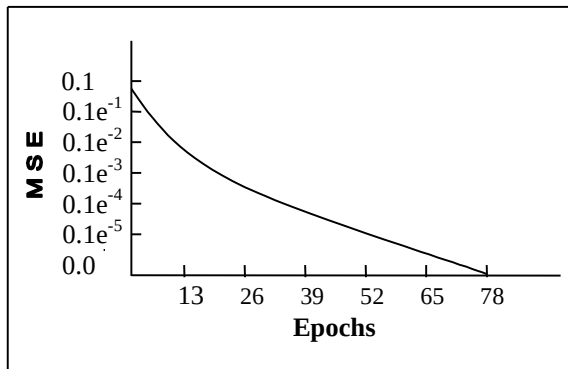


Figure (9) error versus number of epochs of the fuzzy neural network (FN2) for the computer network (CN2)

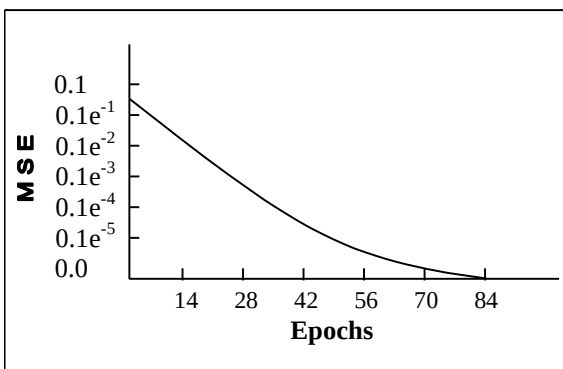


Figure (10) error versus number of epochs of the fuzzy neural network (FN3) for the computer network (CN1)

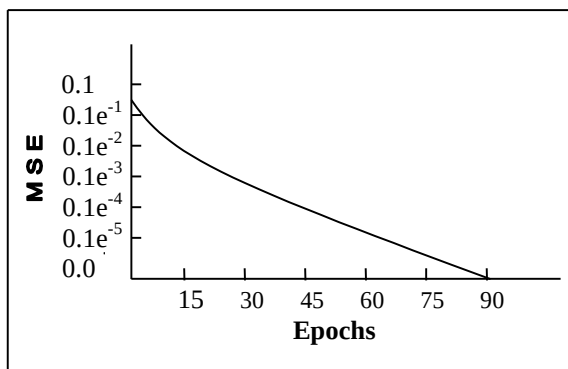


Figure (11) error versus number of epochs of the fuzzy neural network (FN3) for the computer network (CN2)

Table (5) some of the test results of the fuzzy neural network (FN1) for the computer network (CN1)

Inputs of FN1			Desired outputs		outputs of FN1	
Link time cost	Queue time cost ($Tq=Lq / P$, let $P=2$)	Neighbor_ destination path length	Neighbor time cost	Destination path level	Neighbor time cost	Destination path level
10	2.5	4	11.8825	18.3784	11.8821	18.3787
8	4	2	13	20	13.0035	19.9965
10	2.5	8	11.8825	23.3784	11.8821	23.3787
8	4	6	13	22.1441	13.0004	22.1444
1.6	4	2	5.6043	11.8626	5.6041	11.8629
15	5	5	23.9035	30.9035	23.8244	30.8244

Table (6) some of the test results of the fuzzy neural network (FN1) for the computer network (CN2)

Inputs of FN1			Desired outputs		outputs of FN1	
Link time cost	Queue time cost ($Tq=Lq / P$, let $P=2$)	Neighbor_ destination path	Neighbor time cost	Destination path level	Neighbor time cost	Destination path level
3.352	2.5	3	6.3277	12.4482	6.3277	12.4482
3.352	2.5	6	6.3545	14.6233	6.3512	14.6206
2.624	4	5	16.7822	23.7822	16.7864	23.7805
2.624	4	8	16.7822	33.7822	16.7864	33.7805
6.312	2.5	2	9.7045	15.6941	9.7042	15.6935
3.152	4	4	8.6055	14.533	8.3482	14.465

Table (7) some of the test results of the fuzzy neural network (FN2) for the computer networks (CN1, CN2)

	Inputs of FN2		Desired output	Output of FN2
	Packet average	Packet variance level	Congestion prediction	Congestion prediction
	4	0.1	1.46316	1.46316
	9	0.4	5.13696	5.13696
	12	0.6	9.19229	9.1923
	6	0.5	4.73698	4.73699
	8.5	0.4	4.84137	4.84134
	5.5	0.5	4.4763	4.7369

Table (8) some of the test results of the fuzzy neural network (FN3) for the computer network (CN1)

	Inputs of FN3		Desired output	Output of FN3
	Link cost	Congestion prediction	Neighbor selection level	Neighbor selection level
	2	1.46316	7.0755	7.0755
	2.1	5.13696	18.3602	18.3602
	2	9.19229	35.1393	35.1393
	8	4.73698	28.8917	28.8917
	2.1	1.46316	7.4201	7.4206
	1.6	11.2359	41.8882	42.0732

Table (9) some of the test results of the fuzzy neural network (FN3) for the computer network (CN2)

Link cost	time	Inputs of FN3		Desired output	Output of FN3
		Congestion prediction	Neighbor selection level	Neighbor selection level	
3.352		1.46316	9.0804	9.0804	
6.312		5.13696	26.8333	26.8333	
12.624		9.19229	65.8843	65.8844	
3.352		4.73698	19.5252	19.5253	
3.152		10.4559	38.6157	38.6152	
6.312		8.367	34.5458	34.8242	

Table (10) the success rates of testing the proposed fuzzy neural networks for the computer network (CN1)

Fuzzy neural network	Success rate of test on trained sets	Success rate of test on other sets
FN1	100 %	92 %
FN2	100 %	94 %
FN3	100 %	95 %

Table (11) the success rates of testing the proposed fuzzy neural networks for the computer network (CN2)

Fuzzy neural network	Success rate of test on trained sets	Success rate of test on other sets
FN1	100 %	88 %
FN2	100 %	94 %
FN3	100 %	93 %

4. Conclusions

The proposed methods which used fuzzy neural networks to solve the routing problem of the computer networks are described in this paper. These fuzzy neural networks can be included as a part of a routing system that contains a monitor for providing them with necessary information. The congestion control with solving of routing problem is realized at one of these proposed methods as a combination of two fuzzy neural networks. For making the routing decision, system contains central monitor and the fuzzy neural network (FN1) at each node, to determine the best neighbor node depending on two primary criteria that are link time cost and queue time cost. The fuzzy neural networks (FN2, FN3) as congestion predictor and routing decision maker are included at each node to manage routing with congestion control. Results of the training and the testing of the fuzzy neural networks proof on their good performance for both computer networks (CN1, CN2).

References

- [1] S. Kumar and R. Miikkulainen, "Confidence_ Based Q_ Routing: An On_ Line Adaptive Network Routing Algorithm ", Proceedings of the Artificial Neural Networks in Engineering Conference, Texas University, Austin, 1998.
- [2] D. Davies, D. Barber, W. Price and C. Solomonides, "Computer Networks and Their Protocols ", J. Wiley and Sons Ltd, New York, 1979.
- [3] J. Malrand, "Communication Networks: A First Course ", R. D. Irwin and Akson Associates, Inc, 1991.
- [4] W. Newton, "A Neural Network Algorithm for Internetwork Routing ", Report in Software Engineering, for Degree of Bachelor, 2002.
- [5] G. Caro and M. Dorigo, "AntNet: A Mobile Agents Approach to Adaptive Routing ", Technical Report, Iridia University, No. 12, 1997.
- [6] J. Zue, S. Ng and L. Hanzo, "Fuzzy Logic Aided Dynamic Source Routing in Cross-Layer Operation Assisted Ad Hoc Networks ", Southampton University, United Kingdom, 2010.
- [7] L. Lersuwanakul, "Fuzzy Logic Based Routing in Grid Overlay Network ", Fern University, Hagen, 2010.
- [8] M. Dehyadegari and S. Mohammadi, "An Adaptive Fuzzy Logic-based Routing Algorithm for Networks-on-Chip ", NASA/ESA Conference on Adaptive Hardware and Systems, 2011.

- [9] A. Haboush, “ Fuzzy Optimized Metric for Adaptive Network Routing ”, International Journal of Computer Science and Security (IJCSS), Vol. 6, 2012.
- [10] A.Dana and N. Salehi, “ Congestion Aware Routing Algorithm for Mesh Network-on-Chip Platform ”, Indian Journal of Science and Technology, Vol. 5, No. 6, 2012.
- [11] M. Chelliah and B. Sivaselvan, “ Routing for Wireless Mesh Networks with Multiple Constraints Using Fuzzy Logic ”, International Arab Journal of Information Technology, Vol. 9, No. 1, 2012.
- [12] A.Tanenbaum, “ Computer Networks ”, Fourth Edition, Pearson Education, Inc, 2003.
- [13] S. Pithani and A. Sethi, “ A Fuzzy Set Delay Representation for Computer Network Routing Algorithms ”, Second International Symposium on Uncertainty Modeling and Analysis, College Park, MD, April, 1993.
- [14] R. Habil, “ Data Analysis with Neuro_Fuzzy Methods ”, Academic Report, Guericke University, Magdeburg, 25 Februar, 2000.
- [15] Y. Chen and C. Teng, “ Fuzzy Neural Network Systems in Model Reference Control Systems ”, Fuzzy logic and Expert Systems Applications, Academic Press, 1998.

ايجاد مسار شبكة حاسبات باستخدام الشبكات العصبية المضببة

خلود احمد نصار أ.د. تركي يونس عبد الله د. عبد الكريم يونس عبد الله
قسم علوم الحاسبات قسم هندسة الحاسبات قسم علوم الحاسبات
جامعة البصرة، البصرة ، العراق

المستخلص

الشبكات العصبية المضببة ممكن ان تستخدم بكفاءة كتقنية حل لمسألة ايجاد مسار في شبكات الحاسبات وذلك لغرض تحسين الانجازية. في هذا البحث، اقترحت طريقتين لحل هذه المسألة. تم استخدام الشبكات العصبية المضببة لصنع قرار ايجاد مسار، ومعالجة مشكلة الاكتظاظ مع حل مسألة ايجاد مسار. الطرق المقترحة طبقت لأمثلة نموذجية لشبكات الحاسبات. الشبكات العصبية المضببة تم تدريبها، ونتائج الفحص اكدت على انجازيتها العالية.

M OUT OF N ERROR CORRECTING CODE USING DIGITAL FILTERS

Rawnaq A. Habeeb

Electrical Engineering Department, College of Engineering, University of Basrah

ali.nasseir72@yahoo.com

Tel.No. 07704979696

Abstract:

In this work, forward and reversed digital filters are used with (m out of n) error correcting code in the form [xxxx000] i.e.(4 out of 7code). The hardware circuit is constructed using flip flops that constitute the digital filters. Practical results are applied for correcting a deliberately inserted error in the message.

1- Introduction

There are several known methods of detecting an error in digital message, namely: Redundancy, Echoplex, Exact Count Encoding, Parity, Checksum, Vertical-Horizontal Redundancy Checking, and finally Cyclic Redundancy Checking (CRC).

Most methods use codes which are "systematic": the transmitter sends a fixed number of original data bits, followed by fixed number of check bits (usually referred to as redundancy) which are derived from the data bits by some deterministic algorithm. The receiver applies the same algorithm to the received data bits and compares its output to the received check bits; if the values do not

match, an error has occurred at some point during the transmission. A block code converts affixed length of K data bits to a codeword of fixed length N, where $N > K$. The block of encoded message consists of K message bits in addition to N-K error check bits. The rate of the code is the ratio N/K , and the redundancy of the code is $(1-K/N)^{[1-5]}$.

In this study a correcting code of seven bits is used. The message constitutes the most significant four bits while the rest are zeroes (the added redundancy bits), so the codeword is in the form [xxxx000].

In case of error correction few methods are used namely: Symbol Substitution,

Retransmission, and Forward Error Correction (FEC). Nowadays the plausible method FEC is versatile and applied almost universally. It involves applications of Reed-Soloman coding as indicated in the block diagram of Fig.1, where the first three blocks are common parts for cables, satellites, and terrestrials; while, the last block is used for satellites, and terrestrials only.

In Fig.1, the numbers (204,188,8) stands for Transport Packets of 188 bytes long, it uses 16 parity bytes, when added to 188 yields 204. The 8 means that this method can correct up to 8 bytes of errors. The energy dispersal uses Pseudo Random Binary Sequence (PRBS) of 1503 bytes. Forney interleaving is vital in the sense that it increases the efficiency of Reed-Soloman coding by spreading the burst error in the channel. Convolutional coding (inner coding) is the complement of RS and forney interleaving because it corrects other kinds of errors. It lowers redundancies by the method of puncturing the output of the convolutional encoder.

The technique used in this research is another method of correcting an erroneous byte in a message by the method of "search and compare" the system searches for the correct code in the sense of scanning and then compares the scanned result to a standard (xxxx000) byte. If the scanned result matches this standard code, then the byte involved is the correct message byte.

A hypothetical communication system with artificial erroneous message is considered and a correcting circuit is designed so as to correct the received message.

2- Theoretical Circuit:

2.1- Block Diagram of the system

The theoretical circuit can be represented as shown in Fig.2. The message (x) is passed through a forward digital filter of transfer function $f(D)$ at the sending end. During the path of the message towards the receiving end, an error is deliberately inserted in the message using shift and add method (Pseudo Noise [PN] generator)^[6,7]. At the receiving end, a reverse digital filter of transfer function $\frac{1}{f(D)}$ is used in addition to m out of n facility to detect the error and correct it simultaneously.

2.2-Related Algebra of Digital Filters

The forward digital filter at the sending end is shown at Fig.3 for 3-bits.

Assuming the input and output are (x) and (y) respectively, and the delay "by one bit period" to be (D), then^[8]

$$y = x \oplus Dx \oplus D^3x \text{-----}(1)$$

from which we get

$$f(D) = \frac{y}{x} = 1 \oplus D \oplus D^3 \text{-----}(2)$$

Fig.4 shows the reverse digital filter at the receiving end, from which

$$y' = x' \oplus D^3y' \oplus Dy' \text{-----}(3)$$

Using transposition property, the following result is obtained:

$$x' = y' \oplus Dy' \oplus D^3y' \text{-----}(4)$$

Therefore:

$$\frac{1}{f(D)} = \frac{y'}{x'} = \frac{1}{1 \oplus D \oplus D^3} \text{ -----(5)}$$

where (x') and (y') are the input and output respectively and (D) is the delay by one bit period.

3- Circuit Analysis

Fig.5 shows the practical circuit of the detector and corrector system (the scanner) at the receiving end. It is consisted of three registers. The received message is stored in the second register, while the third register is the storage of the corrected message.

The first stage of the system (3-bit shift register) generates the codes due to probability of errors as follows:

Denoting the output of the first register by "z"

i. If the first register of Fig.5 is loaded initially with the logic [100], and the register is drove by clock pulses, the output "z" becomes as in Table_1 .

Table_1: The status of initially loaded shift register of Fig.5 driven by clock pulses

After n th clock pulse	Z	Contents of shift register
Initially	1	100
1	1	110
2	1	111
3	0	011
4	1	101
5	0	010
6	0	001
7	1	100

It is clear from tabl_1 that the final state is the same as the first one, the cycle is then repeats in such a way that "z" (the output) will have the following recurrence cycles

.....1110100 1110100 1110100.....

ii. Consider the register loaded initially with the logic [110].

If the work in (i) above is repeated, the possible outputs of (z) become

....0111010 0111010 0111010....

iii. If the register is loaded with logic [110] then the possible outputs of (z) are

z = 0011101, and so on.

The above functions of [z] are termed ϕE , for which it is clear that the result is shifted one bit to the right each time, so initially the shift register can be loaded by any code, hence the possible outputs of [z] are tabulated as follows:

Table_2 Possible outputs of the first register of Fig.5

Function of [z]	Binary value of [z]
ϕE_1	1110100
ϕE_2	0111010
ϕE_3	0011101
ϕE_4	1001110
ϕE_5	0100111
ϕE_6	1010011
ϕE_7	1101001

Referring to Fig.2; $x' = y \oplus E$ represents an input to the reverse filter denoted by $\frac{1}{f(D)}$ where E is an error bit inserted to the transmitted message for example the third bit i.e. $E = E_3$. So the output y' of this feedback filter is given by

$$y' = \phi(x') \text{-----}(6)$$

$$= \phi[y \oplus E_3]$$

$$= \phi[y] \oplus \phi[E_3]$$

$$= \phi f(x) \oplus \phi[E_3]$$

$$= \phi[D] \oplus x'$$

$$= \frac{1}{f(D)} \oplus x' \text{-----}(7)$$

y' can also be expressed as

$$y' = x \oplus \phi[E_3] \text{-----}(8)$$

where x is the input.

From the final results (equations 7&8), it is noticed that the input x is the original message while $\phi[E_3]$ is the response of the reverse filter to the impulsive error (0010000), hence the output x is expressed as

$$x = y' \oplus \phi[E_3] \text{-----}(9)$$

practically the position of the error is not known, hence all of the responses should be examined, namely $\phi[E_1]$, $\phi[E_2]$, $\phi[E_3]$, ..., $\phi[E_7]$, (as in Fig.6) which correspond to impulse function that may be at the 1st, 2nd, 3rd, ..., 7th position respectively. Therefore, the general form of equation 9 will be

$$x = y' \oplus \phi[E_i] \text{-----}(10)$$

Where $\phi[E_i]$ is the output generated from the first register of the system.

After achieving all examinations, the XoR operation (according to equation 10) continues until a code with three zeroes at the right hand side (in the form xxxx000) is obtained in the

third register, where the correct message is xxxx.

4-Practical Results

The electronic circuit of the detector and corrector system (the scanner) as shown in Figs.2 and 6 has been built practically. An erroneous message $y'=0011011$ was chosen as an example which is supposed to be belonging to the seven bit code [xxxx000], then y' is examined according to equation (10), by considering all possible ϕE_i (from Table_2) until a code of the form xxxx000 is reached as in Table_3:

Table_3: TheXoR results for correcting the erroneous message.

i	ϕE_i	$y' \oplus \phi E_i$	Result
1	1110100	1101111	Incorrect
2	0111010	0100001	Incorrect
3	0011101	0000110	Incorrect
4	1001110	1010101	Incorrect
5	0100111	0111100	Incorrect
6	1010011	1001000	Correct
7	1101001	1110010	Incorrect

Hence, the error is at the 6th position of the message and the message is corrected.

5-Conclusions:

This research work claims juniority and originality, whereas in the existed literature the standard method of error detection and correction (as mentioned above), the plausible of which is the FEC that utilizes either Hamming code or Reed-Soloman code, none of such methods includes "SCAN and COMPARE" method as in this work.

A standard (N,K) code, in this research, i.e. (xxxx000 code) was applied as a stem or a reference to all received messages. Such received messages are either correct or faulty. If erroneous message is received it is corrected so as to match the reference (xxxx000) code..

A final word can be added that to comprehend the literature of this research the reader may concentrate on sequential technique given in the tables throughout this work, the picture then will be elaborated and the ambiguousness will be cleared, consequently the practicality of the circuit will be realized and hence anticipated.

This piece of work can be an integral part of a system that involves digital communication based on this (N,K) code, or else may be developed to further usage and extensions yielding same trend or same principles.

6- References:

1. I.B.Djordjevic, and B.Vasic, "Communication Systems", IEEE Photonics Technology Letters, Vol.17, No.10, October 2005, pp.2224_2226.
2. P.Yun, G.Ning, D.Zaiwang, "CRC Look-up Table Optimization for Single-Bit Correction", TSINGHUA SCIENCE AND TECHNOLOGY, Vol.12, No.5, Oct. 2007, pp.620_632.
3. G.Castagnoli, S.Bräuer, and M.Herrmann, "Optimization of Cyclic Redundancy-Check Codes With 24 and 32 Parity Bits", IEEE Transaction on Communications. Vol.41, No.6, Jun. 1993, pp.883_892.
4. C.Heegard, Fellow, and A.J.King, "FIR Parity Check Codes", IEEE Transactions on Communications, July 2000, Vol.48, No.7, pp.1108_1113.
5. A.V.N.Rao, K.S.Rajan, and K.S.Rao, "Integrating Error Detection With DATA Encryption Algorithm Using Permutation Invariant RAO ALAKA Shift Transform", ARPN Journal of Engineering and Applied Sciences, Vol.3, No.5, October, 2008, pp.59_67.
6. R.N.Mutagi, "Pseudo Noise Sequences for Engineers", Electronics and Communication Engineering Journal, April, 1996, pp.79_87.
7. M.M.Islam, M.M.Hussain, M.S.Rahman, and M.M.Rahman, "Performance Evaluation of Pseudo_Random Noise and Hadamard_Walish Sequences for Hamming Coded WCDMA System Over AWGN and FADING Channels", Banglادish Research Publications Journal, Vol.3, Issue:4, May, 1997, pp.1122_1128.
8. W.J.Hurd, "Efficient Generation of Statistically Good Pseudo-Noise by Linearly Interconnected Shift Registers", JPL technical Report 32_1526, Vol.XI, April, 1998, pp.92_103.

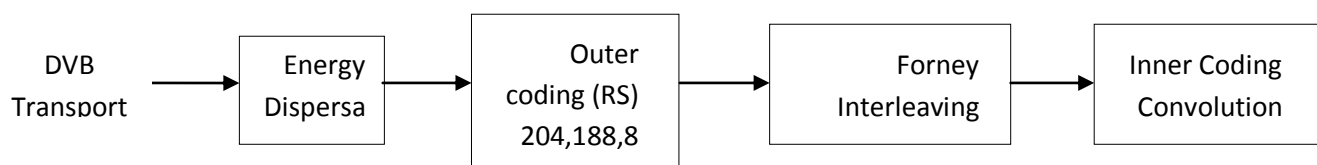


Fig.1- Block diagram of FEC system

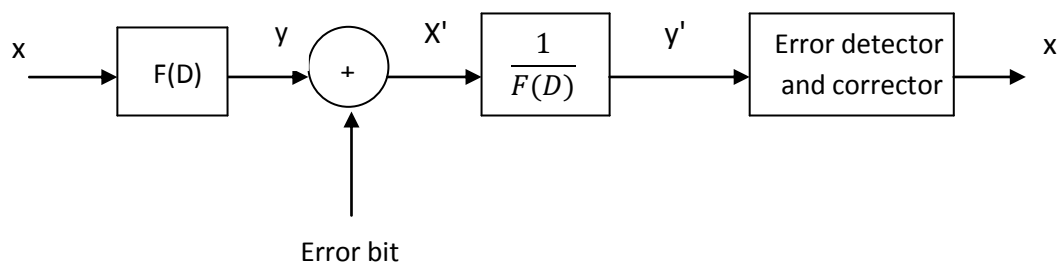


Fig.2_Theoretical circuit

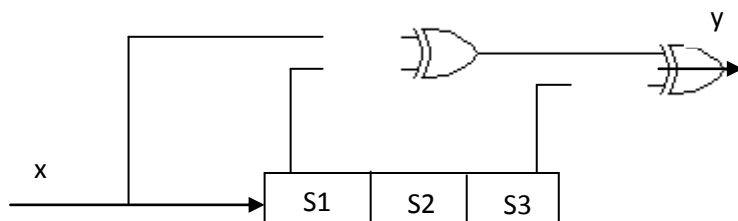


Fig.3_ Forward digital filter

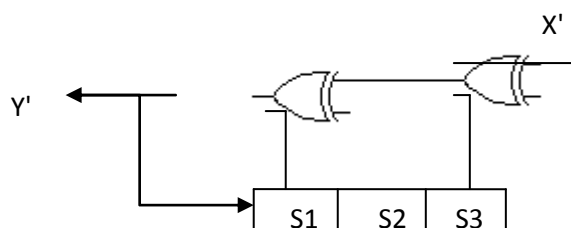


Fig.4_Reversed digital filter

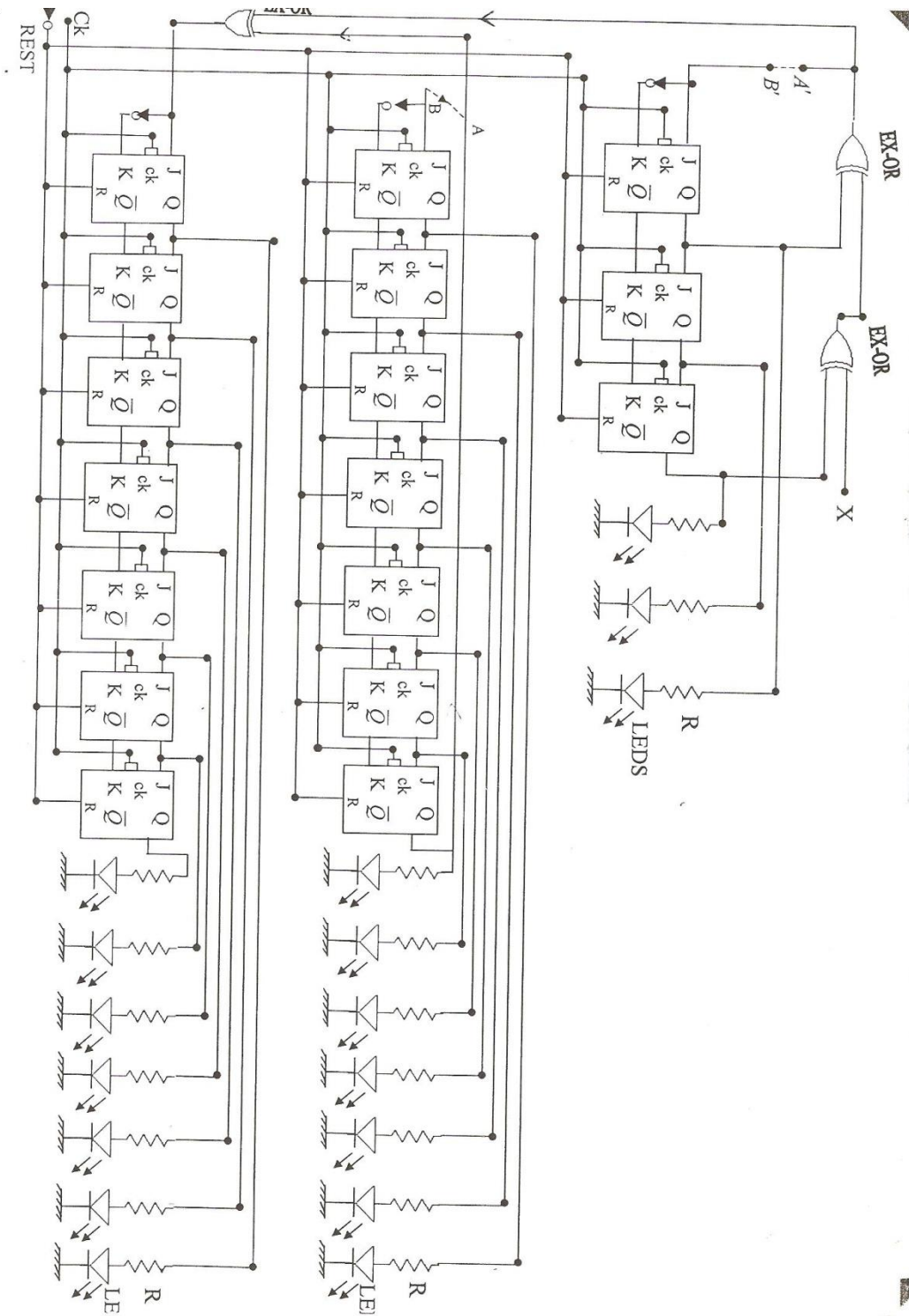


Fig .5 Practical circuit

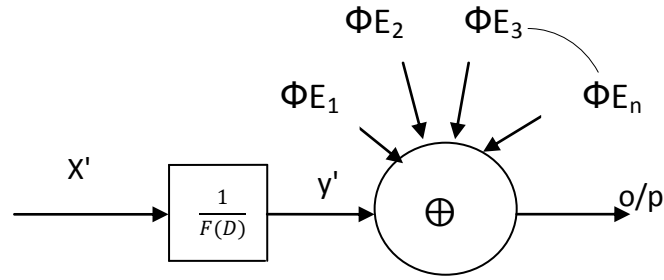


Fig.6 Theoretical circuit of the receiving end

شفرة m من أصل n لتصحيح الخطأ باستخدام المرشحات الرقمية

رونق علي حبيب

قسم الهندسة الكهربائية، كلية الهندسة، جامعة البصرة

ali.nasseir72@yahoo.com

جوال: 07704979696

المستخلص

يتعلق العمل باستخدام مرشحات رقمية أمامية وعكسية مع شفرة (m out of n) لتصحيح الخطأ والتي تكون بالصيغة (xxxx000)، (أي شفرة 4 من أصل 7) حيث يتم استخدام نطاقات لتمثيل المرشحات الرقمية في بناء الدائرة. ولقد تم عرض النتائج العملية لتصحيح خطأ محقق في إشارة المعلومات المرسله عمدا.

Currency exchange rate prediction using wavelet network

Adalla Mehdi

Department of computer science - College of science - University of Basrah

Abstract

Currency exchange rates prediction is one of the most important applications of modern time series prediction. The currency rates are inherently noisy, and chaotic. There is no complete information that could be obtained from the history of the past behavior of currency exchange rate markets to fully capture the dependency between the future exchange rates and that of the past. In this paper, the currency exchange rate prediction problem is studied and a new proposed currency exchange rate prediction scheme based on using wavelet network is presented. The proposed scheme is tested and results are compared with other known methods. Different cases are considered.

1- Introduction

Exchange rates prediction is one of the challenging applications of modern time series prediction. The rates are inherently noisy, non-stationary and chaotic [1, 2]. These characteristics suggest that there is no complete information that could be obtained from the past behavior of such markets to fully capture the dependency between the future exchange rates and that of the past. One general assumption is made in such cases is that the historical data incorporate all those behavior. As a result, the historical data is the major player in the prediction process. Although the well-known conventional

prediction techniques provide predictions, for many stable forecasting systems, of acceptable quality, these techniques seem inappropriate for non-stationary and chaotic system such as currency exchange rates, interest rates and share prices. The purpose of this paper is to investigate the use of wavelet networks based techniques for prediction of currency exchange rate. Auto-Regressive Integrated Moving Average (ARIMA) technique [3,4] has been widely used for time series prediction. However, ARIMA is developed based on the assumption that the time series being forecasted are linear and stationary [4]. The Artificial Neural

Networks, have been used for prediction and system modeling, with great applicability in time-series analysis and prediction [5-8].

In this paper, we develop a method based on using wavelet network for predicting currency exchange rates. Wavelet neural networks combine the theory of wavelets and neural networks into one. A wavelet neural network generally consists of a feed-forward neural network, with one hidden layer, whose activation functions are drawn from an orthonormal wavelet family.

One applications of wavelet neural networks is that of function estimation. Given a series of observed values of a function, a

wavelet network can be trained to learn the composition of that function, and hence calculate an expected value for a given input.

2- Wavelet Network

The structure of a wavelet neural network is very similar to that of a feed-forward neural network, taking one or more inputs, with one hidden layer and whose output layer consists of one or more linear combiners or summers (see Figure 1). The hidden layer consists of neurons, whose activation functions are drawn from a wavelet basis. These wavelet neurons are usually referred to as wavelons.

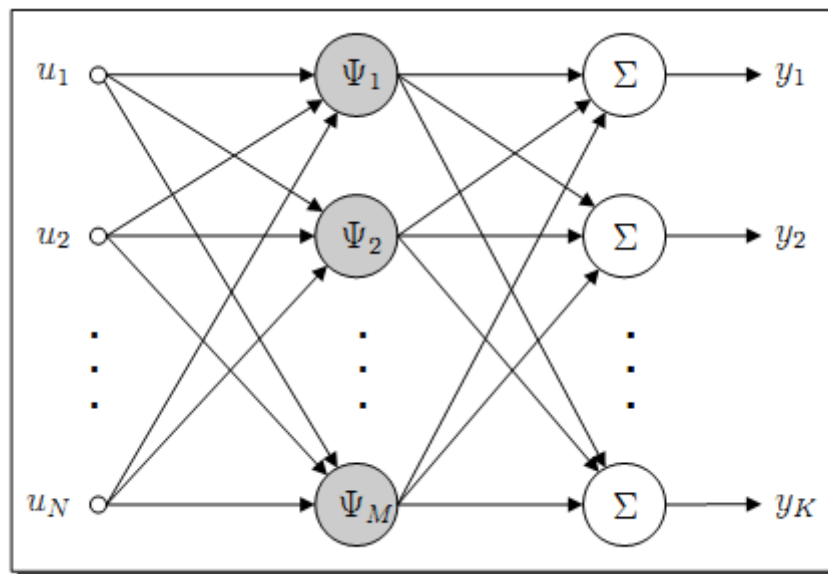


Figure (1): Structure of a Wavelet Neural Network

There are two main approaches to creating wavelet neural networks. In the first the wavelet and the neural network processing are performed separately. The input signal is first decomposed using some wavelet basis by the

neurons in the hidden layer. The wavelet coefficients are then output to one or more summers whose input weights are modified in accordance with some learning algorithm. The second type combines the two theories.

In this case the translation and dilation of the wavelets along with the weights are modified in accordance with some learning algorithm. In general, the first approach is known as a wave net while the second type is known as a wavelet network[8]. The architecture of a

single input single output wavelet network is shown in Figure 2. The hidden layer consists of M wavelons. The output neuron is a summer. It output a weighted sum of the wavelon

$$y(u) = \sum_{i=1}^M w_i \psi_{\lambda_i, t_i}(u) + \bar{y} \quad \dots (1)$$

The output of a wavelon is given by:

$$\psi_{\lambda, t}(u) = \psi\left(\frac{u - t}{\lambda}\right)$$

The addition of the bias $\bar{y}$ is for functions whose mean is non zero (since the

wavelet function $\psi(u)$ is zero mean). In a wavelet network all parameters $\bar{y}$, w_i , t_i and λ_i are adjustable by some learning procedure .

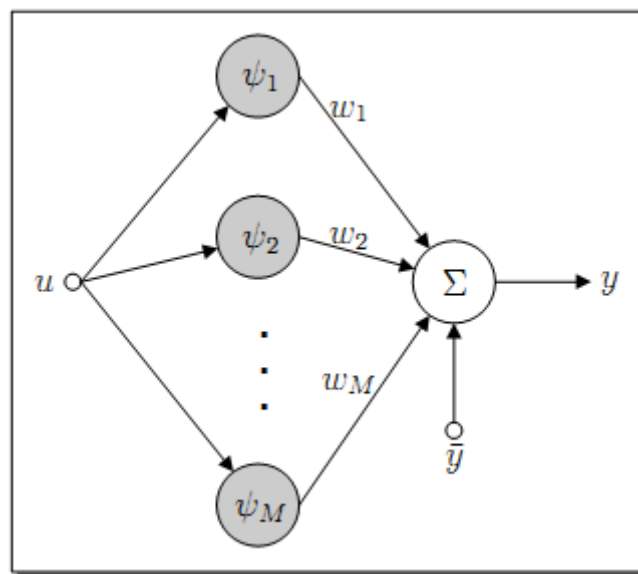


Figure (2): Wavelet Neural Network

3- Multidimensional Wavelet Network

The input in this case is a multidimensional vector and the wavelons consist of multidimensional wavelet activation functions. They will produce a non - zero output when the input vector lies within a

small area of the multidimensional input space. The output of the wavelet network is one or more linear combinations of these multidimensional wavelets. Figure 3 shows the form of a wavelon . The output is expressed as:

$$y_j = \sum_{i=1}^M w_{i,j} \psi_i(u_1, \dots, u_N) + \bar{y}_j \text{ for } j = 1, \dots, K \quad \dots (2)$$

where $\bar{y}_j$ is needed to deal with functions of non zero mean

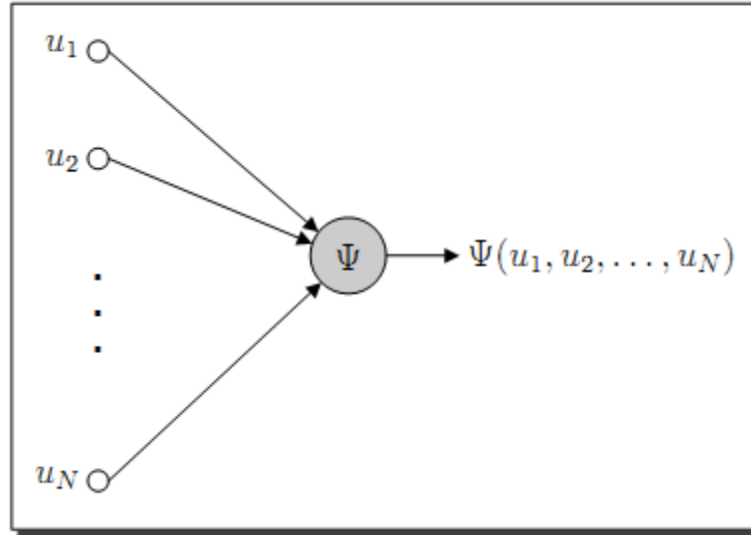


Figure (3): A Wavelet Neuron with a Multidimensional Wavelet Activation Function

The input - output mapping of the network is defined as:

$$y(u) = \sum_{i=1}^M w_i \Psi_i(u) + \bar{y} \quad \text{where} \quad \begin{cases} y = (y_1, \dots, y_K) \\ w_i = (w_{i1}, \dots, w_{iK}) \\ u = (u_1, \dots, u_N) \\ \bar{y} = (\bar{y}_1, \dots, \bar{y}_K) \end{cases}$$

One application of wavelet networks is function approximation. In [8] an algorithm for adjusting the network parameters for the one-dimensional case. Learning is performed from a random sample of observed input - output pairs

using of gradient type algorithm for the learning.

The parameters $\bar{y}$, w_i 's, t_i 's and λ_i 's should be formed into one vector θ . Now $y_\theta(u)$ refers to the wavelet network, expressed in (4) with parameter vector θ .

$$y_{\theta}(\mathbf{u}) = \sum w_i \psi\left(\frac{\mathbf{u} - \mathbf{t}_i}{\lambda_i}\right) + \bar{y} \quad \dots \quad (3)$$

The objective function to be minimized is then

$$C(\theta) = \frac{1}{2} E\{(y_{\theta}(u) - f(u))^2\}.$$

The minimization is performed using a gradient algorithm. This recursively modifies θ , after each sample pair $\{u_k, f(u_k)\}$, in the opposite direction of the gradient of

$$c(\theta, u_k, f(u_k)) = \frac{1}{2} (y_{\theta}(u_k) - f(u_k))^2.$$

The gradient for each parameter of θ can be found by calculating the partial derivatives of $c(\theta, u_k, f(u_k))$ as follows:

$$\begin{aligned} \frac{\partial c}{\partial \bar{y}} &= e_k \\ \frac{\partial c}{\partial w_i} &= e_k \psi(z_{ki}) \\ \frac{\partial c}{\partial t_i} &= -e_k w_i \frac{1}{\lambda_i} \psi'(z_{ki}) \\ \frac{\partial c}{\partial \lambda_i} &= -e_k w_i \left(\frac{u_k - t_i}{\lambda_i^2} \right) \psi'(z_{ki}) \end{aligned}$$

$$\text{where } e_k = y_{\theta}(u_k) - f(u_k), z_{ki} = \frac{u_k - t_i}{\lambda_i} \text{ and } \psi'(z) = \frac{d\psi(z)}{dz}.$$

To implement this algorithm, a learning rate value and the number of learning iterations need to be chosen. The learning rate $\gamma \in (0,1]$ determines how fast the algorithm attempts to converge. The gradients for each parameter are multiplied by γ before being used to modify that parameter. The learning iterations determine how many times the

training data should be fed through the learning process.

4- proposed scheme

Time series analysis is used as a method for currency exchange rate prediction. A time series $x(1), \dots, x(k)$ are used to predict the

value $x(k+1)$. The inputs to the network are chosen as the previous k values $x(1), x(2), \dots, x(k)$ and the output will be the predicted value $x(k+1)$.

In this section we present the proposed wavelet network scheme for solving the

prediction problem. The network has k input unit and one output as shown in figure(4). The k past currency rate values are used as input to the network to predict the next (the $k+1$) currency rate value.

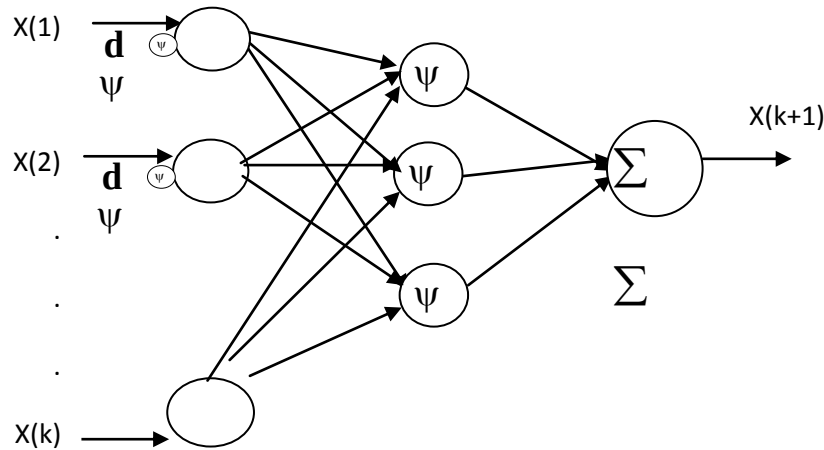


Figure (4): proposed wavelet network

5- Simulation result

The data set contains weekly averaged exchange rates between two main currencies - the British pound and US dollar – in the period from 31 December 1979 to 26 December 1983. The first 110 data points are considered [9]. The data are normalized to values in the range between 0 and 1 using the formula. The proposed wavelet network scheme shown in Figure (4) is used for currency exchange rate prediction. two cases are considered.

- 1- A wavelet network with three input($k=3$) and one output as shown

in figure(4). It has three input units (each time three consecutive exchange rates are used as inputs to the network to produce a prediction for the fourth exchange rate).

- 2- A wavelet network with four input($k=4$) and one output as shown in figure(4). It has four input neurons (each time four consecutive exchange rates are used as inputs to the network to produce a prediction for the fifth exchange rate).

The data set is divided into two groups: training group and testing group. The gradient algorithm is used as a learning algorithm. For

the 110 data items, the first 100 data are used as training group. The remaining 10 data are used for testing the trained network to unseen data.

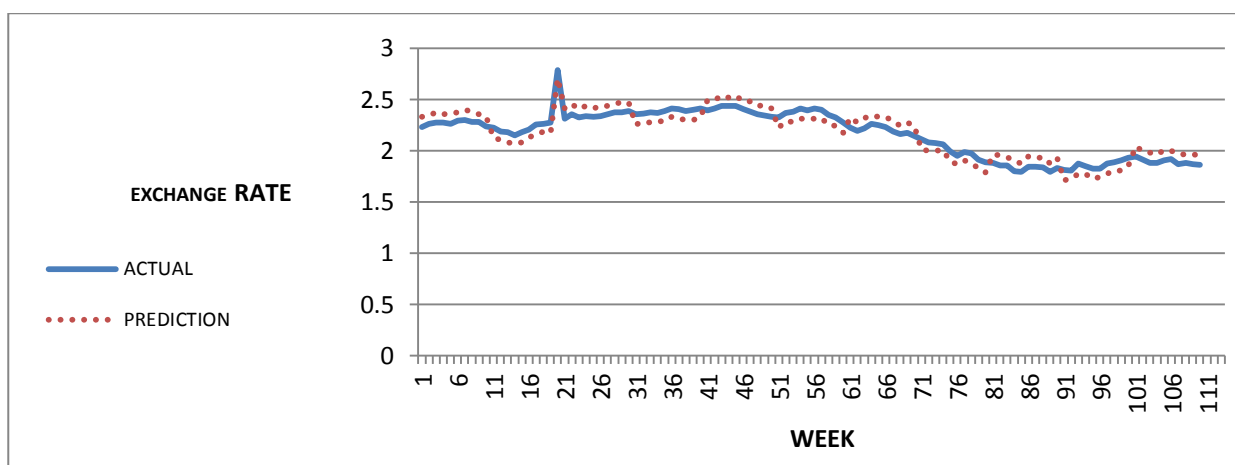
The wavelet network in both cases are trained for 500 epochs with 0.4 learning rate. Three Rasp2 wavelet functions are used ($f(t) = \frac{\tau \cos(\tau)}{\tau^2 + 1}$ where $\tau = \frac{u-t}{\lambda}$). The trained network is tested with the overall 110 data. The results of the testing(the prediction) are given in Figure (5) and Figure (6) for the two cases respectively.

To compare the performance with that of using neural network the multilayer

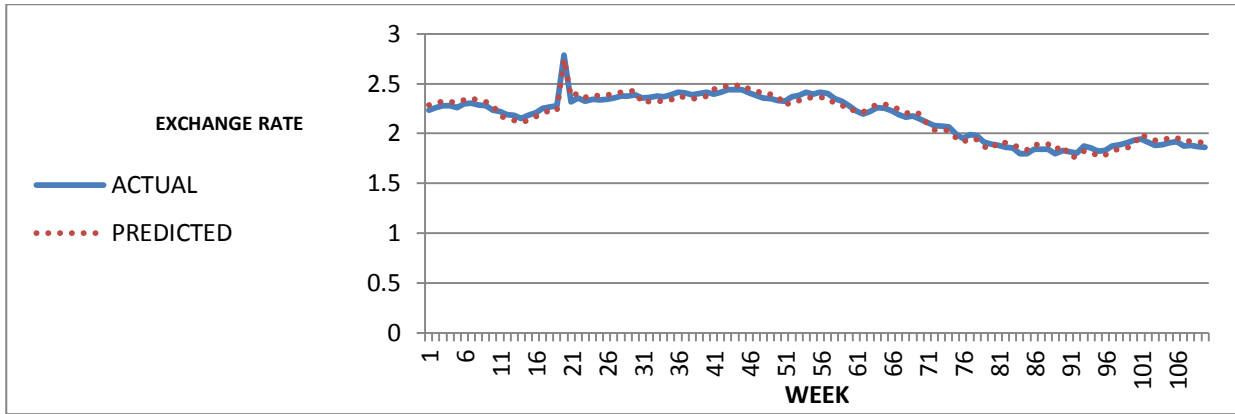
perceptron network is also used for prediction .The neural network is used in two different cases . In the first case three input network is used while in the second case four input network is used. The neural network in both cases are trained for 500 epochs with 0.4 learning rate .The activation function is a sigmoid function. The results of testing(the prediction) for the cases of three input and four input are shown in Figure (7). and Figure (8) respectively. The mean square errors in testing for the four cases is shown in table(1).

Table(1)Mean square error in testing.

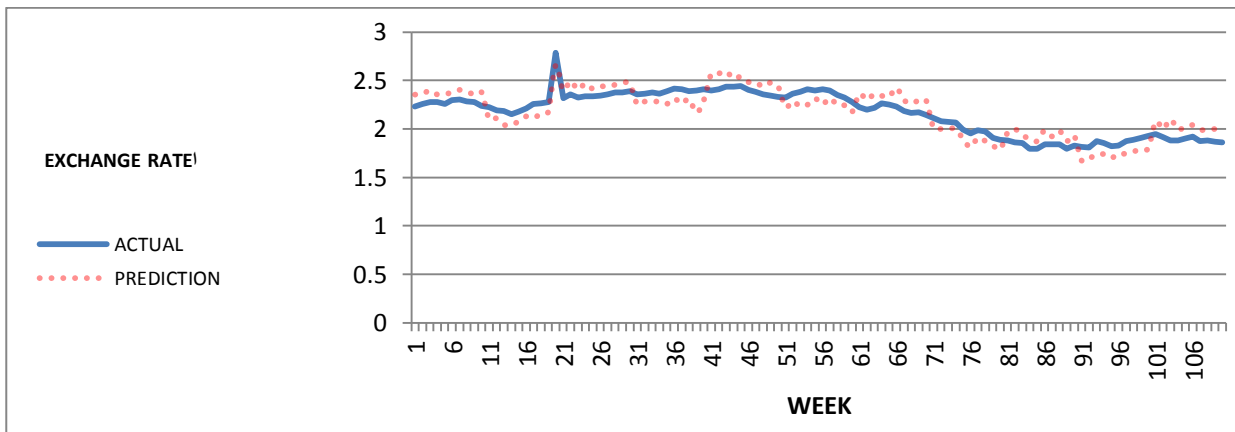
Wavelet Network (3-Input)	Wavelet Network (4-Input)	Neural Network (3-Input)	Neural Network (4-Input)
0.0054	0.0012	0.0145	0.0043



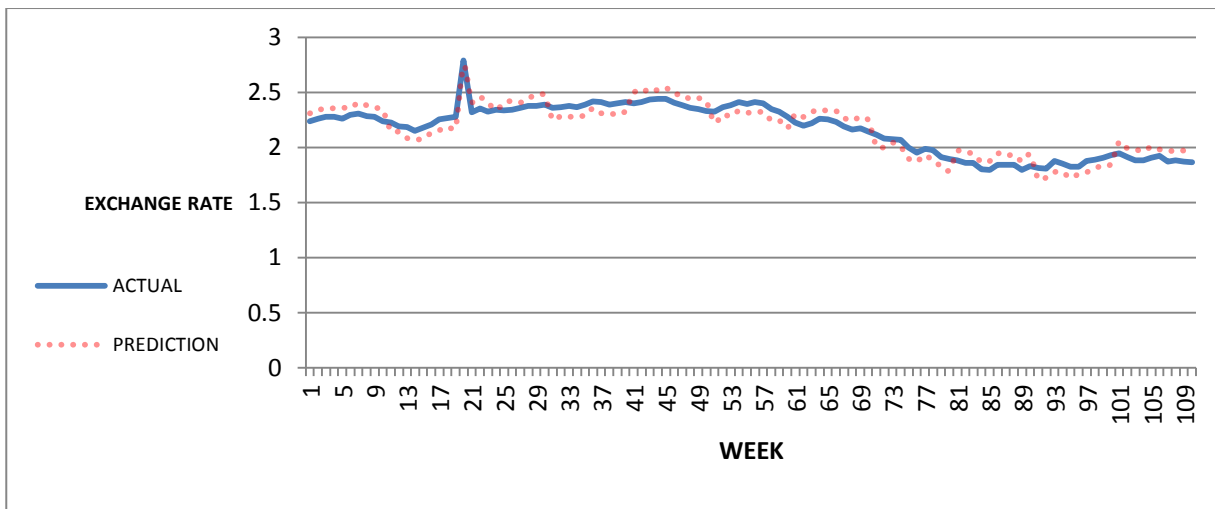
Figure(5): Exchange rate prediction using wavelet network [Three Input]



Figure(6): Exchange rate prediction using wavelet network [Four Input]



Figure(7): Exchange rate prediction using Neural network [Three Input]



Figure(8): Exchange rate prediction using Neural network [Four Input]

6- Conclusion

Wavelet network prediction scheme is presented and used for currency exchange rate prediction. The method is applied for

data set contains weekly averaged exchange rates between two currencies - the British pound and US dollar. Simulation results show good performance for the proposed wavelet

network scheme. Results of using wave let network is better than that of using neural network.

References

- [1] G. Deboeck, Trading on the Edge: Neural, Genetic and Fuzzy Systems for Chaotic Financial Markets, New York Wiley, 1994.
- [2] S. Yaser and A. Atiya, "Introduction to Financial Forecasting," Applied Intelligence, vol. 6, pp 205-213, 1996.
- [3] Y. Chena, , Bo Yanga,b, and J. Donga, "Time-series prediction using a local linear wavelet neural network", Neurocomputing ,vol. 69 , pp 449 – 465 ,2006.
- [4] G. E. P. Box and G. M. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, San Francosco, CA, 1990.
- [5] L. Cao and F. Tay, "Financial forecasting using support vector machines," Neural Comput. & Applic, vol. 10, pp.184-192, 2001.
- [6] J. Yao, Y. Li and C. L. Tan, "Option price forecasting using neural networks," OMEGA: Int. Journal of Management Science, vol. 28, pp 455-466, 2000.
- [7] J. Yao and C.L. Tan, "A case study on using neural networks to perform technical forecasting of forex," ,Neurocomputing, vol. 34, pp. 79-98, 2000.
- [8] Hu Tao," A Wavelet Neural Network Model for Forecasting Exchange Rate Integrated with Genetic Algorithm", International Journal of Computer Science and Network Security, VOL.6 No.8A, PP. 60-63 , 2006.
- [9] G. Zhang and M. Y. Hu, "Neural network forecasting of the British Pound/US dollar exchange rate," OMEGA: Int. Journal of Management Science, vol. 26, pp. 495-506, 1998.
- [10] D avi d Vei tch, Wavelet Neural Networks and their application in the study of dynamic systems, MSc thesis , University of York, UK, 2005.
- [11] D.T.Phamand L. Xing,"Neural network for Identification , prediction and control, Springer 1995.

التنبؤ بأسعار صرف العملات باستخدام شبكة تحويلية المويجة

عدالة مهدي جيا

قسم علوم الحاسبات - كلية العلوم - جامعة البصرة

الملخص:

يعتبر التنبؤ بأسعار صرف العملات من التطبيقات المهمة لموضوع التنبؤ بالسلاسل الزمنية الحديثة. ان الاسعار في جوهرها غير واضحة وعشوائية. ولا توجد معلومات كاملة يمكن الحصول عليها من السنوات السابقة لأسعار الصرف في اسواق تبادل العملات لوضع علاقة بين أسعار الصرف المستقبلية والسابقة لها. تم في هذا البحث دراسة مسألة التنبؤ بأسعار صرف العملات واقتراح اسلوب للتنبؤ بأسعار صرف العملات باستخدام شبكة تحويلية المويجة. تم اختبار الطريقة المقترحة ومقارنة النتائج مع الطرق الاخرى.

Satellite image enhancement using histogram hyperbolization

Hussain Zaydan Ali

Expert / Image processing

Ministry of Science and Technology

E-mail: hussainzayali53@yahoo.com

Mobile : 07901726448

Abstract

One of the most important quality factors in satellite images comes from its contrast. Contrast enhancement is frequently referred to as one of the most important issues in image processing. Image histogram is a powerful engineering tool to portray information of an image. In this paper a Landsat ETM+ image was analyzed to find out the power of histogram hyperbolization in detecting and delimiting different features in the scene. A visual basic program is designed and implemented on the spectral bands of the satellite image. The results were combined in color composites using ERDAS Imagine Ver. 9.2. Most conventional contrast enhancement algorithms usually fail to provide detailed contrast information in the dark and bright areas of remotely sensed images. The test results indicate that the proposed method provide better contrast image than the conventional enhancement methods.

I. Introduction

Image enhancement is the improvement of digital image quality (wanted e.g. for visual inspection or for machine analysis), without knowledge about the source of degradation. If the source of degradation is known, one calls the process image restoration[1]. Image Enhancement techniques are instigated for making satellite imageries more informative and helping to achieve the goal of image interpretation. The term enhancement is used to

mean the alteration of the appearance of an image in such a way that the information contained in that image is more readily interpreted visually in terms of a particular need. The image enhancement techniques are applied either to single-band images or separately to the individual bands of a multiband image set. This operation seeks to improve the appearance of image data as a way of assisting in visual interpretation and analysis. Image enhancement techniques improve the

subjective quality of the picture, by intensifying image characteristics. The goal is to emphasize certain image features for analysis, image display, or to increase their usefulness. Examples include contrast and edge enhancement, noise filtering, and sharpening[2]. Image enhancement seeks to make the image look better and be more easily interpreted, and can either be radiometric or geometric. Radiometric enhancement is designed to improve the contrast and colour balance in an image while geometric enhancement seeks to make spatial features such as edges more apparent[3]. Radiometric enhancements include contrast stretching, histogram matching, band ratio, band arithmetic and principal component analysis. Radiometric enhancement makes extensive use of the image histograms so it is important to understand what the image histogram is and how it is used. An image histogram is a bar graph of DN's against pixel frequency (number of times that particular grey scale value occurs in the image) is termed an image histogram. The histogram represents a probability distribution function that tells us the chance of finding a pixel with a certain DN's in an image[1]. The histogram has many uses in radiometric correction but can also give clues as to the nature of the image we are analyzing. Each image band has its own unique histogram although the converse is not necessarily true. Histograms do not give any indication of the

spatial distribution of the DN's but merely give an indication of their spectral distribution. Image histograms are very useful for determining the spectral distribution of the DN's in an image.

The principal objective of enhancement techniques is to process the image so that the result is more suitable than its original[1]. Image enhancement are algorithms which were applied to remotely sensed data to improve the appearance of an image for human visual analysis, or occasionally for subsequent machine analysis[4]. Image enhancement is useful in feature extraction, image analysis, and visual information. The enhancement process itself does not increase the inherent information content in the data but it simply emphasizes certain specified image characteristics. Enhancement algorithms are, generally, interactive and application-dependent[5]. Image enhancement techniques, such as contrast stretching which maps each gray level into another gray level by a predetermined transformation. An example is the histogram equalization method, in which the input gray levels are mapped so that the output gray level distribution is uniform. This method has been found to be a powerful technique for improving low contrast images[6]. Other enhancement techniques perform local neighborhood operations as in convolution, transform operation as in the discrete fourier

transform, and other operation as in pseudo-coloring where a gray – level image is mapped into a color image by assigning different colors to different features[7].

II. The Histogram Hyperbolization

It consists of a memoryless nonlinear, space-invariant transformation of the pixel brightness values. This transformation is based upon the histogram of the picture to be processed and the nature of human brightness perception. Contrast is generally defined as the relative difference between light and dark areas

$$p(J) = \frac{d}{dJ} \int_{I_{min}}^{I(J)} p(I) d(I) \quad (1)$$

where

$I(J)$ is inverse of $J(I)$.

The above equation represents the continuous case, which is discussed first for simplicity. Also and without loss of generality, it is assumed that I is normalized to the range $0 \leq I \leq 1$.

The basic idea underlying histogram hyperbolization is to seek a function $J(I)$ such that all perceived brightness levels of the processed image are equiprobable because the perception of brightness is a nonlinear function of light intensity, this goal is not achieved by histogram equalization. In order to specify an appropriate transformation, we need an expression for the perception of brightness for which about half a dozen formulas have been

of a picture because the visibility by such relative differences, it is often possible to enhance the intelligibility of pictures by an appropriate transformation $J(I)$ of the pixel brightness values $I(X,Y)$. This is especially true when images occupy a small portion of the available dynamic range. when the pixel brightness values of an image are subjected to a memory less, space – invariant transformation $J(I)$, the empirical probability density function $P(I)$ is modified as follows[8]:

proposed . The difficulty is that brightness perception depends upon viewing conditions such as target size, background illumination, etc. Which are unpredictable in pictures, and has been arbitrarily chosen here to express the perceived brightness B as:

$$B = \log(J + c)$$

And the constant c is the adjusted experimentally (normalization is not necessary here).

Letting

$$B_{\max} = \log(1 + c)$$

$$B_{\min} = \log c_2$$

And requiring that B be equally distributed,

$$p(B) = \frac{1}{B_{\max} - B_{\min}}$$

To find that the desired density function

$$p(J) = \frac{d}{dJ} \int_{B_{\min}}^{B(J)} \frac{1}{B_{\max} - B_{\min}} dB = \frac{1}{(J + c) \log(1 + 1/c)} \quad (2)$$

Has a hyperbolic form. Because the effect of the parameter c is some

what obscure a more convenient parameter is defined, namely, the slope s of $P(J)$ at the origin $J = 0$

$$s = \left. \frac{d}{dJ} \right|_{J=0} = \frac{-1}{e^2 \log(1 + 1/c)^2} \quad (3)$$

The desired function $J(I)$ can now be determined by observing that (see equations (1) and (2)).

$$p(J) = \frac{1}{(J + c) \log(1 + 1/c)} = \frac{d}{dJ} \int_0^{I(J)} p(I) d(I) \quad (4)$$

Integrating both sides with respect to J , to obtain

$$I(J) = F_{I^{-1}} \frac{\log(1 + J/c)}{\log(1 + 1/c)} \quad (5)$$

where F^{-1} is the inverse of the cumulative distribution

$$F_I = \int_0^I p(I) d(I) \quad (6)$$

$$J(I) = c(\exp[\log 1 + 1/c \int_0^I p(I) dI] - 1) \quad (7)$$



III. Results and discussion:

Figure (1) represent the flowchart of the visual basic program. Different values of the parameter c have been used , and applied to satellite images. Its effects are shown in the following figures.

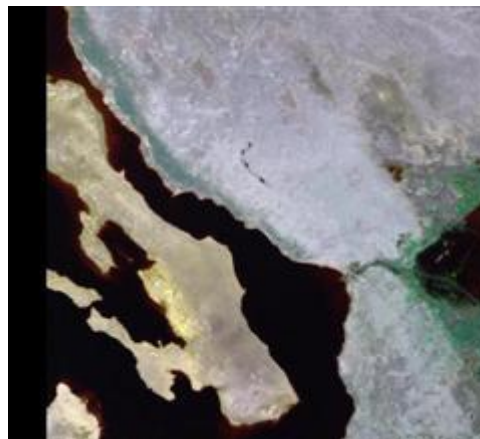
The histogram hyperbolization depends on the value of a special variable of the light intensity levels. It gave the satellite image details that is more clear and accurate than the original image.

We analyzed a Landsat image shown in figure(2), and its subset shown in figure (3), to find out which methods of digital image processing give the most useful results for

detecting and delimiting different features, by using different values of the controlling parameter c , shown in figures (7),(8),(9),(10),(11), and (12) . The best results were obtained with enhanced colour composites shown in figures (13) and (14), especially when histogram hyperbolization was applied: the features types that were known from the area were clearly visible in the image products, and also previously unknown regional patterns were found . Most conventional contrast enhancement algorithms usually fail to provide detailed contrast information in the dark and bright areas of remotely sensed images.



Figure(2): Landsat Image.



Figure(3): Subset of image.

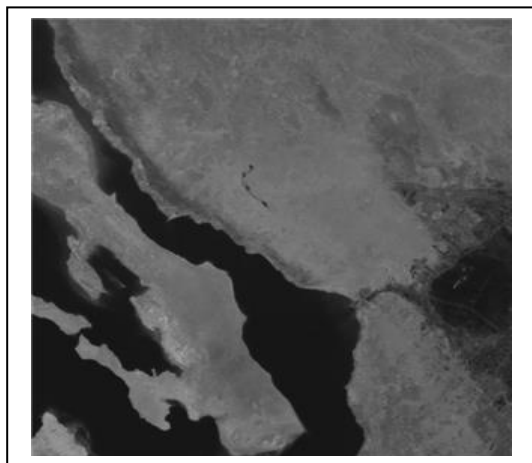
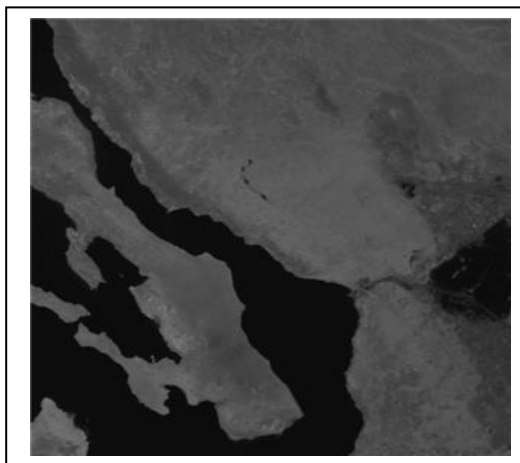
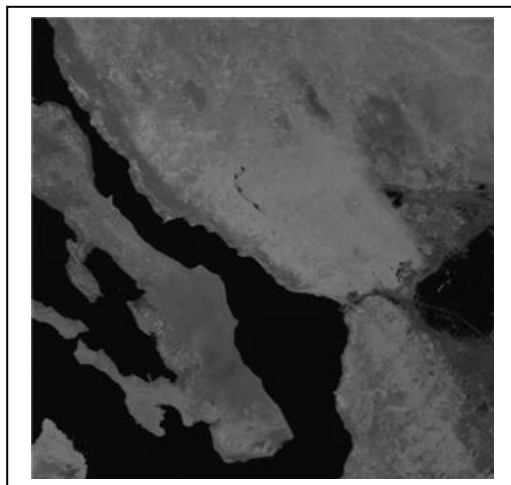


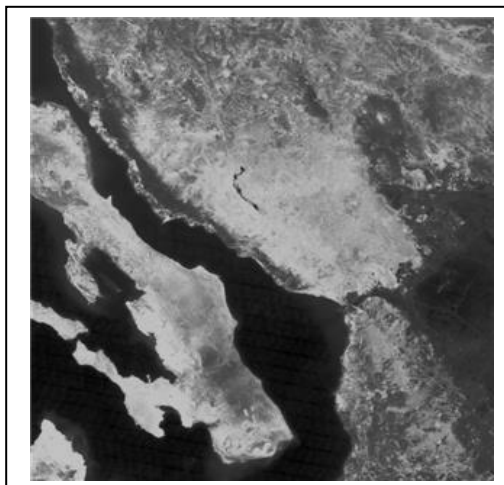
Figure (4): Band one.



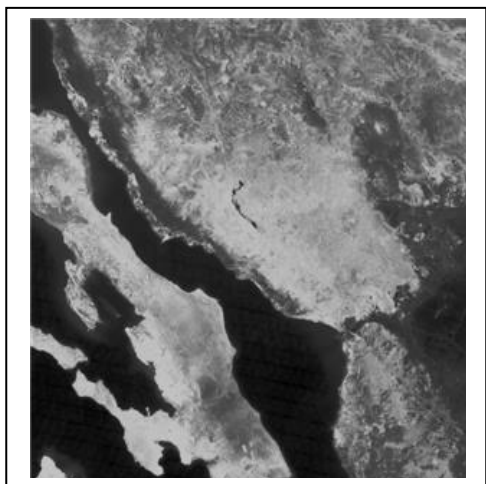
Figure(5): Band two.



Figure(6): Band three.



Figure(7): Band one , $c=4.5$.



Figure(8): Band one , $c=5.5$.

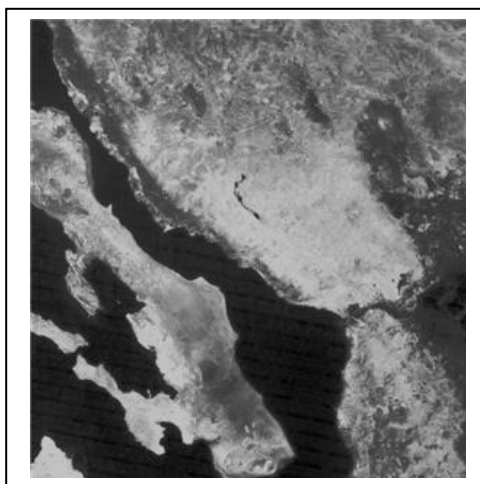
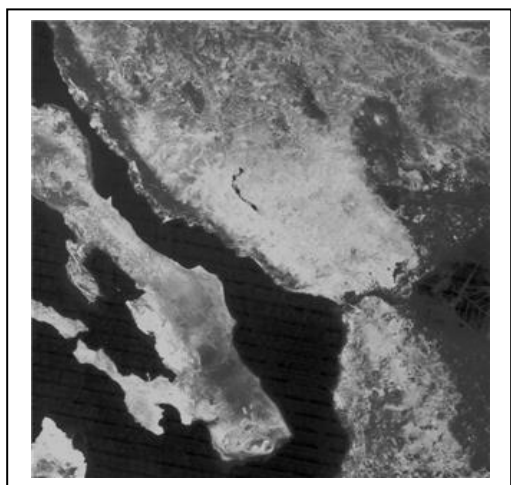
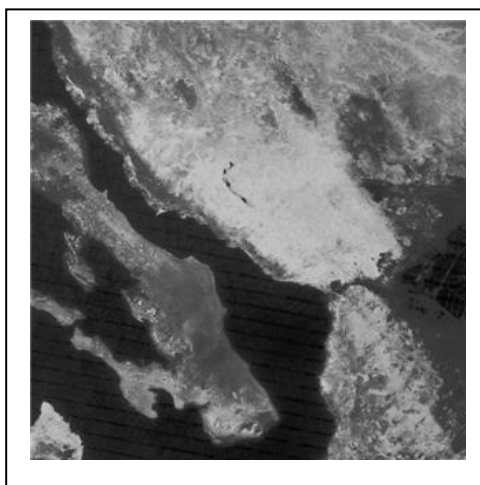


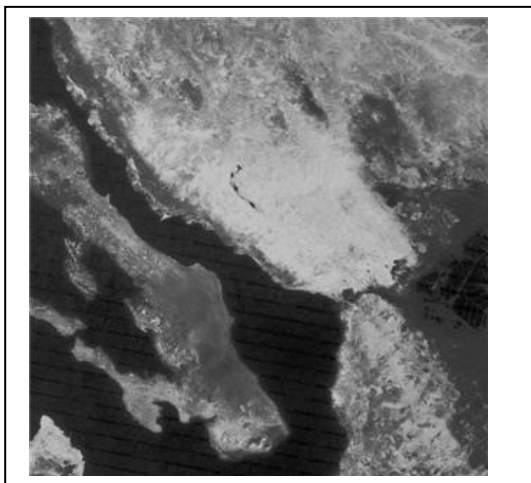
Figure (9): Band two, $c= 4.5$.



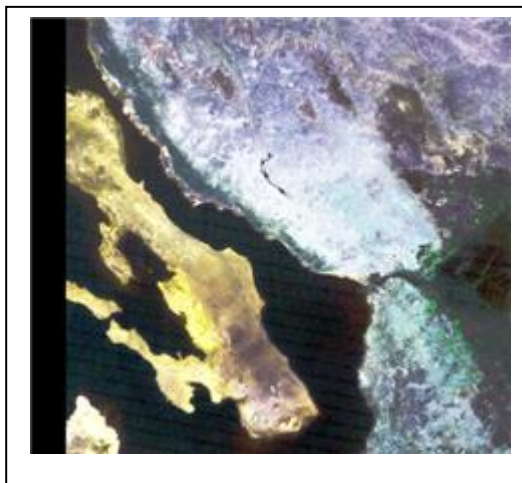
Figure(10): Band two , $c=5.5$.



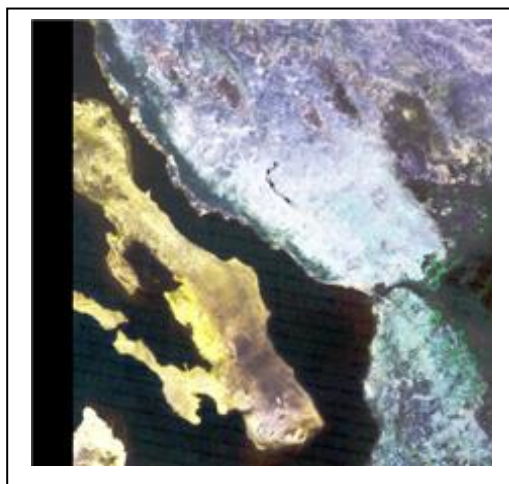
Figure(11): Band three, $c=4.5$.



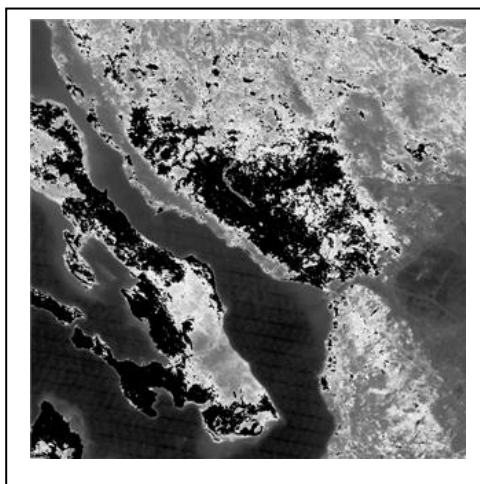
Figure(12): Band three, $c=5.5$.



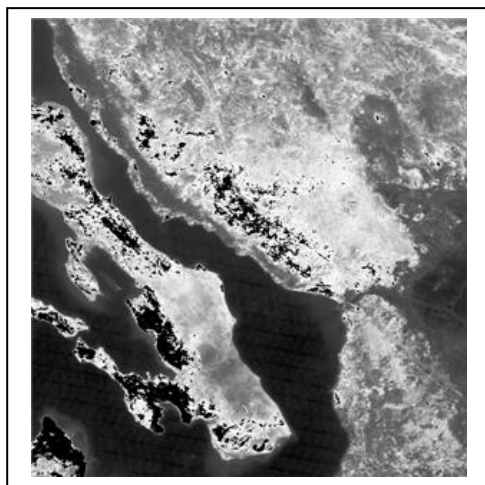
Figure(13): Band combination, $c=4.5$.



Figure(14): Band combination, $c=5.5$.



Figure(15): Band one , $c=3.5$.



Figure(16): Band one , $c=4$.

The effects of changing the values of the parameter c are also clear in figures (15) and (16). Therefore a special attention is necessary to avoid negative effects when applying the algorithm on satellite images. Distortion on image in these two figures is evident.

REFERENCES

1. Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", Addison-Wesley Publishing Company, 1992.
2. A., Jain, "Fundamentals of Digital Image Processing", Prentice-Hall Englewood Cliffs, NJ, 07632, 1989.
3. Lillesand Thomas M. and Kiefer R.W., "Remote Sensing and Image Interpretation. 3rd ed., New York, 1994.
4. Jensen, J.R., "Introductory Digital Image Processing, A Remote Sensing Perspective", Prentice Hall, Englewood Cliffs, New Jersey, 1986.
5. J. Gomes and L. Velho, "Image Processing for Computer Graphic", Springer-Verlag New York, Inc., 1997.
6. Hord, R. Michael, "Digital Image Processing of Remotely Sensed Data", Academic Press Inc. London, 1982.
7. ERDAS Field Guide, 1994, 3rd Edition, ERDAS, Inc. Atlanta.
8. Thomas S. Huang, "Image Enhancement using Histogram Hyperbolization", Computer Graphics and Image Processing, No., pp, 256-299, 1977.

تحسين الصور الفضائية باستخدام المخطط التكراري الفوقي

حسين زيدان علي

خبير معالجة صور

وزارة العلوم والتكنولوجيا

ملخص

ان أحد أهم العوامل النوعية للصور الفضائية يأتي من تباين الصورة. يشار الى تحسين التباين غالبا كأحد القضايا المهمة في معالجة الصور. يعتبر المخطط التكراري أداة هندسية فعالة لتصوير معلومات الصورة. تم في البحث تحليل صورة القمر لاندسات لايجاد قوة المخطط التكراري الفوقي و قابليته في كشف وتحديد المعالم المختلفة بالمشهد. صمم برنامج بلغة البيسك المرئي واستخدم على الحزم الطيفية للصورة الفضائية. تم دمج النتائج للحصول على صور مركبة ملونة باستخدام برنامج ايرداس النسخة 9.2. ان معظم خوارزميات تحسين تحسين التباين التقليدية تفشل عادة في ايضاح معلومات التباين التفصيلية في المناطق المضلمة و المضيئة لصور التحسس النائي. لقد اثبتت النتائج الاختبارية بان الطريقة المقترحة توفر صورة ذات تباين افضل من طرق التحسين التقليدية.

Financial Prediction using Inductive Models

Turki Y. Abdalla*

Abdulkareem. Y. Abdalla**

Lamya A. Dexon**

***College of engineering , University of Basrah**

****College of science, University of Basrah**

Abstract

Financial prediction is an example of a prediction problem which is challenging due to small sample sizes, high noise, non-stationary, and non-linearity. Neural networks have been frequently used in financial prediction because of their ability to deal with uncertain, fuzzy, or insufficient data. Despite that, neural networks(NN) have limitations; they still require a significant amount of a priori information about the model structure. Group Method of Data Handling (GMDH) is an inductive approach which attempts to overcome the subjectiveness of neural networks based on the principle of self-organization. We have developed an algorithm inspired from the evolutionary manner of conventional GMDH to generate an inductive model based on using multilayer perceptron that can avoid some of GMDH problems like the exhaustive computations on candidate Adalines and the increasing number of Adalines in the following layers.

1. Inductive and deductive modeling

The capability of induction is fundamental for human thinking. It is the next human ability that can be utilized in soft-computing, besides that of learning and generalization. The induction means gathering small pieces of information, combining it, using already collected information in the higher abstraction level to get complex overview of the studied object or process. Inductive modeling methods utilize the

process of induction to construct models of studied systems [1].

The inductive methods have a similar concept to that of evolution introduced by Holland where a number of solutions are created and an external criterion plays the role of finding the fittest [2].

Inductive methods are objective, they can produce a self organization models On the other hand, Deductive methods are subjective

since they are totally based on detailed instructions from the person performing the modeling as well as his/her idea about the object behavior [3].

2. Neural Networks in Financial Prediction

Many attempts have been made to predict the behavior of bonds, currencies, stocks, stock markets, or other economic markets . These attempts were encouraged by various evidences that economic markets do not behave randomly, but rather perform in a chaotic manner [4-8].

Neural networks (NNs) have become very important method for finance and investing applications because of their ability to deal with uncertain, fuzzy, or insufficient data which fluctuate rapidly in very short periods of time [9-11].

NNs differ from conventional techniques in that the analyst is not required to specify the nature of the relationships involved; the analyst simply identifies the inputs and the outputs. In addition, the NNs' main strength lies in its ability to vary in complexity, from a simple parametric model to a highly flexible, nonparametric model [11,12].

Despite the small number of assumptions in NNs in comparison to statistical methods,

they still require a significant amount of a priori information about the model's structure. Experts should decide on the quality and quantity of input arguments, the number of hidden layers and neurons as well as the form of their activation function [11].

3. GMDH Neural Networks

GMDH neural network is an inductive approach which attempts to overcome the subjectiveness of neural networks based on the principle of self-organization. An inductive approach is similar to neural networks but is unbounded in nature, where the independent variables of the system are shifted in a random way and activated so that the best match to the dependent variables is ultimately selected [12].

An idea behind GMDH algorithms which developed by Ivakhnenko (1966) is based on an evolution principle introduced by Holland which implies the *generation* and *selection* of the *candidate neurons* and an external criterion plays the role of finding the fittest [13].

The GMDH network is a kind of multilayered networks composed by a suitable combination of an Adalines which is illustrated in Figure 1.

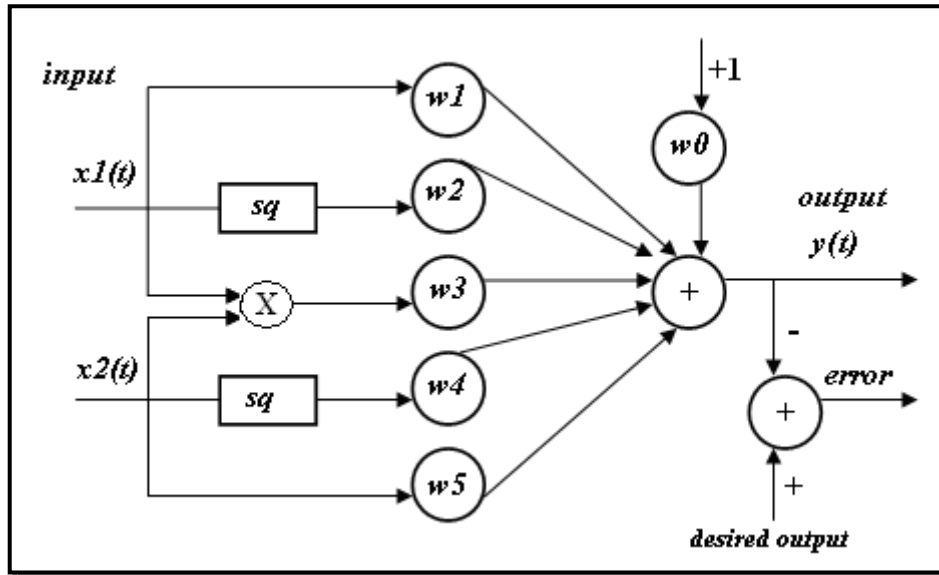


Figure 1: structure of Adaline with a bi-quadratic polynomial function

The output variable $y(t)$ is expressed as follows:

$$y(t) = w_0 + w_1 x_1(t) + w_2 x_1^2(t) + w_3 x_1(t) x_2(t) + w_4 x_2^2(t) + w_5 x_2(t) \quad ..(1)$$

Where w_i are weights, and are determined by the least squares method and x_1, x_2 are the inputs to the unit.

Let $W = [w_0 \ w_1 \ w_2 \ w_3 \ w_4 \ w_5]^T$ and $X = [1 \ x_1 \ x_1^2 \ x_1 x_2 \ x_2^2 \ x_2]^T$

The Widrow – Hoff delta rule adopted for training W is as follows [9, 10]:

$$W_{k+1} = W_k + \beta (X_k / |X_k|^2)(y_d^k - W_k^T X_k) \quad ..(2)$$

Where y_d^k is the desired output at time k , β is the learning rate with the range (0,1) and $|X_k|$ is the square of length of input vector. The application of equation (2) causes W to be modified so the difference between the desired and actual outputs will reduce. The

network structure is determined by the following steps [13-16]:

Step 1: separate the experimental data set into a training data set and a selection data set;

Step 2: create a layer of N Adalines based on the number of inputs. Every pair of inputs produces an Adaline. A combination C_2^i Adalines are created if there are i input variables;

Steps 3: use the training data to train all Adalines in the created layer and estimate the weighing coefficients for each Adaline with the least squares method;

Step 4: input the selection data set to the network and obtain the error of all trained Adalines in the layer. Assign a threshold; keep the Adalines whose errors are below the threshold and use them to make the next layer;

Step 5: If the smallest error of the current layer is larger than that of the previous layer or the current layer has only one Adaline, stop the training and trim the network.

Step 6: test the performance of the trained network with evaluation data set. The

evaluation data could be a combination of the training and selection data sets or a completely new data set.

The schematic diagram of a GMDH network with selection factor $F=0.5$ is shown in Figure 2.

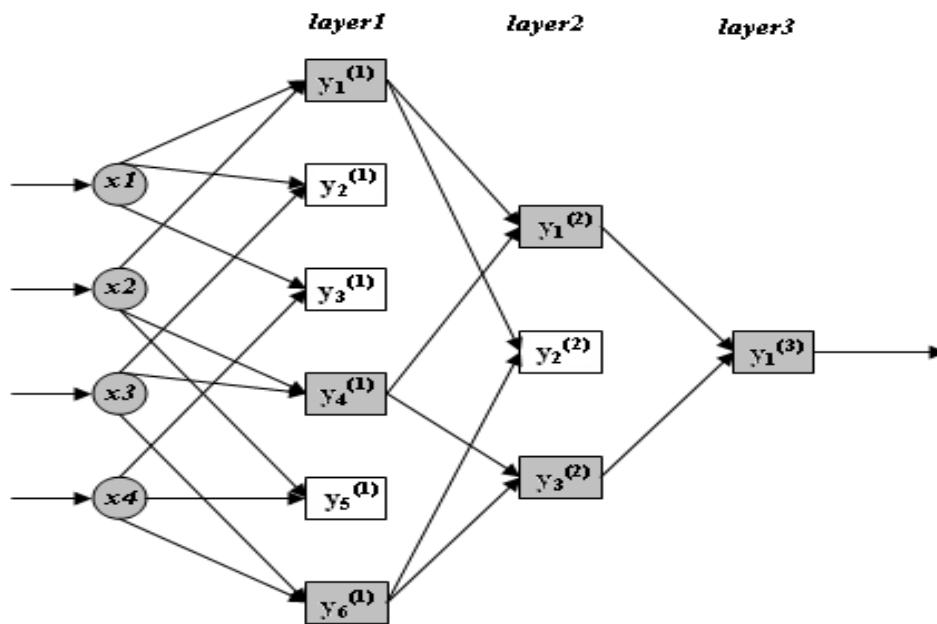


Figure 2 : The structure of neural network grown by GMDH algorithm

The Adalines that were selected after trimming process at the layers are depicted in Figure 2 as the gray boxes. A resulting network is the three-layer network consisting of six Adalines and four input nodes.

4. The Proposed Inductive Model

An idea behind our algorithm is to select the neurons one-by-one and then add them to a binary network according to an external

criterion . Only the neuron with the best selection error is added to an inductive network where the independent variables of the system are shifted in a random way and activated so that the best match to the dependent variable is ultimately selected. Following such a procedure there is a gradual increase of complexity and the optimum model is found which has a binary tree structure that is shown in Figure 3.

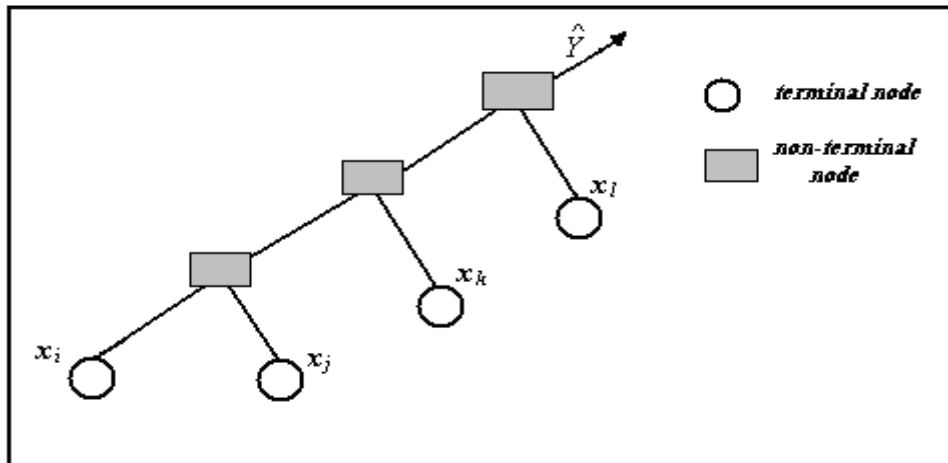


Figure 3: A binary tree with four terminals

In Figure 3, the binary tree takes four terminals or inputs $\{x_i, x_j, x_k, x_l\}$ to produce a predicted $\hat{Y}$, Where, i, j, k, l are in an arbitrary order. Each non-terminal node is performed a transfer function to estimate the output which is re-entered as an input in the following non-terminal node. The network is growing in an evolutionary principle; each time one input is added to the network until some a specific condition is verified.

Collected data are distributed between training and selection sets. Training set is used during training stage to estimate weight coefficients while the selection set is used to measure the generalization ability of the trained network to unseen data.

Deductive Neural Networks have the design problem. Also, these neural networks cannot provide an automated selection of essential input variables and all these omissions solved by trial and error experiments. Our idea is to exploit the good training ability of these deductive neural networks in building an inductive model (shown in Figure 3) where each non terminal node is a Multi-Layer Perceptron (MLP). The algorithm can be summarized as follows:

Step1: initialization: a population of N individuals (chromosomes) is randomly created. Each chromosome represents a proposed binary network. The chromosome has the form and the features as explained in Figure 4.

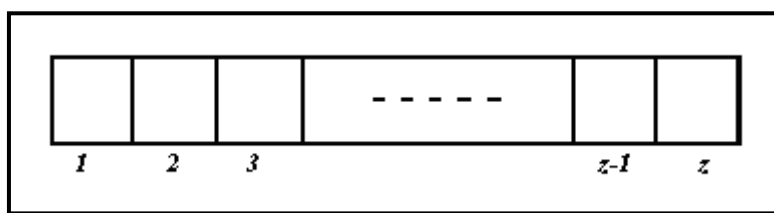


Figure 4: The form of the networks chromosome

Where:

The integer $i=1,2,...,z$: means the i^{th} input variable . The i^{th} input is not active if the corresponding gene contains zero, otherwise it is active.

Step2: **N** networks are constructed corresponding to **N** chromosomes then trained using training data set and a suitable training algorithm.

Step3: evaluation: a selection stage is performed by testing the generalization ability of all trained network to unseen data (selection data).

Step4: Reproduction: Genetic Algorithm operators of Rank selection, Uniform crossover, One-point mutation, evaluation and Replacement is repeatedly executed until a stopping condition is satisfied.

Figure 5 displays the network structure of this proposed inductive model. Each building unit in the proposed model is MLP trained using Back- Propagation learning algorithm; it has two inputs; hidden layer with one or more hidden neuron and one output neuron represents the predicted output.

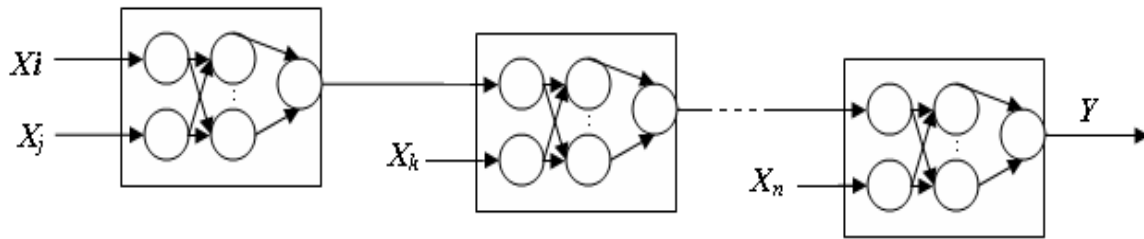


Figure 5: MLP inductive model network

5. Application of the proposed models for stock market Prediction

The value of a stock market price is established by analyzing the fundamental information associated with the market and affected it. The fundamental information consists of several input parameters which are used as input/output for our proposed prediction models. The input variables are:

1. Listed companies: number of listed companies in the market within a time period.

2. Exchange rates: the average of exchange rates between the local currency and US dollar within a period.

3. Market capitalization: Sum of the capitalizations of all listed companies in the market within a time period.

4. Value traded: the overall value of stocks that traded in the market within a time period.

5. Shares traded: sum of the shares (stocks) that traded in the market within a time period.

6. Days traded: number of the traded days in the stock market within a time period.

7. Turnover (TOR): is the result of division the value traded on the market capitalization in a time period multiplied by 100.

$$\text{TOR} = (\text{value traded} / \text{capitalization}) * 100$$

8. Capitalization average: is the result of the division of the Gross Domestic Product Average (GDPA) of the country to its market capitalization.

$$\text{Capitalization average} = \text{GDPA} / \text{capitalization}$$

The output variable is the price of the market's stock and it is computed as:

$$\text{Stock price} = \text{value traded} / \text{shares traded}.$$

A Saudi stock market is taken as a case study. The application data consist of the monthly averaged for the input parameters described above. There are 141 trading

months within the period from January 1996 to September 2007 taken from the website www.amf.org [17].

The input and the output data sets to the networks are preconditioned by normalizing the data. The purpose of normalizing the data is to modify the variable levels to a reasonable value. Normalizing data entails manipulating it to be in the range between 0 and 1. The networks are trained utilizing the normalized data sets. In our work data are first preprocessed by log function then normalization is done by dividing each data item on the greatest input value.

A Genetic Algorithm population of 7 binary individuals (proposed networks) is randomly initialized. The algorithm that is described in earlier is executed with Genetic parameters of 0.9 for crossover and 0.1 for mutation. The network that is shown in Figure 6 is the final result of the algorithm execution.

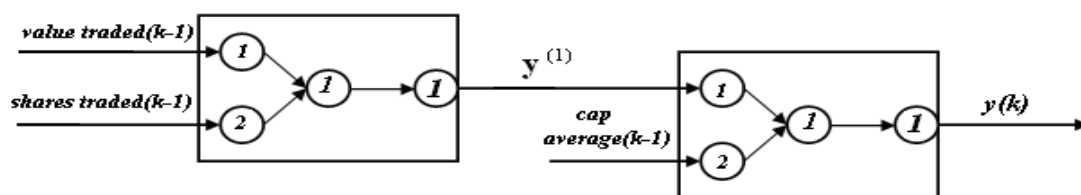


Figure 6: Trained MLP based inductive network

Trained MLP inductive model consists of two MLP networks; each one of them consists of 2 input units, 1 hidden and 1 output neurons. The first and the second MLP

networks are trained by 8000 and 6000 training cycles respectively. The adopted learning rate was 0.45 and the momentum was 0.5. The prediction ability of the trained

MLP inductive model for stock market application was tested by applying it with the

overall data and the test results for the prediction are shown in Figure 7.

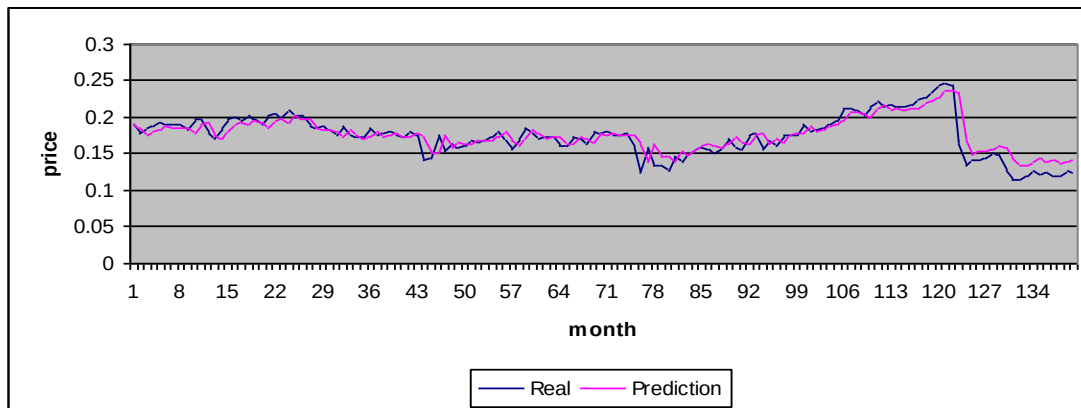


Figure 7: MLP based inductive model prediction results

6. Conclusion

MLP inductive model gives good prediction results. For comparison, the SE errors at the evaluation stage are computed for the proposed model and some other types of NNs as shown in Table 1.

$$SE = \sum_k (y_t(k) - y(k))^2$$

Where: $y_t(k)$ is the desired output and $y(k)$ is the network output at time k

Table 1: SE of the proposed model for stock market prediction application

No	proposed method	SE
1	Bilinear GMDH NN	0.211835
2	Biquadratic GMDH NN	0.069382
3	Modified Biquadratic GMDH NN	0.174637
4	MLP network	0.112403
5	MLP inductive model	0.02435

References:

[1] P. Kordik, M. Snorek, M. Genyk-Berezovskyj, "Hybrid Inductive Models: Deterministic Crowding Employed", 2004.

[2] H-S Park, B-J Park, H-K Kim, and S-K Oh, "Self-Organizing Polynomial Neural

Networks Based on Genetically Optimized Multi-Layer Perceptron Architecture", International Journal of Control, Automation and Systems, vol. 2, no. 4, pp. 423-434, 2004.

- [3] L. Anastasakis, and N. Mort, "The Development of Self-Organization Techniques in Modelling: A Review of the Group Method of Data Handling (GMDH)", 2001.
- [4] K. Chakraborty, M. Kishan, K. M. Chlukuri, and R. Sanjay, "Forecasting the Behavior of Multivariate Time Series Using Neural Networks", Neural Networks, 5,96 1-970, 1992.
- [5] R. Tsay, "Analysis of Financial Time Series", Wiley Inter science, 2005
- [6] M. E. Malliaris, "Modeling the Behavior of the S&P500 Index: A Neural Network Approach", Proc. 10th IEEE Conference on AI for Applications, 86-90, 1994.
- [7] J. Yang, and N. Gradjoevic , "Non-linear, non-parametric, non-fundamental exchange rate forecasting". Journal of Forecasting 25(4), 2006.
- [8] AK Dhamija, and VK Bhalla, "Financial Time Series Forecasting: Comparison of Neural Networks and ARCH Models", International Research Journal of Finance and Economics , Issue 49, (2010).
- [9] E. Schoeneburg, "Stock Price Prediction Using Neural Networks: A Project Report", Neurocomputing, vol. 2, 1990, pp. 17-27.
- [10] S. Simon, and A. Rao, "Accuracy driven artificial neural networks in stock market prediction", International Journal on Soft Computing (IJSC) Vol.3, No.2, May 2012.
- [11] N. W. Tan, "A Hybrid Financial Trading System Incorporating Chaos Theory, Statistical and Artificial Intelligence/Soft Computing Methods", Queensland Finance Conference 1999.
- [12] G. R. Weckman, and R. Agarwala, "Identifying Relative Contribution of Selected Technical Indicators in Stock Market Prediction", 2003.
- [13] A. G. Ivakhnenko, and J-A. Muller, "Recent Developments of Self-Organising Modelling in Prediction and Analysis of Stock Market", 2000.
- [14] M. Caudill, and C. Butler, "Understanding Neural Networks: Computer Explorations", 2000.
- [15] V. Schetinin, J. Schult, and A. Brazhnikov, "Neural Network Techniques for visual mining clinical electroencephalograms", 2005.
- [16] A. Sakaguchi, and T. Yamamoto, "A Study on System Identification Using GA and GMDM Network", IEEE 2003.
- [17] [86] Arab Monetary Foundation, www.amf.org.

New Three Methods for Improving Initialization of k -Means Clustering

Abbas H. Hassin Alasadi, Moslem Mohsinn Khudhair*

Department of Computer Science, College of Science, University of Basra, Basra, Iraq

**Corresponding author: Email address : abbashh2002@yahoo.com (Abbas H. Hassin),
P..Box 716, Ashar, Basrah, Iraq.
Tel. 009647809835559, Fax: 0096440414811*

Abstract

The traditional k -means algorithm is a classical clustering method which widely used in variant application such as image processing, computer vision, pattern recognition and machine learning. It is known that, the final result depends on the initial starting points. Generally, initial cluster centers are selected randomly, so the algorithm could not lead to the unique result. In this paper, we present a new algorithm which includes three methods to compute initial centers for k -means clustering. First one is called geometric method which depends on equal areas of distribution. The second is called block method which segments the image into uniform areas. The last method called hybrid which combined between first and second methods. The experimental results appeared quite satisfactory.

Keywords: clustering; k -means algorithm; color Image; Image segmentation

1. Introduction

Image segmentation is the first step of the most critical tasks of image analysis, as shown in Figure (1). It is used either to distinguish objects from their background or to partition an image onto the related regions [1, 2].

The process of image segmentation is defined as: “the search of homogenous regions in an image and later the classification of these regions”. It also means the partitioning of an image into meaningful regions based on homogeneity or heterogeneity criteria. Image segmentation techniques can be differentiated

into the following basic concepts: pixel oriented, Contour-oriented, region-oriented, model oriented, color oriented and hybrid. Color segmentation of image is a crucial operation in image analysis and in many computer vision, image interpretation, and pattern recognition system, with applications in scientific and industrial field(s) such as medicine, Remote Sensing, Microscopy, content based image and video retrieval, document analysis, industrial automation and quality control. The performance of color segmentation may significantly affect the quality of an image understanding system. The

most common features used in image segmentation include texture, shape, grey level intensity, and color [3].

Partitional clustering algorithms such as *k*-means and Expectation Maximize (EM) clustering are widely used in many applications such as data mining, compression, image segmentation, and machine learning.

Therefore, the advantage of clustering algorithms is that the classification is simple and easy to implement. Similarly, the drawbacks are of how to determine the number of clusters and decrease the numbers of iteration.

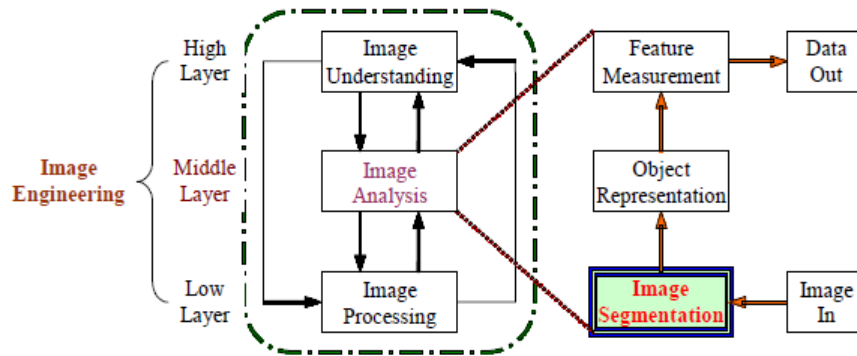


Figure (1): Image Engineering and Image Segmentation [1].

2. Related Works

Several attempts were made by researchers to improve the effectiveness and efficiency of the *k*-means algorithm. There are many researchers suggest initialized method of centroids of *k*-means algorithm.

Bradley and Fayyad [4] present a technique for initializing the *K*-means algorithm. They begin by randomly breaking

the data into 10, or so, subsets. Then they perform a *K*-means clustering on each of the 10 subsets, all starting at the same set of initial seeds, which are chosen using Forgy's method. The result of the 10 runs is 10*k* centre points. These 10*k* points are then used as inputs of *K*-means algorithm and the algorithm run 10 times, each of the 10 runs initialized using the *k*-final center locations (known as centroid)

from one of the 10 subset runs. The resulting K -center locations are used to initialize the K -means algorithm for the entire datasets.

Douglas and Michael [5] proposed a method to select a good initial solution by partitioning dataset into blocks and applying k -means to each block. But the time complexity is slightly more. Though the above algorithms can help finding good initial centers for some extent, they are quite complex and some use the k -means algorithm as part of their algorithms, which still need to use the random method for cluster center initialization.

3. Clustering Analysis

Clustering analysis is one of the major data analysis methods widely used in many practical applications of emerging areas.

Clustering is the process of finding groups of objects such that the objects in a group will be similar to one another and different from the objects in other groups. A good clustering method will produce high quality clusters with high intra-cluster similarity and low inter-cluster similarity. The quality of a clustering result depends on both the similarity measure used by the method and its implementation and also by its ability to discover some or all of the hidden patterns [6,7].

There are two main branches of clustering: (1) hierarchical and (2) partitional [8]. In this paper, we concentrate on partitional clustering. Particularly, a popular partitional

clustering method called k -means clustering. The problem of clustering is to partition a data set consisting of n points embedded in m -dimensional space into K distinct set of clusters such that the data points within a cluster are more similar among them than to data points in other clusters. There are a number of proximity indices that have been used as similarity measures [9]. Unfortunately, K -means algorithm is extremely sensitive to the initial choice of cluster centers, and a poor choice of centers may lead to a local optimum that is quite inferior to the global optimum [10,11].

4. K-means Clustering Algorithm

Let $X = \{x_1, x_2, \dots, x_n\}$ be the set of n data in d -dimensional points to be clustered into a set of k clusters, $C = \{c_1, c_2, \dots, c_k\}$; K -means algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized [12, 13]. Let μ_k be the mean of cluster c_k . The sum squared error (SSE) between μ_k and the points in cluster c_k is defined as:

$$SSE = \sum_{X_i \in c_k} \|X_i - \mu_k\|^2 \quad (2-5)$$

The goal of K -means is to minimize the sum of the squared error over all k clusters,

$$SSE = \sum_{k=1}^K \sum_{X_i \in c_k} \|X_i - \mu_k\|^2 \quad (2-6)$$

The centroid (mean) of the i^{th} cluster is defined by Equation (2) [14]:

$$\mu_i = \frac{1}{m_i} \sum_{x \in C_i} X \quad (2-7)$$

where m_i the number of objects in the i^{th} cluster
 K-means starts with an initial partition
 with k clusters and assign patterns to clusters so
 as to reduce the squared error. Since the
 squared error always decrease with an increase

in the number of clusters k (with $SSE=0$ when
 $k=n$), it can be minimized only for a fixed
 number of clusters [15]. A pseudo-code for the
 K-means algorithm1 is shown in Figure (3).

```

1. Input data           // Image of size (length x width)
2. Input k             // Number of clusters
3. MSE = Large number ;
4. CALL Initialization ;
5. Do
6. Old-MSE = MSE ;
7. MSE1 = 0 ;
8. For j = 1 to k
9.      $\mu_j = 0$  ;
10.     $n_j = 0$  ;
11.  end for j
12.  For i = 1 to n
13.    For j = 1 to k
14.      Compute squared Euclidean distance  $d^2(x_i, m_j)$  ;
15.    end for j
16.    Find the closest centroid  $m_j$  to  $x_i$  ;
17.     $\mu_j = \mu_j + x_i$  ;
18.     $n_j = n_j + 1$  ;
19.     $MSE1 = MSE1 + d^2(x_i, \mu_j)$  ;
20.  end for i
21.  For j = 1 to k
22.     $n_j = \max(n_j, 1)$  ;
23.     $\mu_j = \mu_j / n_j$  ;           // Re-calculation centroids
24.  end for j
25.  MSE = MSE1 ;
26. WHILE (MSE - MSE1 < T)
    
```

Figure (3): A pseudo-code for the K-means algorithm1.

The specification of the points of cluster centroids is done by **call initialization** which is depicted in Figure (4).

CALL Initialization

Select initial cluster centroids $\{ \mu_j \}$ ($j=1, 2, \dots, k$) randomly
 from input image that has bounded with size (length x width) ;

Return

Figure (4): Call algorithm for initializing cluster centroids.

5. Proposed Methods and Experimental Results

We proposed an efficient algorithm which consists of three methods to compute initial centers for k -means clustering. First one is called geometric method which depends on equal areas of distribution. The second is called block method which segments the image into uniform areas. The last method called hybrid which combined between first and second methods.

5.1. Geometric K -means Algorithm

A propose algorithm is presented and includes a method to compute initial center clusters for K -means clustering. It is called geometric method which depends on equal areas of distribution. The *geometric algorithm2* has the same *algorithm1* in Figure (3) but it is different in the method of generation of initial centroid of clusters. Figure (5) depicts an algorithm of initialized points of cluster centroids.

```

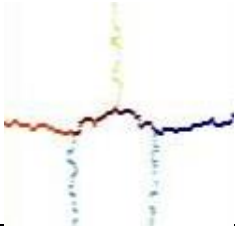
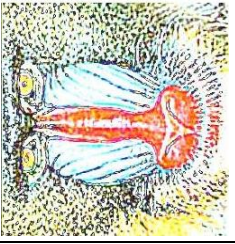
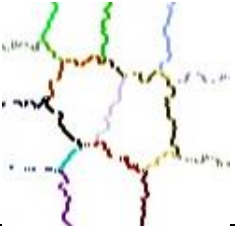
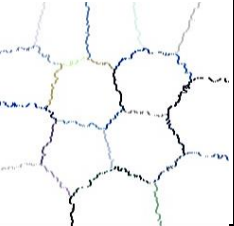
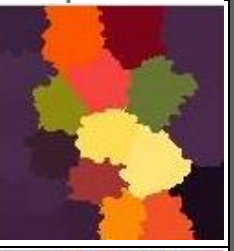
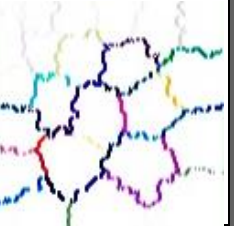
CALL Initialization
     $r = \text{Integer}(\sqrt{K})$ ;
     $q = \text{Integer}(K / r)$ ;
     $p1 = \text{Integer}(\text{width} / r)$ ;
     $p2 = \text{Integer}(\text{height} / q)$ ;
     $xx = \text{Integer}(p1 / 2)$ ;
     $yy = \text{Integer}(p2 / 2)$ ;
    For  $i = 1$  to  $r$ 
        For  $j = 1$  to  $q$ 
             $x(1,i) = xx$ 
             $y(1,j) = jj$ 
             $xx = xx + p1$ 
        Next  $j$ 
         $yy = yy + p2$ 
         $xx = \text{Int}(p1 / 2)$ 
    Next  $i$ 
Return
  
```

Figure (5): geometric Algorithm2 distribution for initializing cluster centroids.

In this method, the algorithm scans the dataset block by block. The same calculations and experiments in previous the algorithm have been implemented. Although this algorithm is more expensive in calculation than the previous algorithm but it is regarded more accurate for images that have regular distribution intensity.

This matter is very considered to be very important in later section to implemented hybrid algorithm.

Figure (6) shows experimental results obtained by implementing geometric K -means algorithm including different operations on four images.

Original Image	Segment. result	Smooth & Lap.	Region Cluster.	Sobel Edge	K	No. Iteration
					5	20
					12	23
					15	59
					24	20

5.2. Block K-means Algorithm

The proposed algorithm is compresses the data into smaller dataset by producing K -means from each block, if the dataset contains m blocks, then the compressed data will contains

$(k \times m)$ objects. Our algorithm scans the original dataset only one times and produces better clusters. Figure (7) summarizes the steps of the block K -means algorithm2.

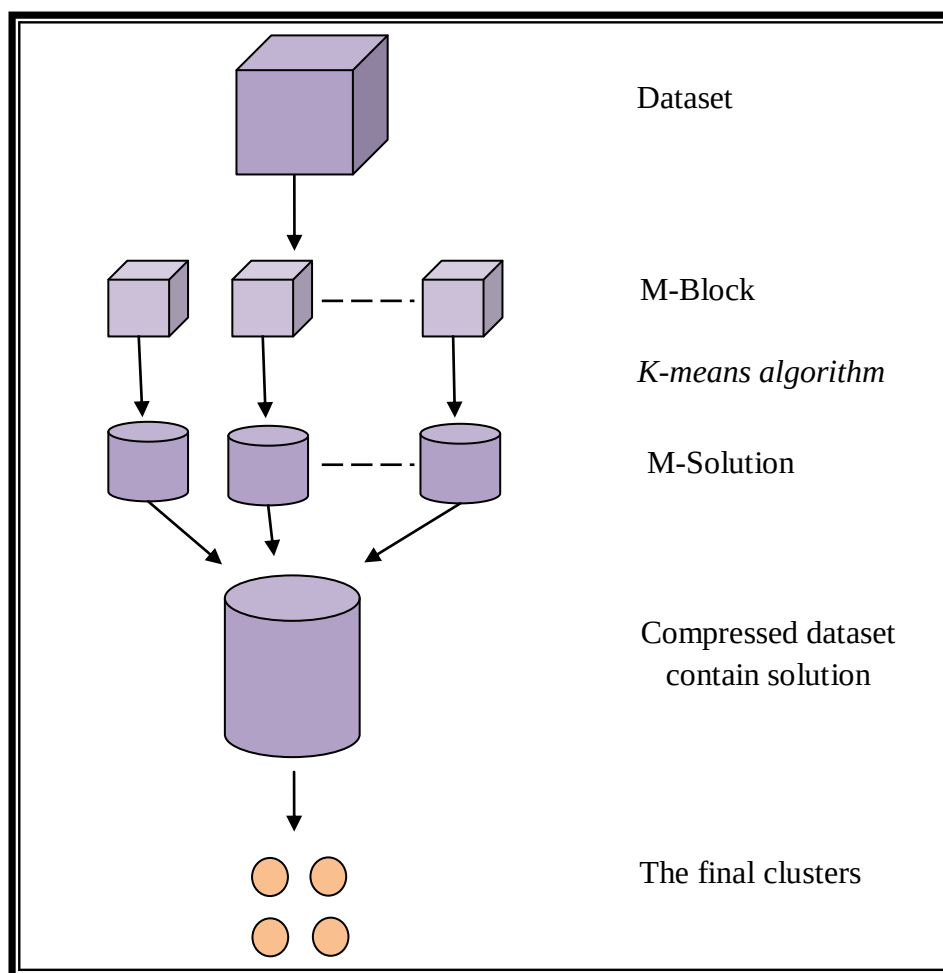


Figure (7): An overview of the Block K-means.

The main idea of this algorithm is to compress the dataset into finite number of representatives. Each representative indicates the mean value of some data points form a small cluster. This process has been done at the

first phase. In the second phase we apply the *k*-means on the compressed dataset. Figure (8) exhibits the *block algorithm*.

```

1. Set the size of the block to s length;
2. row = width/s ;
3. col = height/s ;
4. For i = 1 to row
5.     For j = 1 to col
6.         read block(i, j);
7.         Call algorithm1 ; // With Random Initialization
8.         save the result ;
9.     end for j
10. end for i

```

Figure (8): Block K-means algorithm3.

Note that, in Figure (8), the method determines the size of each block and the user should determine the required number of partitions in each block. It is clear that in line 7, it is required to call implementing standard K -means algorithm for each block as same as *algorithm1*. From experimental results it will

be better to use a small value of k (2-9) for each block and depends on size of image.

Figure (9) shows experimental results obtained by implementing block K -means algorithm including different operations on four images.

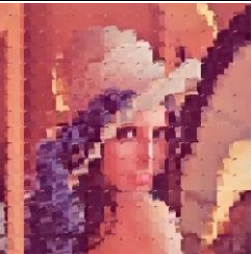
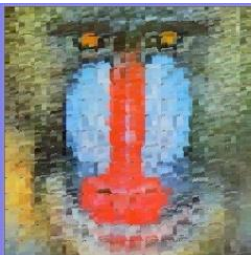
Original Images	Segment Result	Smooth& Laplacian	K
			5
			12
			15
			24

Figure (9): Result of execution Block K -means algorithm.

5.3. Hybrid K-means Algorithm

Hybrid K-means algorithm combined of the block K-means algorithm and the geometric K-means algorithm. It's the same block method

but it has used the geometric initialized method instead of randomly distributed cluster centroids. Figure (10) exhibits the *hybrid algorithm4*.

```

1. Set the size of the block to s length ;
2. row = width/s ;
3. col = height/s ;
4. For i = 1 to row
5.     For j = 1 to col
6.         read block(i, j);
7.         Call algorithm1 ; // With Geometric Initialization
9.         save the result ;
10.    end for i
11. end for j

```

Figure (10): Hybrid K-means algorithm4.

Note that, in Figure (10), the method determines the size of each block. It is clear that in line 7, It is demanded to call implementing geometric K-means algorithm for each block as same as *algorithm2*.

Figure (11) shows experimental results obtained by implementing hybrid K-means algorithm including different operations on four images.

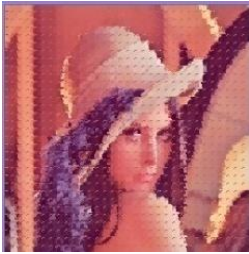
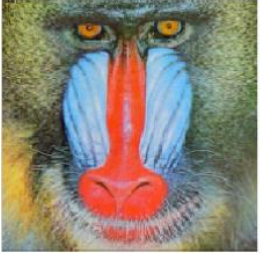
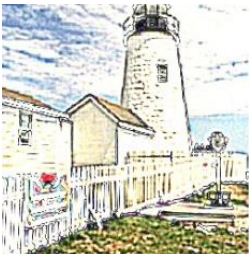
Original Images	Segment Result	Smooth & Lap.	K
			5
			12
			15
			24

Figure (11): Result of execution hybrid K-means algorithm.

6. Conclusion

K-means algorithm is a popular clustering algorithm applied widely, but do not always guarantee good results as the accuracy of the final clusters depend on the selection of initial centroids. Experimental results show that selecting centroids by our methods can lead to a better clustering. Moreover, the computational complexity of the standard algorithm is objectionably high owing to need reassign the

data points a number of times for each iteration in the loop.

We have evaluated our methods on several different standard images. So, we have compared our results with that of *k*-means algorithm in terms of the total execution time and quality of clusters. Our experimental results are executed on PC 2.0GMHz CPU, 2.0GB RAM, 512 KB Cache.

Finally, we have compared the *CPU* time of the proposed methods (geometric, block & hybrid) with the standard *K*-means methods. The execution time of proposed methods is much lesser than the average execution time of *K*-means as shown in Figure (12).

Figure (13) demonstrates that the proposed methods (geometric, block, and

hybrid) provide better cluster accuracy than the existing methods. It shows that proposed method performs much better than the random initialization algorithm. This is due to the initial cluster centers generated by proposed method which are quite closed to the optimum solution.

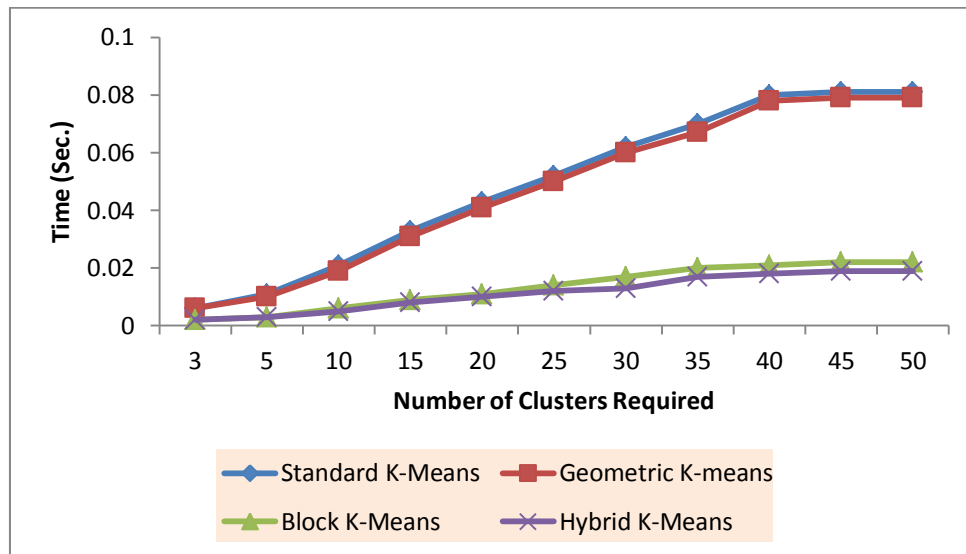


Figure (12) Execution Time (Lena Image).

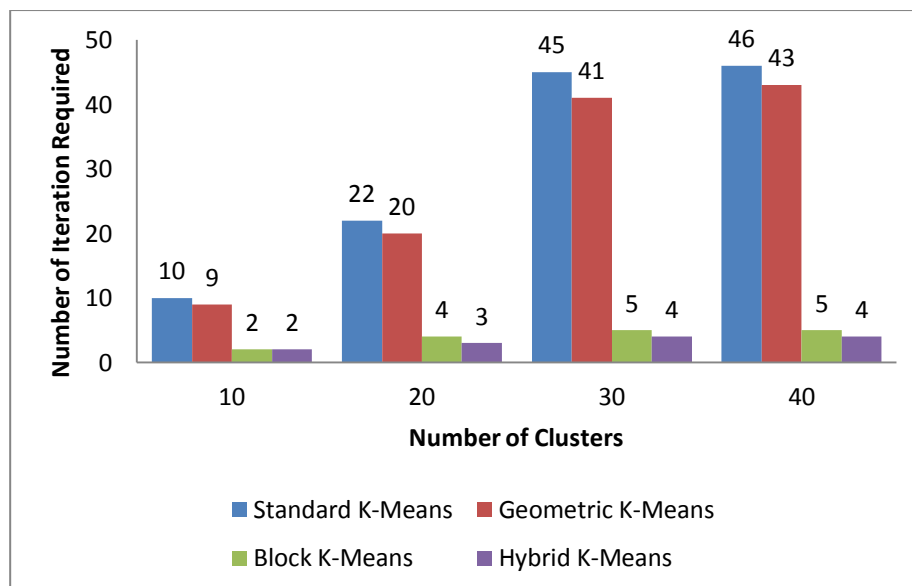


Figure (13) Number of Iteration (Lena Image).

References

- [1] Yu-Jin Zhang, " Advances in Image and Video Segmentation", IRM Press, 2006.
- [2] Ali Salem Bin Samma and Rosalina Abdul Salam, Adaptation of *K*-Means Algorithm for Image Segmentation, Int. J. Signal Processing 5:4 (2009) 270-274.
- [3] Anil Z. Chitade, Dr. S.K. Katiyar, Colour Based Image Segmentation Using *K*-Means Clustering, Int. J. Eng. Sci. Tech. 2:10(2010) 5319- 5325.
- [4] Paul S. Bradley and Usama M. Fayyad," Refining Initial Points for *K*-means Clustering", International Conference 15th on Machine Learning (ICML98), pp. 91-99. Morgan Kaufmann, San Francisco, 1998.
- [5] Douglas Steinley, Michael J. Brusco, Initializing *K*-means Batch Clustering: A Critical Evaluation of Several Techniques, J. Classification 24 (2007) 99-121.
- [6] Rajashree Dash, Debahuti Mishra, Amiya Kumar Rath, Milu Acharya, A Hybridized *K*-Means Clustering Approach for High Dimensional Dataset, Int. J. Eng. Sci. Tech. 2:2(2010) 59-66.
- [7] Eshref Januzaj, Hans-Peter Kriegel, and Martin Pfeifle, " Towards Effective and Efficient Distributed Clustering", Workshop on Clustering Large Data Sets (ICDM), Melbourne, FL, 2003.
- [8] Fisher, D., Knowledge Acquisition via Incremental Conceptual Clustering. Mach. Learn.1987.
- [9] Jain, A.K., Murty, M.N., Flynn, P.J., Data clustering: A review, ACM Comput. Surveys 31:3(1999) 264–323.
- [10] Anderberg, M.R., Cluster Analysis for Applications, Academic Press Inc.,1979.
- [11] Xiaoping Qing, Shijue Zheng, A New Method for Initializing the *K*-means Clustering Algorithm, 2nd Int. Sym. Knowledge Acquisition and Modeling, (2009) 41-44.
- [12] Joaquín Pérez O., Rodolfo Pazos R., Laura Cruz R., Gerardo Reyes S., Rosy Basave T., and Héctor Fraire H., " Improving the Efficiency and Efficacy of the *K*-means Clustering Algorithm Through a New Convergence Condition", O. Gervasi and M. Gavrilova (Eds.): ICCSA, LNCS 4707, Part III, pp. 674–682, 2007.
- [13] Shehroz S. Khan, and Amir Ahmad, "Cluster center initialization algorithm for *K*-means clustering ", Pattern Recognition Letters 25, pp.1293–1302, 2004.

[14] Zuyi Huang, Fatih Senocak, Arul Jayaraman, and Juergen Hahn, "Solution of Inverse Problems for Obtaining Protein Concentrations from Fluorescent Microscopy Images", American Control Conference, Hyatt Regency Riverfront, St. Louis, MO, USA, pp. 1688-1693, 2009.

[15] Hong Liu and Xiaohong Yu, "Application Research of *K*-means Clustering Algorithm in Image Retrieval System", Proceedings of the Second Symposium International Computer Science and Computational Technology(ISCST'09) Huangshan, P. R. China, pp. 274-277, 26-28 Dec. 2009.

A Field Bus Network With CAN Protocol And a Fuzzy Neural Petri Net Controller

Ali A. Abed¹, AbdulAdhem A. Ali², Ali F. Marhoon³, Nauman Aslam⁴

^{1,2,3}College of Engineering-University Of Basra, IRAQ.

⁴College of Engineering-Dalhousie University, CANADA.

*¹ aaad_bah@yahoo.com, ² abduladem1@yahoo.com, ³ ali_marhoon2003@yahoo.com,
⁴ naslam@dal.ca*

Abstract

In field bus systems different smart sensors are functionally interconnected, requiring exchange of information accurately with a specified communication latency and sample time. This paper describes implementation methodologies of efficient distributed real time controllers for monitoring and actuation. Software for these controllers with CAN communication protocol are written. CAN-Based embedded microcontrollers and transceivers are used for the practical implementation. Temperature control by a robust neural fuzzy Petri net (RNFPN) controller based on an indirect forward control structure is also proposed. After offline learning to get the initial weights, the RNFPN is online constructed by concurrent structure/parameter learning. The RNFPN has many advantages when applied to temperature control plants such as: high learning ability which reduces the controller training time, no a priori knowledge of the plant is required which simplifies the design task, and lastly the high control performance. As a case study, the field bus system is tested with a water bath temperature control system. The RNFPN control of the plant is implemented in the host computer connected to the master controller via the RS232 port. Implementation results provide a good indication that the CAN protocol is a high reliable and can be potentially adopted in numerous industrial applications. Also, the RNFPN intelligent controller has a reasonable robustness against disturbance, rapidity and good dynamic performance.

I. INTRODUCTION

The origin of field bus was to replace the point-to-point 4-20mA analog links between the field devices (sensors and actuators) and their

controllers (e.g. PLC's, CNC's,...) with a digital single link on which all data is transmitted serially and multiplexed in time. The serial transmission reduces the number of data lines

and increases distance. There are many benefits of using field bus instead of the point-to-point communications such as:

- Flexibility: network extension is easy.
- Larger distance can be satisfied.
- Physical wiring reduction.
- Reduced cost.
- Simple installation, operation and maintenance.
- Interoperability: connecting of units from different manufacturers.
- High reliability.

The field bus played a major role in the modern industry automation development. It is used for interconnecting real-time distributed controllers of the DCS (Distributed Control Systems). It gives the user the freedom in choosing the suitable protocol that best matches his needs without changing the whole automation system architecture. Since the field bus is a network, there is a relation with the famous 7-layers OSI model. However, for real time applications layers 3, 4, 5, and 6 are not considered since they deal with the data transfer among networks. So, the field bus is a 3-layers architecture with physical, data link, and application layers.

The communication protocol is responsible of two main rules in the field bus: medium access control (MAC) and synchronization among the different slaves. Choosing a MAC protocol is a crucial issue in the design of a DCS system. There is a need to know how the field bus works in order to choose the suitable communication protocol for the application. There are many types of field bus such as: BACnet, ISP, CAN,

CEBus, WorldFIP, etc [1]. Profibus and WorldFIP are used widely in industrial control applications e.g. [2] [3]. So, in this paper we adopt the CAN (Controller Area Network) protocol for this reason and due to the high specifications of the CAN. The field bus network may then be used in the design of a DCS. There are three main issues that must be considered in the design of a DCS systems:

- The adopted communication protocol.
- The adopted network topology.
- The Interoperability: Using components from different manufacturers.

The design of a DCS should satisfy many requirements such as: QoS (Quality of Service) of the communication protocol, system throughput, system performance, fast response time, and some real time requirements such as sample time.

Soft computing techniques like fuzzy logic, neural networks, and fuzzy neural networks have been used widely in the identification and control of dynamic systems [4]. The merging of fuzzy and Back propagation neural network (BPNN) has inspired new resources for the design of efficient controllers [5]. These types of computing techniques have many advantages, such as: it is used for nonlinear systems, model-free systems, and good self learning abilities. In [6] the BPNN has been applied to temperature control system but the convergence speed is slow, which makes it unsuitable for real time applications. In [7] a neural fuzzy inference network is proposed for

temperature control and proved good learning ability but the inclusion of too many input and output variables increases the network size and decreases the learning speed. In [8], a recurrent neural fuzzy network controller is presented. In [9], a dynamic fuzzy neural network is introduced. In these recurrent neural fuzzy networks i.e. in [8,9], one common characteristic is that the recurrence is achieved by including external feedback. To apply these networks to temperature control problems, they still need to know the order of both control input and network output to participate in the recurrent model. In [5], a TSK-type (Tagagi-Sugeno) recurrent fuzzy neural network is proposed. It proved a good performance but with inverse control structure and it is still need at least 8 input nodes and 10^7 iterations of training to get a sum of square error (SSE) of 3.1. Therefore, in spite of that the fuzzy neural (FNN) has been proved to be a powerful technique in system control but its real time applications may be difficult due to heavy computations especially if huge parameters are to be tuned. In order to overcome this problem, the concept of a Petri net (PN) is included into the FNN to reduce the redundant or inefficient computations and hence improves the control reliability. The PN consists of “places” and “transitions” with the input and output functions [10]. The basic concept of PN is incorporated into the conventional FNN to get what is called NFNN controller.

The rest of the paper is organized as follows: Section II is concerned to the CAN communication protocol with its main features and benefits. Section III deals with main principles and requirements for the DCS system design. Section IV describes the hardware components needed for our field bus network. In section V, the complete description of the NFNN controller is presented. Section VI concerns to the identification model. In section VII, the detailed design procedures of our proposed online controller are presented. Section VIII presents the simulation results of the NFNN controller that proves its performance. Section IX presents the overall field bus system setup and test. The last section is specialized with some conclusions, comments and future work.

II. CAN COMMUNICATION PROTOCOL

The CAN bus is a 2 wires serial bus with multi-master capability i.e. multiple nodes (slaves) connecting to a single 2-wires bus can communicate with each other. CAN implements a priority-based bus with a carrier sense multiple access with collision avoidance (CSMA/CA) as a medium access algorithm which is appropriate for sending and receiving real-time messages at baud rates up to 1Mb/s [11]. The arbitration mechanism ensures that one node will have access to the medium at any given time hence reducing collisions. In Ethernet, a device trying to transmit a frame will monitor the line and wait until the line is available then start transmitting.

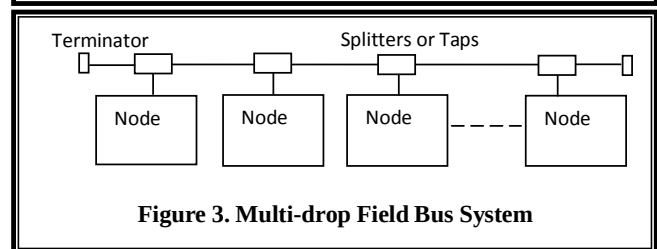
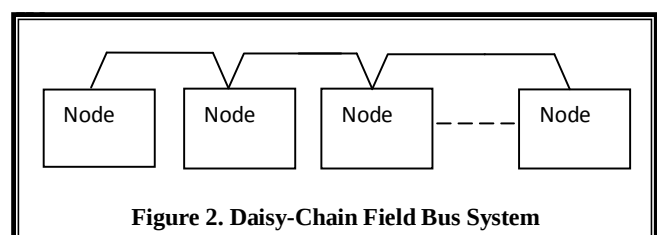
If 2 or more transmitters start, a collision will occur so all transmitters will stop. This explains why in most Ethernet LAN applications, nodes are connected point-to-point to a bridge or router. Errors in CAN protocol are detected using a CRC check. Therefore, the CAN bus provides a 100% data integrity. All nodes are synchronized on the same bit time period. The group delay cannot exceed a fraction of the bit time period which lead to the maximum throughput to be a function of the bus length e.g. 1 Mb/s is satisfied with 40 m bus length while 50Kb/s with up to 1km length [12].

Each message starts with an address field which is unique. So the arbitration will be completed at the end of address transmission and only one device will carry on with the message transmission. CAN protocol has two addressing schemes: CAN2.0A with 11 bit address and CAN2.0B with 29 bit address. CAN2.0A is used for small networks with 2048 different IDs but actually the number of nodes is limited by the type of the CAN transceiver used. The data field (DLC) in CAN2.0A is 8 bytes. The receiver drives the ACK bit. Bit stuffing provides a minimum density of edges to help devices remain synchronized. The CAN physical medium is a 2-wires bus (one can use UTP cable) terminated with a resistor. Differential signal is used for good immunity against noise.

A typical CAN field bus systems with master-slave mode suitable for industrial applications is shown in Figure 1[3].

III. DISTRIBUTED CONTROL AND NETWORKS

A DCS is a control system in which several controllers are operating simultaneously and cooperatively to perform the desired task. They are divided into two types: tightly-coupled multiprocessor systems sharing a common bus; and loosely-coupled systems where processor-based controllers are located at the points of control and exchange information via a network. There are many physical arrangements for field bus network some of them are: Daisy-chain directly connects the field device to the communication cable as shown in Figure 2; and Multi-drop which use a T-tab (stub) for device connection to the cable as shown in Figure 3. In our work, we used the daisy-chain configuration because it is the simplest.



V. HARDWARE COMPONENTS

Before describing the complete implementation of our system, the hardware components used in

the design should be explained in a little bit detail.

A. The CAN microcontroller (T89C51CC01)

The T89C51CC01 is the first member of the CANary family of 8-bit microcontrollers dedicated to CAN networking. It can reach to 20 MHz external clock rate. It is available in many different pin configurations: PLCC44, PLCC52, VQFP, etc. The detailed specification of this hardware is available in its data sheet [12]. It contains about 99 special function registers (SFR) including 34 CAN SFRs. The bit timing and baud rate for the CAN controller should be calculated and the corresponding registers should be downloaded with the calculated values. This is done through the software that the microcontroller should be programmed with. This software should also contain all the required functions to be performed by the slave controller.

B. The High Speed CAN Transceiver (MCP2551) [13]

The MCP2551 is a high speed CAN, fault-tolerant device that works as an interface between a CAN controller and the physical bus. It provides a differential transmit and receive capability for CAN controllers. Each node in a CAN system must have a device to convert the digital signals (Tx and Rx) generated by the CAN controller to differential output (CANH and CANL). It also works as a buffer between the CAN controller and the high voltage spikes that can be generated on the CAN bus by outside

sources. The CAN bus has two states: "Dominant", which occurs when the differential voltage is greater than a defined voltage; "Recessive",- occurs when the differential voltage is less than a defined voltage (typically 0). The two states correspond to the low and high state of the TXD input respectively. The low and high states of the RXD output correspond to the dominant and recessive states of the CAN bus respectively.

The MCP2551 output drives a minimum load of 45 ohms allowing a maximum number of nodes equal to 112 nodes. The termination resistor used here is 120 ohms. The high speed mode is selected by connecting the RS pin of the transceiver chip to Vss.

C. The Water Bath Temperature Control System

The experiment is done on a real water bath to control its temperature. The water bath is an example of an important component in a batch-reactor process. It can be described by the following model [14]:

$$G(s) = \frac{ke^{-t_d s}}{\tau s + 1} \quad \dots 1$$

Where k is magnification, τ is the time constant, and t_d is the delay time. In this paper, the water bath is chosen as the controlled object to study the temperature control system. In order to get the parameters of the plant, a step signal is applied to the input of the plant and the S-shape (process reaction) curve is obtained. From this curve we

got the values of k to be 1, t_d is 60 seconds, and τ is 180 seconds. The difference equation derived for this plant is found to be:

$$y(k) = \frac{\tau}{T_s + \tau} y(k-1) + \frac{kT_s}{T_s + \tau} u\left(k - \frac{t_d}{T_s}\right) \quad \dots 2$$

Where T_s is the sampling time that should be chosen accurately. In this paper, T_s is equal to 15 Sec.

VI. STRUCTURE AND LEARNING OF RNFPN

A. Structure Of RNFPN

The newly presented RNFPN controller for temperature control system is shown in Figure 4. The difference between this framework and that of the FNN is the transition layer. The operation of each layer of this RNFPN framework is introduced as follows:

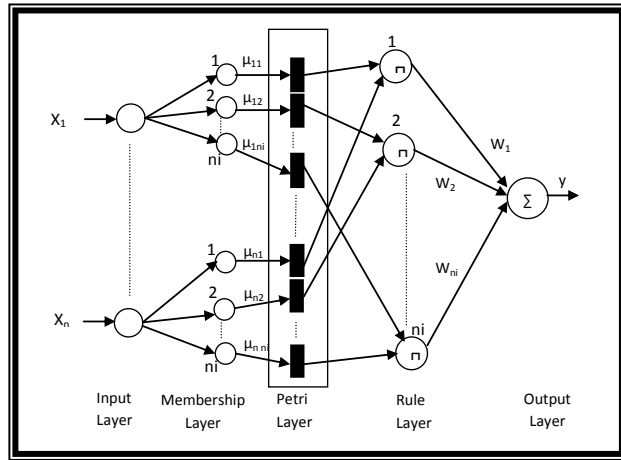


Figure 4: Framework of the RNFPN

1. Input Layer

Each node in this layer transmits the input crisp variable x_i ($i=1,2,3,\dots,n$) to the next layer.

2. Membership Layer

This layer represents the fuzzification stage for the RNFPN in which the crisp inputs x_i are transformed to fuzzy inputs via the adopted Gaussian membership function which is given by:

$$\mu_{ij} = \exp\left(-\frac{1}{2} \frac{(x_j - c_{ij})^2}{s_{ij}^2}\right) \quad \dots 3$$

Where c_{ij} and s_{ij} are the center and width of the membership function.

3. Petri Net Layer

This layer is used to produce tokens that make use of competition laws for node firing as follows:

$$t_{ij} = \begin{cases} 1 & \text{if } \mu_{ij} \geq d_{th} \\ 0 & \text{if } \mu_{ij} < d_{th} \end{cases} \quad \dots 4$$

Where t_{ij} is the transition and d_{th} is the dynamic threshold that varies with error and can be tuned by the following equation [15]:

$$d_{th} = \frac{\alpha \exp(-\beta E)}{1 + \exp(-\beta E)} \quad \dots 5$$

Where α and β are positive constants that can be chosen randomly. It is clear that the larger the error is, the smaller the threshold is. If the error becomes large, the threshold values will be decreased to fire more rules for the current situation. Of course one can use a constant value for the threshold. It is important to mention that if the threshold value is chosen to be 0, then the RNFPN system will transformed to FNN system.

4. Rule layer

The output of each node is the product of its inputs and it is given by:

$$\phi_j = \begin{cases} \prod_i^n \mu_{ij}; & \text{if } t_{ij} = 1 \\ 0; & \text{if } t_{ij} = 0 \end{cases} \quad \dots 6$$

Where ϕ_j is the output of the j^{th} node of the rule layer;

n is the number of crisp inputs.

5. Output Layer

The output node calculates the total output y as a summation of the input signals as follows:

$$y = \sum_j^{ni} w_j \phi_j \quad \dots 7$$

Where the connection weights w_j is the output action strength associated with the j^{th} rule; ni is the number of rules.

B. Learning Algorithm Of RNFPN

The task of constructing the RNFPN is divided into two subtasks: *structure learning* and *parameter learning*. In the structure learning, the number of fuzzy rules, initial location of membership functions, and initial consequent parameters are chosen. The parameter learning is used to tune the free parameters of the constructed network to its optimal values. To explain the learning algorithm of RNFPN using the supervised back propagation method, the error function is defined as:

$$E = \frac{1}{2} (y - y_p)^2 \quad \dots 8$$

Where y is the output of RNFPN network and y_p is the output of the plant.

The update laws for w_j , c_{ij} , and s_{ij} are[15]:

$$\begin{aligned} w_j(k+1) &= w_j(k) - \eta_w \frac{\partial E}{\partial w_j} \\ c_{ij}(k+1) &= c_{ij}(k) - \eta_c \frac{\partial E}{\partial c_{ij}} \\ s_{ij}(k+1) &= s_{ij}(k) - \eta_s \frac{\partial E}{\partial s_{ij}} \end{aligned} \quad \dots 9$$

Where η_w is the learning rate for the weights of layer,

η_c , η_s are the learning rates for the center and width of the Gaussian membership function respectively. Choosing suitable values for these learning rates is very important during training process. The three partial derivatives in the above three equations are derived and given by:

$$\frac{\partial E}{\partial w_j} = (y - y_p) \phi_j$$

$$\begin{aligned} \frac{\partial E}{\partial c_{ij}} &= \begin{cases} (y - y_p) w_j \phi_j \frac{(x_i - c_{ij})}{s_{ij}^2}, & \text{if } t_{ij} = 1 \\ 0, & \text{if } t_{ij} = 0 \end{cases} \quad \dots 10 \end{aligned}$$

$$\begin{aligned} \frac{\partial E}{\partial s_{ij}} &= \begin{cases} (y - y_p) w_j \phi_j \frac{(x_i - c_{ij})^2}{s_{ij}^3}, & \text{if } t_{ij} = 1 \\ 0, & \text{if } t_{ij} = 0 \end{cases} \end{aligned}$$

VII. IDENTIFICATION MODEL

For single input single output (SISO), linear, and time invariant plants with unknown parameters,

the identification model may be similar to the following [16]:

$$y_p(k+1) = \sum_{i=0}^{n-1} \alpha_i y_p(k-i) + \sum_{j=0}^{m-1} \beta_j u(k-j) \quad \dots 11$$

Where α_i and β_j are unknown parameters, u is the input to the plant, and y_p is the output. Two representations for the identification models are available: Parallel model and Series-Parallel model. In this paper, the second model is adopted.

Series-Parallel Identification Model

This model is obtained by feeding back the past values of the plant output as shown in Figure 5. The identification method depends on the control strategy. The forward control (indirect) strategy requires forward identification and the inverse control (direct) requires inverse identification.

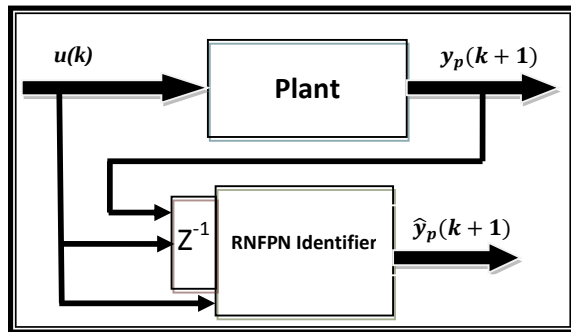


Figure 5: Series-Parallel Identification

In this paper, the forward control with forward identification is used because it is simpler and more accurate than the inverse control. The forward (offline) identification system for our plant is shown in Figure 6.

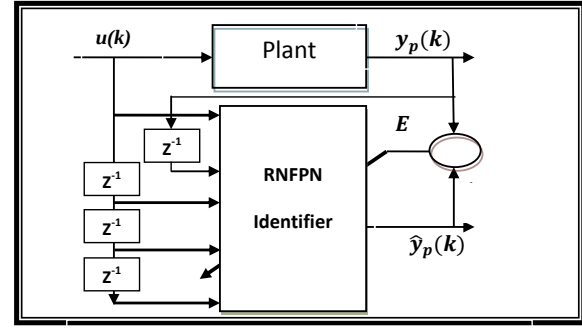


Figure 6: Forward Offline Identification ($t_d/T_s=3$)

In Figure 6, the value of (t_d/T_s) of the difference equation is 3 i.e. there is only $u(k)$, $u(k-1)$, $u(k-2)$, and $u(k-3)$ inputted to the identifier. Of course the number of delays (z^{-1}) depends on the transport lag of the plant.

VIII. CONTROL SYSTEM DESIGN

The PID controllers were used to control industrial temperature systems and some of these systems are still usable today. These systems were used as PID or PI controllers. Today PID controllers are modified to be fuzzy PID controllers. In general, PID controller is a direct controller placed in the feed forward path of the system, it receives the error signal and produces control command according to the value of the error signal, its derivative and its integral. Adaptive control is introduced to deal with complex systems that could not be controlled by conventional methods such as ill-defined plants or time varying parameters plants. In this type of control systems, plant response should follow the response of a defined system. Adaptive controllers could take many forms like Variable Structural Systems (VSS), Self Tuning Systems

(STS) and Model Reference Adaptive Control systems (MRAC). Any of these types of adaptive control systems may be used in the field of temperature control. In this work, the basic concept of PN is used with the FNN to create the presented RNFPN controller for any type of temperature control system. The proposed RNFPN forward (online) control system is shown in Figure 7. This control system, which is based on MRAC, performs two tasks: system identification and control. The initial values for w_j , c_{ij} , and s_{ij} of the identifier should be taken from offline learning that is achieved according to Figure 6. The purpose of the identification process is to overcome the changes that may occur in the system parameters due to disturbance. Also, the control process requires the sensitivity parameter (S) of the plant to be introduced by the identifier. The reference model specifies the desired output performance of the control system i.e. the system will track the desired output of the reference model. Therefore, it is designed as a PID closed loop system with the same plant to be controlled but with different parameters. The RNFPN for the controller and the identifier have the same structure with a small difference in the number and type of the inputs and outputs. The update training laws for the RNFPN is the same as the previously mentioned equations but the error function is given by:

$$Ec = \frac{1}{2} ec^2 = \frac{1}{2} (y_r - y_p)^2 \quad \dots 12$$

It is clear that y_r is the output of the reference model. Now, the partial derivatives $\frac{\partial E_c}{\partial w_{cj}}$, $\frac{\partial E_c}{\partial c_{cij}}$ and $\frac{\partial E_c}{\partial s_{cij}}$ of the updating laws should be derived according to the chain rule as follows:

$$\frac{\partial E_c}{\partial w_{cj}} = \frac{\partial E_c}{\partial e_c} \frac{\partial e_c}{\partial y_p} \frac{\partial y_p}{\partial m} \frac{\partial m}{\partial w_{cj}} \quad \dots 13$$

Where u is the plant input or the controller output. The term $S = \frac{\partial y_p}{\partial m}$ represents the system sensitivity that should be computed because the convergence of the RNFPN controller cannot be insured if S is unknown. Since the RNFPN identifier of Figure 7 is used to provide the S function and since it has the same plant input and output i.e.:

$$\frac{\partial y_p}{\partial m(k)} = \frac{\partial \hat{y}_p}{\partial m(k)} \quad \text{then:}$$

$$\frac{\partial \hat{y}_p}{\partial m(k)} = S = \frac{\partial \hat{y}_p}{\partial \phi_j} \frac{\partial \phi_j}{\partial \mu_{ij}} \frac{\partial \mu_{ij}}{\partial m(k)} \quad \dots 14$$

Hence, the S function is derived to be:

$$S = \begin{cases} -\sum_{i=1}^{n_i} w_j \phi_{cj} \frac{u(k) - c_{ik}}{S_{ik}^2}, & \text{if } t_{ij} = 1 \\ 0, & \text{if } t_{ij} = 0 \end{cases} \quad \dots 15$$

Therefore, we can summarize the updating laws for RNFPN controller by:

$$\frac{\partial E_c}{\partial w_{cj}} = ec S \phi_{cj}$$

$$\frac{\partial E_c}{\partial c_{cij}} = \begin{cases} \phi_{cj} (X_{ci} - c_{cij}) \frac{w_{cj} S ec}{S_{cij}^2} & \text{if } t_{ij} = 1 \\ 0 & \text{if } t_{ij} = 0 \end{cases} \quad \dots 17$$

$$\frac{\partial E_c}{\partial S_{cij}} = \begin{cases} \emptyset_{cj}(X_{ci} - C_{cij}^2) \frac{W_{cj} S_{ec}}{S_{cij}^3} & \text{if } t_{ij} = 1 \\ \mathbf{0} & \text{if } t_{ij} = 0 \end{cases}$$

In practice, the simple form of the PID algorithm, which depends on error, its derivative, and its integral, is not satisfactory and has to be modified to overcome its limitations [17]. The required modifications are: *Bumpless transfer*, and *saturation (integral windup)*.

A. Bumpless Transfer

In the steady state with zero error, the controlled variable is zero. In many practical applications this is not suitable e.g. the case of temperature control system which requires a non zero voltage to be applied to the heater input to provide heat output. The availability of the integral term may

lead to avoidance of the zero value for the controller output, but there will be difficulties in changing smoothly, without disturbance to the plant, from manual to automatic control. There will also be the danger that a large change will occur on changeover. Plant operating requirements demanded that manual/automatic changeover is done in the so-called “bumpless” manner [17]. Bumpless transfer can be satisfied by several methods. One of the most widely used methods is the velocity algorithm, which gives the change in the value of the manipulated variable Δm for each sample rather than the absolute value of the variable m . The difference equation for Δm is derived to be:

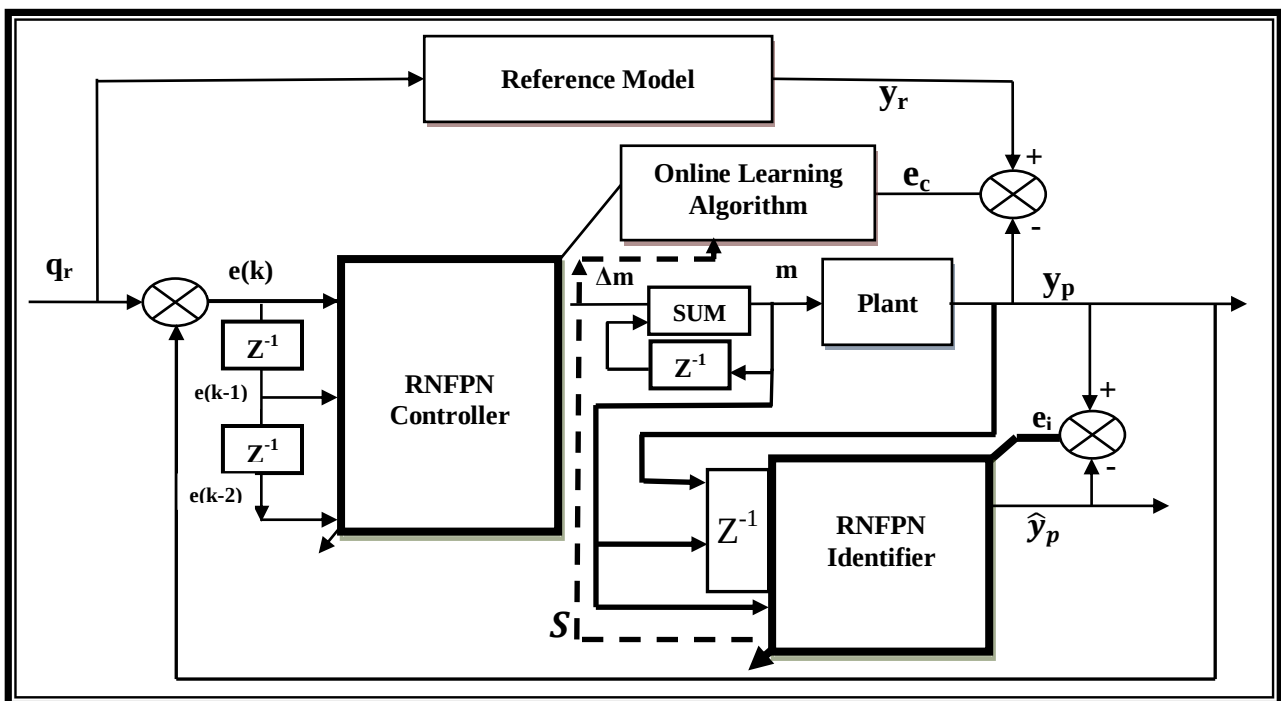


Figure 7: The proposed RNFPN forward (online) control

$$\Delta m(k) = k1 * e(k) + k2 * e(k-1) + k3 * e(k-2)$$

Where $e(k) = q_r(k) - y_p(k)$ is the error signal, $q_r(k)$ is the set commands, and $k1, k2, k3$ are constants to be tuned by the RNFPN controller. Hence, the inputs to the RNFPN controller are $e(k), e(k-1), e(k-2)$ as shown in Figure 7. Since it outputs only the change in the controller position, this

algorithm automatically satisfies “bumpless transfer”. Also, it is easy to program and safer in that large changes will not occur.

B. Saturation (Integral Windup)

The second major reason for using the velocity algorithm in our control system design is to overcome the problem of integral windup. In any practical application the value of the manipulated variable m is limited by physical constraints. An electric heater can supply only a given maximum heat and cannot supply a negative heat. If the value of the manipulated variable exceeds the maximum output of the actuator, effective feedback control is lost. Under some conditions, a large standing error in temperature will exist. If a PID controller is used, the integral term will continue to grow because there is a standing error. This effect is called integral windup which may lead to a poor response. The integral windup can be avoided by using the velocity algorithm since the integral action is obtained by a summation of the increments in the controller output (plant input). Therefore, there is an inherent integral

limit which prevents a buildup of error leading to avoiding the saturation problem.

IX. SIMULATION RESULTS FOR THE RNFPN

A. Offline Identification

The software written for the simulation results of the offline identification is an m-file. As mentioned before the results of the offline is taken to be the initial values for the online identification and control. The initial values will speed up the training process of the online control very much. The offline itself needs an initial values for the weights which are taken to be 0.002 and the center and width of the Gaussian membership functions which can be calculated from the following two equations respectively [18]:

$$c_{ij} = X_{n \max} - (i-1) \frac{X_{n \max} - X_{n \min}}{ni-1} \quad \dots 19$$

$$s_{ij} = 2 \frac{X_{n \max} - X_{n \min}}{ni-1}$$

Where $X_{n \max}$, $X_{n \min}$ are the pre-specified maximum and minimum values for the universe of discourse for each input to the controller network; ni is the number of membership functions for each input variable. The model to be identified is a type 0 system which represents a small water bath temperature control system. A velocity PID controller is designed to generate the appropriate voltage inputs $u(k)$ to the plant and the identifier as shown in Figure 6.

B. Online Identification and Control

The online forward control of Figure 7 is implemented by another m-file. The forward

identifier, which is used to estimate the plant sensitivity, will take its initial parameters from the final values of the offline identification system. All initial values for parameters such as: learning rates, number of rules, number of membership functions, constants of the dynamic threshold, etc. are chosen by trial and error for the online and offline identification. Figure 8 shows the trained response after few epochs while Figure 9 shows the result after 8000 epochs. The sum of square error (SSE) during the 8000 training epochs is drawn in Figure 10. It shows an average value for error of nearly 0.1, which is very good and acceptable in industrial applications. Of course, the value of SSE may be improved if the number of epochs is increased beyond 8000. In fact, we reach to 0.05 after about 16000 epochs then the SSE became fixed to this value. It is clear that these values (0.05 error in 16000 iterations) are much more better than the values obtained in [5] (3.1 error in 10^7 iterations). The value of error may change if some parameters are changed such as plant time constant, plant time delay, plant gain, learning rate, etc. If the values of weights and center and width of the membership functions become fixed during training i.e. there is no large change in their values then this is a good criterion for accuracy and convergence of the training process. Hence, Figures 11,12,13 show the norm for weight, center, and width. They show a very small change in their norm values, which means

that the training process is in the right direction and convergence is reached. Now, the proposed controller is ready to be used in real time water bath temperature control system.

X. PRACTICAL IMPLEMENTATION

The schematic diagram of our setup is shown in Figure 14.

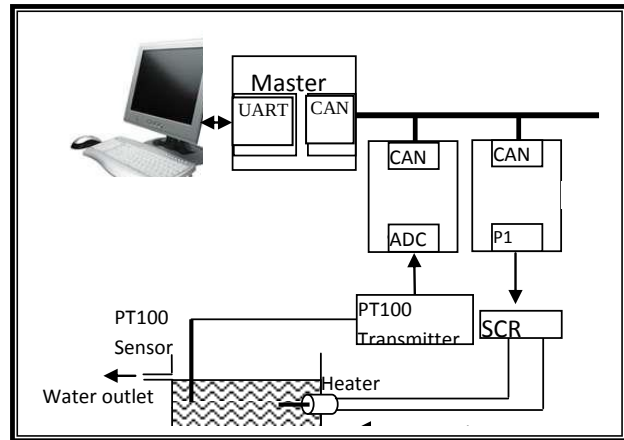


Figure 14: Water Bath Temperature Control System Based On Field Bus

One can use only one slave for inputting the sensor value and outputting the control signal to one of the ports of the microcontroller. This will reduce the cost and implementation effort. Note that the T89C51CC01 has an internal ADC components and some SFRs for doing that function.

The implementation process starts with the building of the master and slave controllers shown in Figure 15 and 16 respectively.

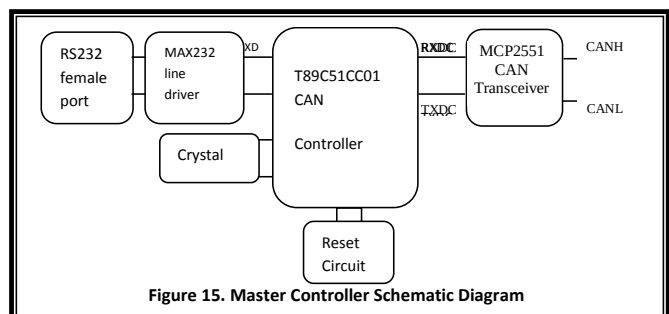
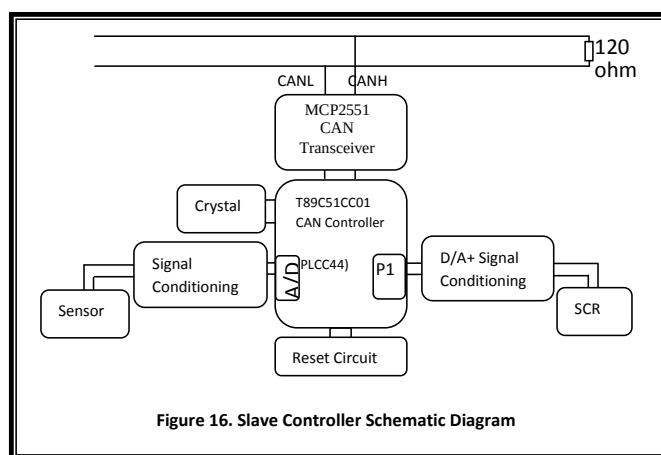


Figure 15. Master Controller Schematic Diagram



The output signal from the sensing element (PT100) is fed to the A/D converter input of the CAN controller via a signal conditioning circuit used for converting the current signal to 0~3V signal suitable for the A/D converter. The set point generation is done in the software driver (control program). The error signal is then transferred to the RNFPN controller and an output control signal is generated in accordance and sent via the master through the CAN bus to the slave. The signal is received by the CAN controller and output it to port 1. Then the data on port 1 is converted to analog by DAC0808 converter [19] then to the signal conditioning circuit that is responsible for producing a 4-20 mA actuation signal that should be fed to the SCR. This control signal is used to control the phase angle of the SCR leading to a variable incident driving voltage to the heater and hence a temperature control is satisfied. The max power of the heater used in this setup is 2000 W. The obtained response is shown in Figure 17 which

proves the performance of the implemented setup.

We clearly observe from Figure 17 that the steady state is achieved after approximately 2 minutes. This value is normal and accepted in such temperature control systems since these systems have a large inertia and time delay.

XI. CONCLUSIONS AND FUTURE WORK

In this paper, a general design methodology of a DCS system based on field Bus network protocol with CAN is presented. The design proved that the CAN protocol is a high reliable protocol and therefore can be adopted and highly recommended in industrial applications. As a case study, a temperature control system is designed and implemented. Because the temperature control system has characteristics of large time delay and large inertia, we need a high speed communication system to satisfy the real time requirements of the system. The overall system is implemented and controlled with a newly designed RNFPN controller designed with a velocity algorithm to satisfy the bumbless transfer and to avoid integral windup that may occur in real time. The response curve obtained is acceptable. There may be a development to this work which is the use of wireless sensor network (WSN) for data collection and monitoring. This may reduce the congestion on the field bus network i.e. the field bus will be used only for actuation leading to what is called wireless sensor/actuator networks (WSAN). WSAN may

be considered as a first gradual step towards building a wireless control system.

XII. REFERENCES

- [1] Web site: www.CERN.org.
- [2] F. M. Behno, "Design and Implementation of an Integrated Control System Using Field Bus (Control Center)", MSc Thesis, University of Technology, 2002.
- [3] M.N. Hanna, "Real-Time Analysis of FIP-based Systems", MSc Thesis, Cairo University, 2004.
- [4] Abraham, A., "Optimization of Evolutionary Neural Networks Using Hybrid Learning Algorithms", IEEE, Joint International Conference on Neural Networks, IEEE Press, Vol. 3, pp. 2797-2802, 2002.
- [5] Chia F. J, Jung S.C., "A Recurrent Neural Fuzzy Network Controller For A Temperature Control System", IEEE International Conference on Fuzzy Systems, pp. 408-413, 2003.
- [6] Tanomaru J., Omatu S., "Process Control by online Trained Neural Controllers", IEEE Transaction on Industrial Electronics, Vol. 39, No.6, pp. 511-521, Dec. 1992.
- [7] Lin C.T., Juang C.F., and Li C.P., "Temperature Control With a Neural Fuzzy Inference Network", IEEE Trans. Syst., Man, and Cyber., - Part C: Applications And Reviews, Vol.29, No.3, pp. 440-451, Aug., 1999.
- [8] Zhang J. and Morris A.J., "Recurrent Neuro-Fuzzy Networks For non-linear Process Modeling", IEEE Trans. Neural Networks, Vol.10, No.2, pp. 313-326, 1999.
- [9] Mastorocostas P.A., and Theocharis J.B., "A Recurrent Fuzzy-Neural Model For Dynamic System Identification", IEEE Trans. Syst., Man, and Cyber., - Part B: Cybernetics, Vol.32, No.2, pp. 176-190, 2002.
- [10] J. L. Peterson, "Petri Nets", ACM Computing Surveys, Vol.9, No. 3, pp. 221-252, 1977.
- [11] L.M. Pinho and F. Vasques," Reliable Real-time Communication in CAN Networks", IEEE Transaction on Computers, Vol. 52, No.12, December 2003.
- [12] "Enhanced 8-bit Microcontroller with CAN Controller and Flash memory (T89C51CC01), Atmel, www.atmel.com.
- [13] "High Speed CAN Transceiver MCP2551", Microchip Technology Inc., 2003.
- [14] Mohammed A.B. et al., "Design and Analysis of PI-Fuzzy Controller For Temperature Control System", Fourth Asia International Conference on Mathematical. Analytical Modeling and Computer Simulation, pp. 378-383, 2010.

- [15] Rong-Jong Wai, Chia-Ming Liu," Design of Dynamic Petri Recurrent Fuzzy Neural Network and Its Application to Path-Tracking Control of Nonholonomic Mobile Robot", IEEE transactions on Ind. Elec., Vol. 56, NO. 7, pp.2667-2683, July 2009.
- [16] Kumpati S. Narendra and Kannan Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Transactions on Neural Networks, Vol. 1, No. 1, March 1990.
- [17] S. Bennett, "Real-Time Computer Control: An Introduction", Prentice Hall International (UK) Ltd, First edition, 1988.
- [18] Rong-Jong Wai, Chia-Chin Chu," Robust Petri Fuzzy-Neural-Network Control for Linear Induction Motor Drive", IEEE trans. on Ind. Elect. , Vol. 54, No. 1, pp. 177-189, Feb. 2007.
- [19] "DAC0808 8-bit D/A Converter", National Semiconductor Corporation, 2001.

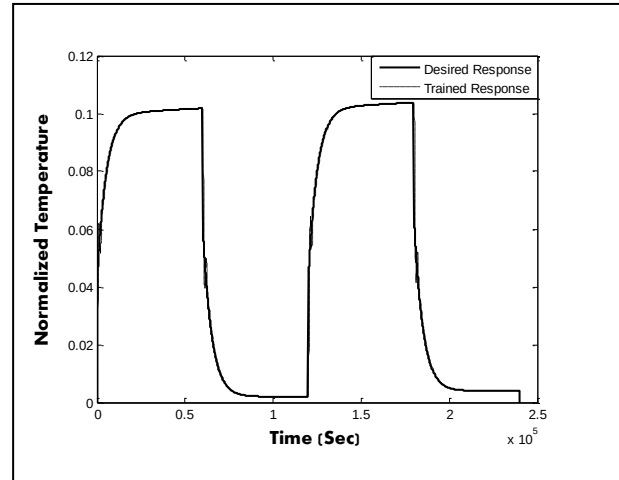


Figure 9: The Trained Response After 8000 Epochs For The Proposed Online Control System

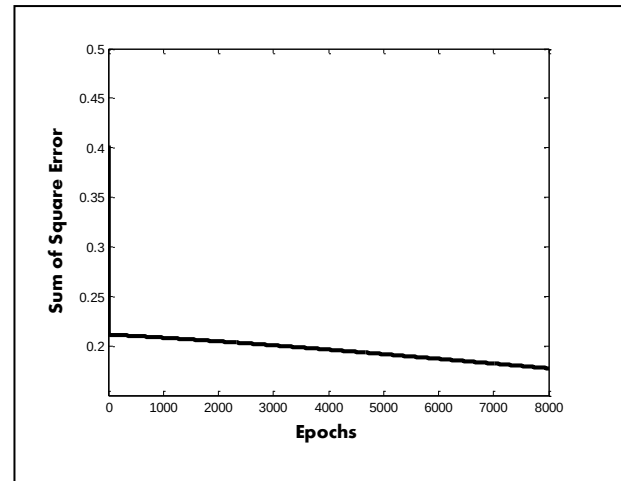


Figure 10: Sum of Square Error After 8000 Epochs For The Proposed Online Control System

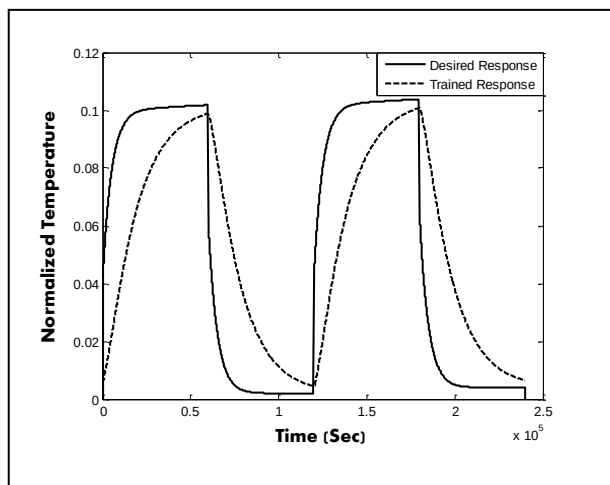


Figure 8: The Trained Response After Few Epochs For The Proposed Online Control System

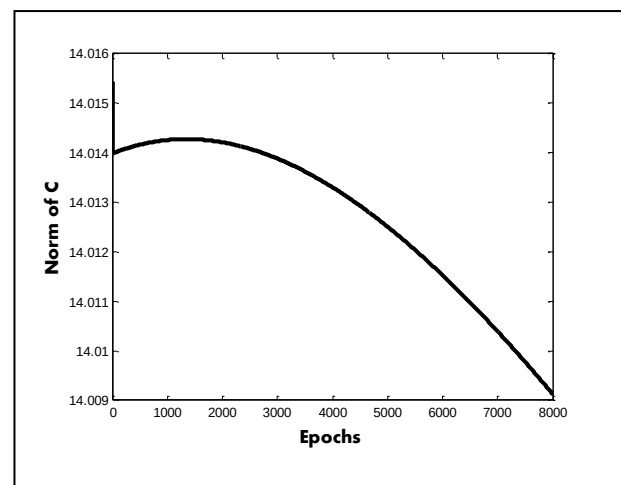


Figure 11: Norm of Center After 8000 Epochs For The Proposed Online Control System

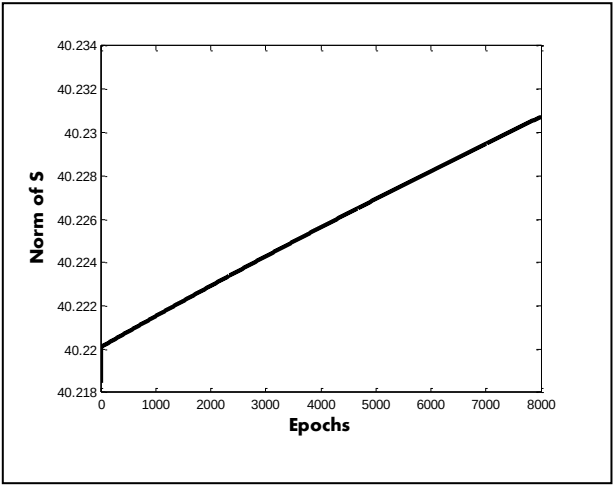


Figure 12: Norm of Width After 8000 Epochs For The Proposed Online Control System

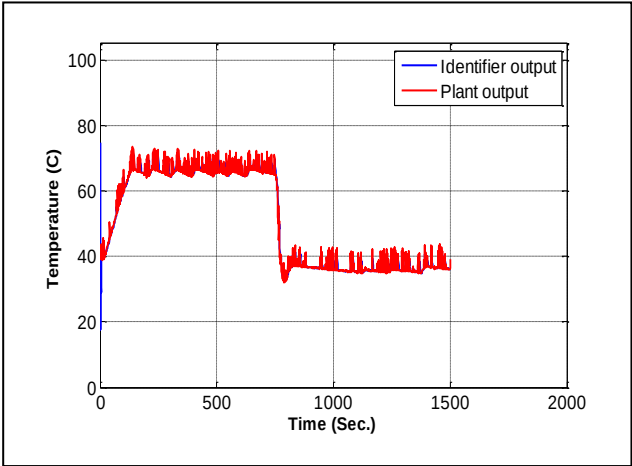


Figure 17. The measured response of the system

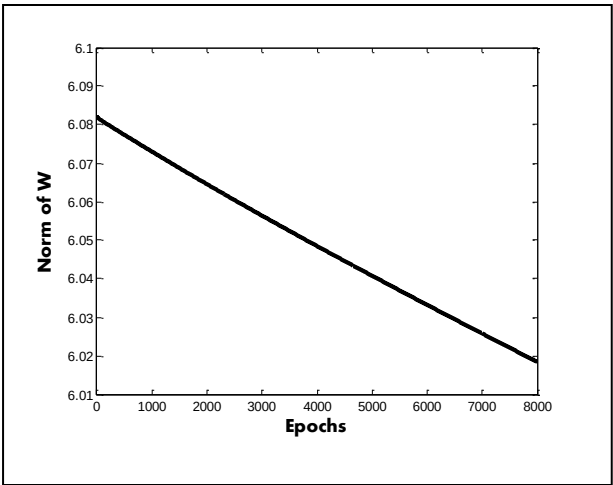


Figure 13: Norm of Weight After 8000 Epochs For The Proposed Online Control System

شبكة الناقل الحقلية المعتمدة على بروتوكول CAN مع مسيطر حرارة مبني

بخوارزمية Fuzzy-Neural-Petri Net

علي احمد عبد¹، عبد العظيم عبد الكريم علي²، علي فاضل مرهون³، نومان اسلام⁴

^{1,2,3} كلية الهندسة-جامعة البصرة-العراق

⁴ كلية الهندسة-جامعة دلهاوزي-كندا

الخلاصة

في شبكات الناقل الحقلية فان هناك مجموعة من الحساسات الذكية المربوطة مع بعضها وظيفياً والتي تتطلب مبادلة البيانات بشكل دقيق وضمن زمن محدد. في هذا البحث قدمت طريقة تنفيذ لنظام سيطرة موزع حقيقي لمنظومة حرارة. البرامج المضمنة للمسيطرات المكونة للشبكة تم كتابتها بلغة C. البروتوكول المعتمد للشبكة هو بروتوكول CAN ذات المواصفات العالية. المسيطر الرقمي الذي تم بناءه لمنظومة السيطرة يعتمد على مبدأ المنطق المضرب المعزز بميكانيكية الشبكات العصبية والتي تم اضافة طبقة جديدة لها هي طبقة ال-Petri. تم بناء هذا المسيطر بأسلوب السيطرة الامامي. ايضاً تم تعليم المسيطر اولاً كـ offline للحصول على بيانات يتم التدريب عليها بالنسبة للـ online controller. المنظومة بنيت عملياً وطبقت على منظومة حرارة (حمام ماء) صناعية لاثبات صحة العمل والنتائج.

The Predator-Prey Model Simulation

Taleb A. S. Obaid

Department of Computer Science, University of Basra, Basra, IRAQ.

e-mail: tasobaid@gmail.com

Abstract

Many of the most interesting dynamics in nature have to do with interactions between organisms. These interactions are often subtle, indirect and difficult to detect. Interactions in which one organism consumes all or part of another. This includes predator-prey, herbivore-plant, and parasite-host interactions. These linkages are the prime movers of energy through food chains. They are an important factor in the ecology of populations, determining mortality of prey and birth of new predators.

Mathematical models and logic suggests that a coupled system of predator and prey should cycle: predators increase when prey are abundant, prey are driven to low numbers by predation, the predators decline, and the prey recover, ad infinitum. One of such models that simulates predator-prey interactions is the Lotka-Volterra Model.

Keywords: Lotka-Volterra Model, Predator-prey interaction, Numerical solution, MATLAB

Introduction

A predator is an organism that eats another organism. The prey is the organism which the predator eats. Some examples of predator and prey are lion and zebra, bear and fish, and fox and rabbit. The words "predator" and "prey" are almost always used to mean only animals that eat animals, but the same concept also applies to plants: Bear and berry, rabbit and lettuce, grasshopper and leaf.

Predator and prey evolve together. The prey is part of the predator's environment,

and the predator dies if it does not get food, so it evolves whatever is necessary in order to eat the prey: speed, stealth, camouflage (to hide while approaching the prey), a good sense of smell, sight, or hearing (to find the prey), immunity to the prey's poison, poison (to kill the prey) the right kind of mouth parts or digestive system, etc. Likewise, the predator is part of the prey's environment, and the prey dies if it is eaten by the predator, so it evolves whatever is necessary to avoid being eaten: speed, camouflage (to hide from the

predator), a good sense of smell, sight, or hearing (to detect the predator), thorns, poison (to spray when approached or bitten), etc..

Predation is an important evolutionary force: natural selection favors more effective predators and more evasive prey. "Arms races" have been recorded in some snails,



Mathematical models of predation are amongst the oldest in ecology. The Italian mathematician Volterra is said to have developed his ideas about predation from watching the rise and fall of Adriatic fishing fleets. When fishing was good, the number of fishermen increased, drawn by the success of others. After a time, the fish declined, perhaps due to over-harvest, and then the number of fishermen also declined. After some time, the cycle repeated. A main purpose of modeling population interactions is to understand what causes such fluctuations. Indeed, the very first Lotka-Volterra system is the result of such an effort.

Predator Prey Model

The Lotka-Volterra equations describe an ecological predator-prey (or parasite-host)

which over time become more heavily armored prey, and their predators, crabs, which over time develop more massive claws with greater crushing power. Predation is widespread and easy to observe. Neither its existence nor its importance is in doubt.



model which assumes that, for a set of fixed positive constants **a** (the growth rate of prey), **b** (the rate at which predators destroy prey), **r** (the death rate of predators), and **c** (the rate at which predators increase by consuming prey). The following conditions will hold for our computer simulation model.

Let the prey population at time **t** be given by **y₁(t)**, and the predator population by **y₂(t)**. Assume that, in the absence of predators, the prey will grow exponentially according to **y'₁ = ay₁** for a certain **a > 0**. We also assume that the death rate of the prey due to interaction is proportional to **y₁(t) y₂(t)**, with a positive proportionality constant. So:

$$y'_1(t) = a y_1(t) - b y_1(t) y_2(t) \quad (1)$$

Without prey, predators will die exponentially according to $y_2'(t) = -r y_2(t)$ for a certain $r > 0$. Their birth strongly depends on both population sizes, so we finally find for a certain $c > 0$:

$$y_2'(t) = -r y_2(t) + c y_1(t) y_2(t) \quad (2)$$

These equations (1) and (2), lead to the following system differential equations:

$$\begin{cases} y_1'(t) = a y_1(t) - b y_1(t) y_2(t) \\ y_2'(t) = -r y_2(t) + c y_1(t) y_2(t) \end{cases}$$

We see that both $(e^{at}, 0)$ and $(0, e^{-ct})$ are solutions of $(y_1(t), y_2(t))$. From this system we find that for every solution we must have

$$y_1' \left(\frac{r}{y_1} - c \right) + y_2' \left(\frac{a}{y_2} - b \right) = 0$$

Integrating both sides gives us, for more details see [1].

$$r \log y_1(t) - c y_1(t) + a \log y_2(t) - b y_2(t) = \text{constant} \quad (3)$$

The Lotka-Volterra equations are a pair of first order, non-linear, differential equations that describe the dynamics of biological systems in which two species interact. Earliest predator-prey model based on sound mathematical principles forms the basis of many models used today in the analysis of population dynamics original form has problems.

Numerical Methods for Ordinary Differential Equation (ODE) in MATLAB

MATLAB is a technical computing environment for high-performance numeric (and not typically symbolic) computation and visualization. It is a proprietary software used

by researchers, educators, and students in industry and academia. GNU Octave* is an open source high-level language, primarily intended for numerical computations that is mostly compatible with MATLAB. For the purpose of these examples, all of the code and commands can be used in MATLAB or Octave and nearly identical results would be produced.

In addition to writing script files, we can create user-defined functions using m-files (also text files with the .m extension). MATLAB and Octave have an extensive library of mathematical functions built in, but there is often a need for a user to create their own functions. New functions may be added to the software "vocabulary" as function files.

MATLAB has a number of tools for numerically solving ordinary differential equations. ODE23 and ODE45 are functions for the numerical solution of ordinary differential equations provided in MATLAB. They employ variable step size Runge-Kutta integration methods. ode23 uses a simple 2nd and 3rd order pair of formulas for medium accuracy and ode45 uses a 4th and 5th order pair for higher accuracy. We will focus on one of its most rudimentary solvers, ode45, which implements a version of the Runge-Kutta 4th order algorithm.

* Octave is a high-level interactive language, primarily intended for numerical computations that is mostly compatible with Matlab. Octave uses the GNU readline library to handle reading and editing input.

In this study we will use the Matlab command ode45 to solve our systems of differential equation. This command is a robust implementation for systems of differential equations, which uses a variable step size method and the fourth and fifth order Runge-Kutta method.

Simulation of Predator-Prey in MATLAB

Continuous simulation concerns the modeling over time of a system by a representation in which state variables change continuously with respect to time. Typically, we use differential equations, that give relationships for the rates of change of the state variables with time.

In biological differential equations models, it is more common to have multiple dependent variables, and hence a system of two or more interrelated differential equations.

Solving a system of ODE in MATLAB is quite similar to solving a single equation, though since a system of equations cannot be defined as an inline function, we must define it as an M-file. We implemented this technique to solve the Lotka–Volterra predator–prey system:

$$\begin{aligned} \frac{dy_1}{dt} &= a y_1 - b y_1 y_2 ; & y_1(0) &= y_1^0 \\ \frac{dy_2}{dt} &= -r y_2 + c y_1 y_2 ; & y_2(0) &= y_2^0 \end{aligned}$$

with $a = .5471$, $b = .0281$, $r = .8439$, $c = .0266$, and $y_1^0 = 30$, $y_2^0 = 4$. (These values correspond with data collected by the Hudson Bay Company [1] between 1900 and 1920.)

The M-file lv.m:

```
function yprime = lv(t,y)
%LV: Contains Lotka-Volterra equations
a = .5471;b = .0281;c = .0266;r = .8439;
yprime = [a*y(1)-b*y(1)*y(2);-r*y(2)+c*y(1)*y(2)];
```

We can now solve the equations as follows:

```
>>[t,y]=ode45(@lv,[0 20],[30;4])
```

The output from this command consists of a column of times and a matrix of populations. The first column of the matrix corresponds with y_1 (prey in this case) and the second corresponds with y_2 (predators). Suppose we would like to plot the prey population as a function of time. In general, $y(n,m)$ corresponds with the entry of y at the intersection of row n and column m , so we can refer to the entire first column of y with $y(:, 1)$, where the colon means all entries. In this way, Figure 2 can be created with the command

```
>>plot(t,y(:,1), 'red',t,y(:,2),'.');
xlabel('Time, t');
ylabel('Population Sizes');
title('Our Predator Prey Example, Solutions Over Time');
legend('Prey','Predator')
```

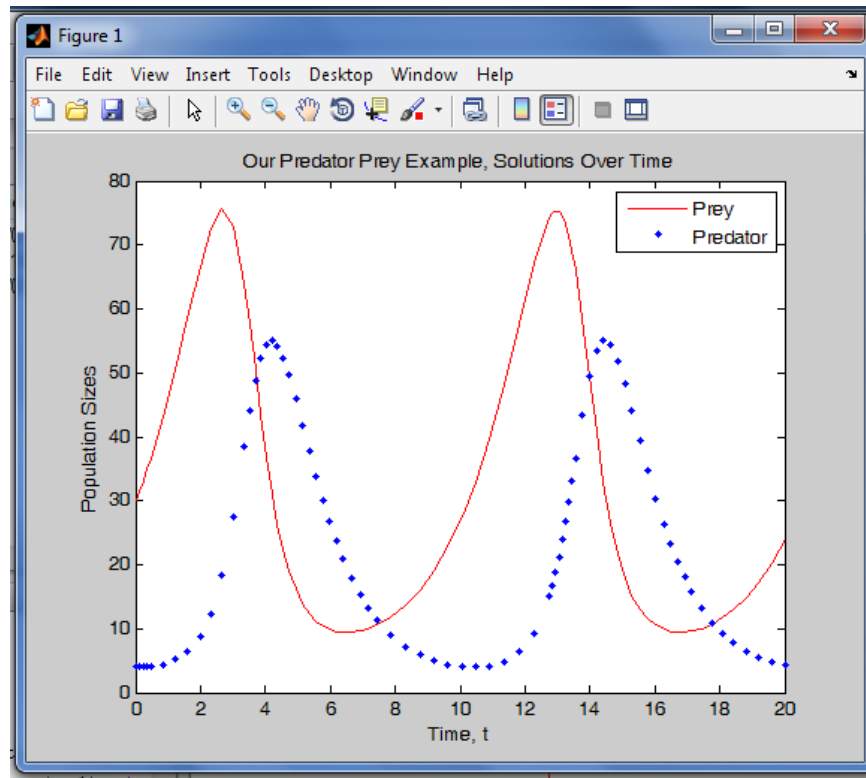


Figure 1: Plot of predator and prey populations for the Lotka–Volterra model

The functions y_1 and y_2 measure the sizes of the prey and predator populations respectively. The quadratic cross term accounts for the interactions between the species. Note that the prey population increases when there are no predators, but the predator population decreases when there are no prey.

To simulate a system, create a function that returns a column vector of state derivatives, given state and time values. For this example, we've created a file called LOTKA.M.

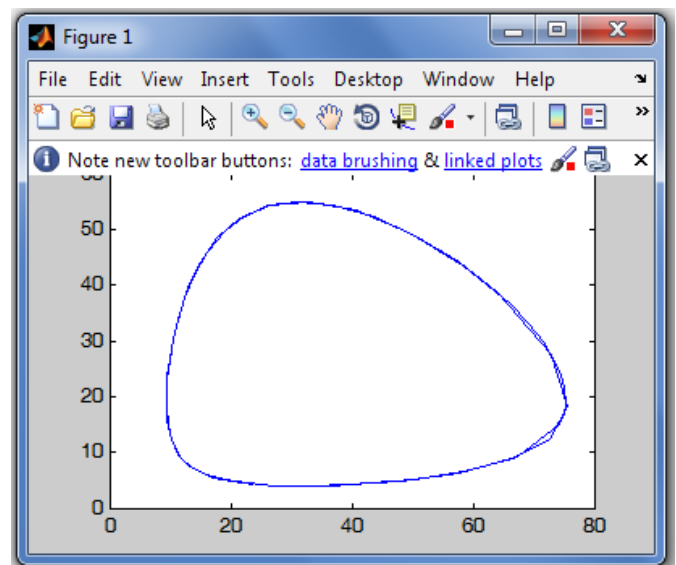


Figure 2, we can plot an integral curve in the phase plane with `>>plot(y(:,1),y(:,2))` which creates Figure 4. In this case, the integral curve is closed, and we see that the

populations are cyclic. (Such an integral curve is called a cycle.)

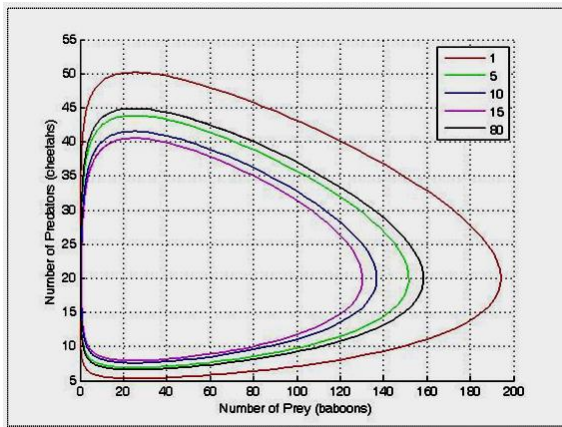


Figure 3, Prey-Predator dynamics as described by the level curves of a conserved quantity.

```
>> r = 1.3; s = .5; u = .7; v = 1.6; a = 1; K = 3;
h = .01;
>> P = .7; Q = 3; y1 = [P Q]; for t = 1:10000;
dP = (r*(1-P/K)-(s*Q)/(a+P))*P*h;
dQ = (-u+(v*P)/(a+P))*Q*h; P = P+dP; Q =
Q+dQ; y1 = [y1; P Q]; end;
>> P = 2; Q = 6; y2 = [P Q]; for t = 1:10000;
dP = (r*(1-P/K)-(s*Q)/(a+P))*P*h;
dQ = (-u+(v*P)/(a+P))*Q*h; P = P+dP; Q =
Q+dQ; y2 = [y2; P Q]; end;
>> P = 0:.1:3; nullP = (r/s)*(1-P/K).*(a+P);
>>
plot(y1(:,1),y1(:,2),y2(:,1),y2(:,2),P,nullP,[a*
u/(v-u), a*u/(v-u)],[0 8])
>> % phase diagram with two trajectories
```

Discussion

We have discussed continuous systems whose process of evolution depends on differential equations given as a nonlinear differential equations system. Such a system contains a number of parameters that must be estimated accordingly to some exist studies.

In this field, (for instance, the $a, b, c, d > 0$ in the predator-prey model). Usually point estimates are calculated and used in the model. These estimates typically have uncertainty associated with them.

Solutions of which are plotted figure 1 where prey are shown in red (solid), and predators in blue (dashed). In this sort of model, the prey curve always lead the predator curve. Figure 1 shows a periodic activity generated by the Predator-Prey model.

The level curves of the figure 2, are closed so that the solution is periodic. It is desirable that the numerical solution of (3) is also periodic, but this is not always the case. Note that the prey population increases when there are no predators, but the predator population decreases when there are no prey

Conclusion and Further Work

Consider an environment consisting of two populations, predators and prey. We are interested in both the predator and prey population size. However, these populations interact. The Lotka-Volterra equations describe an ecological predator-prey (or parasite-host) model. The Lotka—Volterra model is sensitive to modifications in the differential equations that describe the dynamics of the nonlinear system.

We can incorporate uncertainty in our differential equations. This is done by using fuzzy numbers as estimates of the unknown parameters. Finite-difference algorithms for studying the dynamics of spatially extended

predator-prey interactions would be very excited and effective.

References

1. Wouter Duivesteijn, "Continuous Simulation", "Information Sciences Special Issue: Advances in Fuzzy Logic", June, 2006.
2. Freedman, H.I., Deterministic Mathematical Models in Population Ecology, Marcel Dekker, (1980)
3. Hoppensteadt, F., "Predator-prey model" ([http://www.scholarpedia.org/article/](http://www.scholarpedia.org/article/Predator-prey_model)
4. J.D. Murray. Mathematical Biology I: An Introduction. Springer-Verlag, 2003
5. J.J. Buckley, Fuzzy Statistics, Springer-Verlag, Heidelberg, Germany, 2004.
6. Marcus R. Garvie, Finite-Difference Schemes for Reaction-Diffusion Equations Modeling Predator-Prey Interactions in MATLAB, Bulletin of Mathematical Biology (2007) 69: 931-956

الخلاصة

النماذج الديناميكية الأكثر إثارة للاهتمام في الطبيعة هي النماذج التي لها علاقات تفاعلية بين الكائنات الحية. هذه التفاعلات غالبا ما تكون خفية وغير المباشرة ويصعب اكتشافها. في مثل هذه العلاقات فان احد كائنات يستهلك كل أو جزء من الكائن الاخر. ويشمل هذا المفترس والفريسة ، الحيوانات المقتاته على الاعشاب، وكذلك علاقة الطفيلي و المضيف. هذه الروابط هي المحرك الرئيسي للطاقة من خلال السلاسل الغذائية. فهي تشكل عاملا مهما في البيئة الطبيعية للسكان، وتحديد معدل وفيات المواليد وفريسة جديدة من الحيوانات المفترسة.

النماذج الرياضية والمنطق توحى بوجود نظام معتمد من جانب الحيوان المفترس والفريسة بشكل دورة. فمثلا تزداد الحيوانات المفترسة عند وفرة الفرائس، تنخفض اعداد الفرائس بسبب الافتراس، وانخفاض الحيوانات المفترسة، وبذلك يزداد عدد الفرائس، وهكذا الى مالا نهاية. احد هذه النماذج التفاعلية التي تحاكي المفترس والفريسة هو نموذج لوتكا-فولتيرا.

Classification of GIS Image using GLCM and Neural Network

Tawfiq A. Alasadi

Babylon University

Dean of Computer Technology College

Wadhah R. Baiee

Babylon University

College of Science ,Computer Department

Abstract:

GIS can hold agricultural regions data like forest, fruit covered lands and/or cultivate lands, these lands have been managed inside GIS by receiving a selected region remotely sensed image, so GIS users must have an appropriate digital map that represents these lands each one according to its owner, status, and some other data. Normally, in such system, these lands will be classified by the users according to agricultural status depending on human vision. So, hardly to the users to classify these lands manually, and this become a great problem which take a long time depending on human efforts, especially if there is a huge number of lands. The suggested study creates a new ArcMap GIS tool which classifies these given lands automatically. Thus, this tool runs the developed system application; it will gather required information for each one of selected land, by sampling sub-images from their centers depending on the digital map, and gathers related status information from attribute database. On the next stage, the system will extract a vector of textural features for each one of the selected lands from their image samples using second order statistics Gray Level Co-occurrence Matrix (GLCM) and calculate eight textural features for each one of three visible bands (RGB) for each land sample. That vector of features will become the input to supervised multi-layer perceptron with backpropagation neural network classifier which be learned depending on recommended GIS user training data set. As a result the system has accuracy near to 75%; these results were achieved by comparing the classification results from system test trials with desired user classification data.

Introduction:

Geospatial data has both spatial and thematic components. Conceptually, geographic data can be broken up in two elements:

observation or entity and attribute or variable.

GIS have to be able to manage both elements.

Spatial component, the observations have two aspects in its localization, absolute

localization based in a coordinates system and topological relationship referred. A GIS is able to manage both while computer assisted cartography packages only manage the absolute one. The aim of classification is to link each object or pixel in the study area to one or more elements of a user-defined label set, so that the radiometric information contained in the image is converted to thematic information. The process can be regarded as a mapping function that constructs a linkage between the raw data and the user-defined label set. Two types of classification are commonly performed. Supervised classification methods which are based upon prior knowledge of the number and certain aspects of the statistical nature of the spectral classes with which the pixels making up an image are to be identified, and unsupervised classification methods which are performed by running a classification algorithm without any predefinition of spectral classes of interest. Texture is also an innate property of objects. It contains important information about the structural arrangement of surfaces. The use of texture in addition to spectral features for image classification might be expected to result in some level of accuracy improvement, depending on the spatial resolution of the sensor and the size of the homogeneous area being classified. R. Methre et al. [38] investigated the texture retrieval using combination of local features of Haralick

derived from one level discrete wavelet transform coefficients and global statistical features computed from three level wavelet transformed images. Y. Zhang et al. [44] proposed a hybrid classifier for polarimetric SAR images, the feature sets consist of span image, the H/A/ α decomposition algorithm, and the GLCM-based texture features. Then, a probabilistic neural network (PNN) was adopted for classification. E. S. Flores et al [11] used GIS techniques to improve the classification capabilities of a feature extraction algorithm for land use/cover change detection in a deciduous forest environment.

Proposed System Environment:

For programming issues, the proposed system application was programmed using C#.NET. The agricultural lands classification tool which is the proposed system application tool, it was added to ArcMap GIS as a new GIS dll this file contains the application classes code for the proposed system and also, the application interface windows codes to be run into ArcMap GIS software, it become a portable application; it can be used by any machine which has installed ArcMap GIS software.

The agricultural lands had been covered by polygons vector layer previously. These lands represented now by real world satellite image in the raster layer. The user entered only 20 lands information to the database according to his classification decisions to be the

recommended data set for learning purpose, and the remaining lands without information; the system will classified them later.

Information Gathering from GIS database:

The system will clip a colored image with three bands from underneath the polygon to be the raster image sample of that polygon (region) and will go to the textural feature extraction stage, we proposed in this research to clip an image window of size equal to (100 * 100) pixels for each studied lands. The sample is clipped from the center of the

related polygon. The information from database is required for the classification learning reason, so this information is located in attribute database table. Collected values represent the agriculture status for each land, this value will become the classifier learning desired output for each related extracted land object. Now we have gathered information for each studied land represented by (Land ID, Image Sample, and Current Status). Figure (1) illustrates the block diagram of the proposed system .

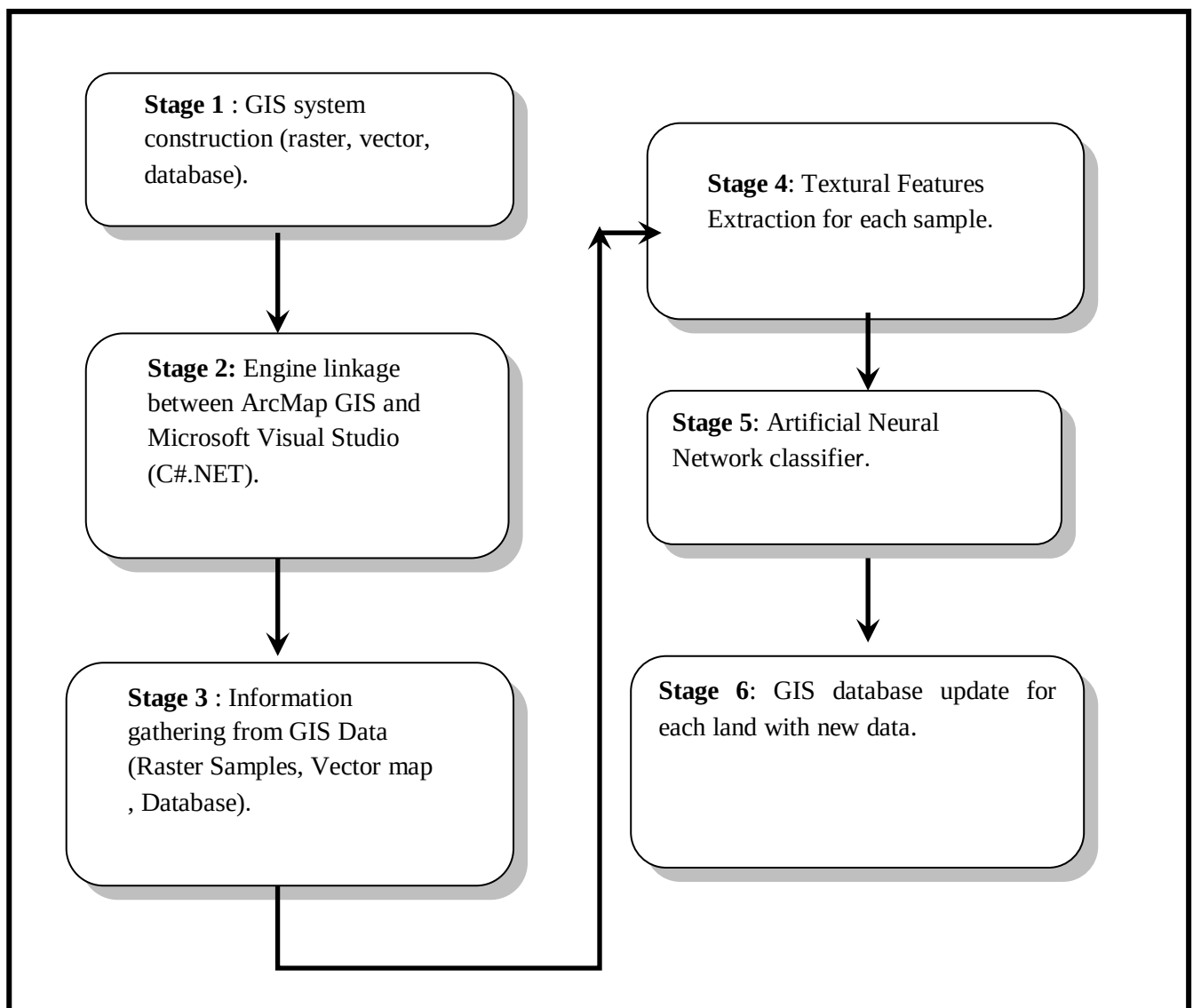


Figure (1) : Block diagram of the proposed system.

GLCM and Textural Features:

Texture feature extraction technique based on the gray-level co-occurrence matrix (GLCM), sometimes called the gray-tone spatial-dependency matrix. The principal concept of GLCM is that the texture information contained in an image is defined by the adjacency relationships that the gray tones in an image have to one another. The matrix element $P(i, j | d, \theta)$ contains the second order statistical probability values for changes

between gray levels i and j at a particular displacement distance d and at a particular angle θ . Instead of using the frequency values in a GLCM directly, it is common practice to normalize them to the range $[0, 1]$ to avoid scaling effects.

Eight textural features was extracted in proposed system for each visible band from each land sample to be the vector of features which represented the entire land sample.

Mathematically, those features can be represented as follows:

Angular Second Moment

$$ASM = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{P(i, j)\}^2 \dots \dots \dots (Eq. 1)$$

Contrast

$$CONTRAST = \sum_{n=0}^{G-1} n^2 \left\{ \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P(i, j) \right\}, \quad |i - j| = n \dots \dots \dots (Eq. 2)$$

Inverse Difference Moment

$$IDM = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{1}{1+(i-j)^2} P(i, j) \dots \dots \dots (Eq. 3)$$

Entropy

$$ENTROPY = - \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P(i, j) \times \log(P(i, j)) \dots \dots \dots (Eq. 4)$$

Correlation

$$CORRELATION = \frac{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i - \mu_i)(j - \mu_j)}{\sigma_i \sigma_j} \dots \dots \dots (Eq. 5)$$

Cluster Shade :

$$SHADE = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{i + j - \mu_i - \mu_j\}^3 \times P(i, j) \dots \dots \dots (Eq. 6)$$

Cluster Prominence :

$$PROMINENCE = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{i + j - \mu_i - \mu_j\}^4 \times P(i, j) \dots \dots \dots (Eq. 7)$$

Haralick Correlation

$$HARRALICK = \frac{\sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (ij) P(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y} \dots \dots \dots (Eq. 8)$$

ANN Classifier:

Each land object has its own vector of textural features that describes the properties of object. These features become the input to the ANN classifier. The neural network which be used in the system is multilayer perceptron with supervised backpropagation algorithm. ANN classifier consists of three main layers (Input layer, one Hidden layer, and Output layer), it can be shown as 24|52|2 completely

Test and Results:

The system collects lands information from GIS. The collected data are saved in system database buffer for 240 lands which covered by polygons in GIS pre-built user system. These lands data are also sampled, and the system calculated their textural features and save them in system database buffer and they were ready to test by the classifier. Here we describe the practical way which be used to choose the appropriate parameters to be applied on system classifier raising the

connected net. Input layer has 24 input nodes each node take one feature of the textural features contained in input vector.

Hidden Layer has 52 hidden nodes (This number is practically chosen by made many training trials on system classifier even to reach to highest system accuracy). Output nodes have 2 nodes (Selected lands has 4 classes (types)). 20 recommended samples were chosen to learn the classifier.

accuracy of classification. All trials have been applied on the system classifier using different parameters with each trial will be list, these parameters are the ANN parameters such as : momentum , learning rate, sigmoid alpha value , error limit , number of hidden layer's nodes, threshold value, and the angel of the GLCM of related textural features..Table (1) lists the extracted texture features from one land, and table (2) lists all trials and their specifications.

Table (1): Example of 24 features extraction for one selected land sample.

Features / Bands(Angle)	Red 90	Green 90	Blue 90
Entropy	0.5604390	0.5069230	0.5701910
Correlation	0.9642070	0.9657530	0.9599290
Haralick	0.0227320	0.0147750	0.0159920
IDM	0.4424280	0.5288610	0.4242930
Cluster Shade	0.0001710	0.0001110	0.0001130
Cluster Prominence	0.0004850	0.0005730	0.0003210
ASM	0.0005730	0.0003210	0.0006050
Contrast	0.0006370	0.0010780	0.0005210

Table (2): Training-Test trials for ANN classifier.

Trial No.	Momentum	Learning Rate	Sigmoid alpha	Error Limit	Iteration	Threshold	Hidden Nodes	GLCM angle	Accuracy%
1	0	0.1	2	0.1	1128	0.5	12	90	62.10045662
2	0	0.1	2	0.1	1189	0.5	6	90	65.75342466
3	0	0.1	2	0.1	1117	0.5	20	90	65.29680365
4	0	0.1	3	0.1	1117	0.9	20	90	69.8630137
5	0.1	0.1	2	0.01	4334	0.9	12	90	68.03652968
6	0.1	0.01	2	0.01	38754	0.9	12	90	73.05936073
7	0.1	0.01	2	0.01	38754	0.5	12	90	70.77625571
8	0.1	0.01	2	0.01	40212	0.9	20	90	72.60273973
9	0.1	0.01	2	0.01	38094	0.9	20	0	66.21004566
10	0.1	0.01	2	0.01	40028	0.9	20	45	66.66666667
11	0.1	0.01	2	0.01	40321	0.9	20	135	63.47031963
12	0.1	0.01	2	0.01	40321	0.5	20	135	57.99086758
13	0.2	0.01	2	0.01	42066	0.9	30	90	71.68949772
14	0.3	0.01	2	0.01	49432	0.9	30	90	72.60273973
15	0.05	0.01	2	0.01	36323	0.9	30	90	73.05936073
16	0.1	0.005	2	0.001	424922	0.9	20	90	73.05936073
17	0.1	0.01	5	0.01	6530	0.9	12	90	70.77625571
18	0.1	0.01	2	0.01	41190	0.9	12	90	71.68949772
19	0.1	0.01	2	0.01	36299	0.9	100	90	72.60273973
20	0.001	0.02	2	0.01	17928	0.9	12	90	71.68949772
21	0.0001	0.01	2	0.01	35815	0.9	12	90	70.3196347
22	0.5	0.01	2	0.01	78009	0.9	12	90	69.40639269
23	0.1	0.01	2	0.01	37887	0.9	12	90	73.05936073
24	0.1	0.01	2	0.01	41724	0.9	10	90	70.77625571
25	0.1	0.01	2	0.01	41593	0.9	9	90	69.8630137
26	0.1	0.01	2	0.01	38362	0.9	15	90	72.60273973
27	0.1	0.01	2	0.01	38963	0.9	13	90	73.51598174
28	0.1	0.02	2	0.01	19982	0.9	13	90	71.23287671
29	0.1	0.005	2	0.01	75138	0.9	13	90	72.14611872
30	0.1	0.005	2	0.01	85791	0.9	15	90	69.8630137
31	0.1	0.01	2	0.01	37677	0.9	50	90	71.68949772
32	0.1	0.01	2	0.01	37677	0.9	50	90	71.68949772
33	0.1	0.01	2	0.01	37677	0.8	50	90	74.42922374
34	0.1	0.01	2	0.01	37677	0.7	50	90	72.14611872
35	0.1	0.01	2	0.01	37677	0.81	50	90	74.42922374
36	0.1	0.01	2	0.01	37677	0.82	50	90	73.97260274
37	0.1	0.02	2	0.01	18447	0.8	50	90	73.97260274
38	0.1	0.01	2	0.01	37091	0.8	52	90	74.93771732

The test with following data was chosen and produced system accuracy $\cong 75\%$: Momentum = 0.1, learning rate = 0.01 , sigmoid alpha value = 2 , error limit = 0.01 , number of hidden layer's nodes = 52 , threshold value = 0.8 , and the angel of the GLCM = 90° , learned with 37091 iterations.

These values with heist accuracy and low iteration of ANN learning is taken from table (1) above. Table (3) illustrates a comparison details between the classified types and shows the identical accuracy of each land type with user classification decisions.

Table (3): Results accuracy for each land type.

User Classification	Proposed System Classification						Accuracy%
	Land Type	not used	houses	used	trees	total	
	not used	45	6	10	2	63	71.42857143
	houses	5	36	0	2	43	83.72093023
	used	12	1	66	7	86	76.74418605
	trees	0	6	3	19	28	67.85714286
	total	62	49	79	30	220	74.9377

System accuracy is 74.94% compared with user classification data, these ratios were recommended by the test trials table, it is a higher accuracy test ratio among the others and depend it as a recommended results. And the details of each type are shown in table (3) above, the system has a higher accuracy 83.72% with houses covered lands ,and 76.44% with used lands , and 71.43% with

not used lands , finally, 67.86% with trees lands. Figure (2) illustrates a compression between system classification on studied lands and the user classification, where figure (2.a) shows classified lands after apply the proposed system on the test lands and picture (2.b) shows manually user classification on same selected lands.



(a)

(b)

Figure (2): (a) System Classification , (b) User Classification.

Conclusions:

The classification method that used in the proposed system doesn't depend on the pixel color values for classification reasons, instead; GIS data (raster, vector, and database) is appointed all together to complete the classification requirements. Using textural features that extracted from grey level co-occurrence matrix for each land samples as the input to the supervised neural network classifier is more efficient than using classical way depending on individual pixel color values only. GLCM with angel 90° is the best among GLCMs with other angels (0° , 45° , 135°) for extraction features to be used later in supervised classification with multilayer perceptron with backpropagation classifier.

References:

1. A. K. Sinha et al., "Fuzzy Neural Network Modeling of Land-Use /Cover using IRS-1D Satellite Image", Map India Conference, 2004.
2. A. M. Coleman, "An Adaptive Landscape Classification Procedure Using Geoinformatics and Artificial Neural Networks", MSc. Thesis; Faculty of Earth and Life Sciences Vrije University, Amsterdam The Netherlands, 2008.
3. B. Booth et al., "Getting Started with ArcGIS", Environmental Systems Research Institute, 2001.
4. B. Tso et al., "Classification Methods for Remotely Sensed Data", Taylor & Francis Group, 2009.

5. C. H. Chen et al., "The Handbook of Pattern Recognition and Computer Vision", World Scientific Publishing Co., 1998.
6. C. Tucker, "ESRI using Arctoolbox" , Environmental Systems Research Institute, 2000.
7. D. Decker, "GIS Data Sources", John Wiley & Sons, 2001.
8. D. I. Verbila, "Practical GIS analysis", Taylor & Francis Group, 2003.
9. D. L. Fitzgerald, "Landing Site Selection for UAV Forced Landing Using Machine Vision", PhD Thesis; School of Engineering Systems; Queensland's University of Technology; Australia, 2007.
10. E. Gose et al., "Pattern Recognition and Image Analysis", Prantice Hall, 1997.
11. E. S. Flores et al., "GIS Improved Object-Based Classification For Land Use/Cover Change Detection In Human Altere Deciduouds Forest", The American Society for Photogrammetry & Remote Sensing 2009 Annual Conference; Baltimore; Maryland, 2009.
12. E. Stefanakis et al., "Geographic Hypermedia: Concepts and Systems", Springer, 2006.
13. E. W. Tennant, "Using ArcGIS to Create Living Documents with Archaeological Data", MSc. Thesis; Michigan Technology University, 2005.
14. F. Aguera et al., "Using Texture Analysis to Improve Per-pixel Classification of Very Hhigh Resolution Images for Mapping Plastic Greenhouses", ISPRS Journal of International Society for Photogrammetry & Remote Sensing (63) 635–646 , 2008.
15. F. Albregtsen, "Statistical Texture Measures Computed from Gray Level Co-occurrence Matrices ", University of Oslo, 1995.
16. F. Bachmann et al., "An Architecture Based on Class Dependent Neural Networks for Object-based Classification", Institute for Mine-Surveying and Geodesy, 2009.
17. F. Harvey, "A primer of GIS fundamental geographic and cartographic concepts", The Guilford Press, 2008.
18. G. E. Sherman, "Desktop GIS: Mapping the Planet with Open Source Tools", The Pragmatic Bookshelf, 2008.
19. G. H. Landeweerd et al., "The Use of Nuclear Texture Parameters in The Automatic Analysis of Leukocytes", Pattern Recognition; Vol. 10; Pages (PP) 57-61, 1978.
20. I. N. Ahmed, "Digital Image Classification", MSc Thesis; College of Scince; University of Babylon, 1999.
21. I. T. Service, "Introduction to GIS using ArcGIS", Durham University, 2006.
22. J. A. Anderson, "An Introduction to Neural Networks", PHI Learning Private Limited, 2009.
23. J. L. Harris, "A Hierarchial Document Description and Comparition Method", Msc. Thesis; Purdue University, 2003.
24. K. C. Clearke et al., "Geographic Information System and Environmental Modeling", Prentice Hall, 2009.

25. K. T. Changg, "Programming ArcObjects with VBA", Taylor & Francis Group, 2008.
26. L. Fauset, "Fundamentals of Neural Networks: Architectures Algorithms and Applications", Prantice Hall, 1994.
27. M. Cheong et al., "An Approach to Texture-Based Image Recognition by Deconstructing Multispectral Co-occurrence Matrices using Tchebichef Orthogonal Polynomials ", IEEE, 2008 .
28. M. F. Insana et al., "Analysis of Ultrasound Image Texture via Generalized Rician Statistics," Optical Engineering , 1986.
29. M. S. Nixon et al., "Feature Extraction and Image Processing", Elsevier, 2008.
30. N. R. Chrisman, "Exploring Geographic Information Systems", John Wiley and Sons, 1997.
31. O. Dassau et al., "GIS for Educators Topic 3: Vector Attribute Data", Spatial Planning & Information; Department of Land Affairs; Eastern Cape; South Africa (DLA), 2009.
32. O. Engler et al., "Introduction to Texture Analysis", Taylor & Francis Group, 2010.
33. P. Croswell, "GIS Design and Implementation Services", IT Consultants—GIS Design and Implementation Services Brochure, 2010.
34. P. Ohanian et al., " Performance Evaluation for Four Classes of Textural Features", Pattern Recognition 25; 819-8133, 1992.
35. R. C. Ghonzaliz et al., "Digital Image Processing", Prentice Hall, 2000.
36. R. Kok et al., "Analysis of Image Objects from VHR Imagery for Forest GIS Updating in The Bravian Alps", ISPRS, Vol. XXXIII; Amsterdam, 2000.
37. R. M. Haralick et al., "Textural Features for Image Classification", IEEE Transiction on Systems Vol. SMC-3 No. 6, 1973.
38. R. Methre et al., "Exploring Spatial Information in Spectral Features for Texture Image Retrieval", International Journal of Computer and Network Security Vol. 1; No. 3, 2009.
39. S. Arivazhagan et al., "Texture Classification Using Wavelet Transforms.", Pattern Recognition Letters 24; 2003
40. S. Carver, "Innovations in GIS", Taylor & Francis e-Library, 2005.
41. S. Haykin, "Neural Networks and Learning Machines", PHI Learning Private Limited, 2009.
42. S. K. Shah et al., "Image Classification Based on Textural Features using Artificial Neural Network", IE (I) Journal ET, 2004.
43. Y. C. Chen et al., "Texture Features for Classification of Ultrasonic Liver Images", IEEE ransactions on Medical; Vol.11; No. 2; Pages 141-151, 1992 .
44. Y. Zhang et al., "Remote-Sensing Image Classification Based on an Improved Probabilistic Neural Network", Sensors 9; pages 7516-7539, 2009.

Enhancing Intrusion Detection System Using Distributed Environment

Safaa O. Almamory,

Wesam S. Bhaya,

Anees M. Hadi

College of Computer Technology, College of Computer Technology , College of Science for woman,

University of Babylon,

University of Babylon,

University of Babylon

safaa_vb@yahoo.com

wesambhaya@yahoo.com

anees_alhamzawi@yahoo.com

07813554590

07801672425

07801342940

Abstract:

The study of Intrusion Detection System (IDS) has become an important aspect of network security. As soon as the IDS detects a set of attacks it will generate many alerts referring to as security breaches. Snort IDS is an open source IDS which is suffer from flooding vulnerability if it will be used with fast network. To solve this problem, we suggest using several Snort sensors with one packet classifier. Each one of these sensors contains a part of the overall signatures. A packet classifier is built to distribute the flow over these Snort sensors, and to avoid spread of the same attack over more than one sensor and then avoid detection. The experimental results, on DARPA 1999 data set, have shown that the efficiency of Snort is enhanced by about 30%.

1. Introduction

An intrusion is an active sequence of related events that deliberately try to cause harm, such as rendering system unusable, accessing unauthorized information, or manipulating such information. There are two types of intrusion systems:

Intrusion Detection System (IDS), and

Intrusion Prevention System (IPS). IDS is simply trying to detect the unauthorized penetration by the intruder in the network. The basic process for the IDS is to passively collects data, Preprocesses and classifies them. There are two IDS types Host-based Intrusion-Detection System (HIDS), Network-based Intrusion-Detection System (NIDS). IPS sits

inline on the network and monitors it. It takes action based on prescribed rules when an event occurs.

IPS has different functionalities and strengths from the IDS. Also, IPS is considered as a blocking product (proactive) while the IDS, is considered as an alerting product (reactive) [1]. There are some of IDS open source softwares (or called network sensors); like Advanced Intrusion Detection Environment (AIDE) [2], Open Source Host-based Intrusion Detection System (OSSEC) [3], Suricata which is the Host-based intrusion-detection and prevention system. [4], Prelude Hybrid IDS which is host-based and network-based IDS [5], Bro NIDS [6], and Snort is the network-based (IPS/IDS). Snort is combining the benefits of signature, protocol, and it is the most widely deployed IDS/IPS technology worldwide [7].

The main aim of this paper is to improve the Snort IDS sensor work by finding new methodology that will permit to process more packets than Snort in normal case.

We will build classifier able to sniff packets and classify them according to its protocols using suitable packet classification algorithm.

These protocols refer to different TCP/IP layers protocols. After packet classifying according to its protocols, the packets will forwarding to different Snort sensor nodes for processing.

This paper is organized as follows. Section 2 surveys the related works.

Section 3 provides preliminaries about the proposed system like Snort IDS and packet classification. Section 4 proposes our system. The results and experiments are presented in section 5. Finally, Section 6 concludes this paper.

2. Related Work

Several researchers tried to enhance the work of Snort which includes Snort rules, Snort components, content matching, compilation, distributed systems, parallel systems, and other fields. Wheeler P. *et al.* [8] approach proposes to use a distributed system where in which each node has the same rules. The packets are sent to each node using load balancer. Thus, each node will receive $1/N$ from original load, where N is the number of nodes.

Kopek C. *et al.* [9] approach proposed that the packets are divided into fragments and forwarded to the array of processors. Each fragment could process independently from other fragments. Match-bit field is assigned to each packet, which allows one processor to quickly indicate to other processors that a match-bit field has been found for a given packet.

When a notification is received, the remaining processors can start inspecting another packet.

Sanz-Bobi M. A. *et al.* [10] approach

proposes an intelligent system for automatic detection of intrusions in computer networks (Intrusion Detection System based on Artificial Intelligence IDSAI). Its architecture is based on multi-agent system in which several types of agents cooperate together to perform a fast and reliable detection of intrusions.

Schuff D. L. *et al.* [11] approach bases on Snort 2.6 version. It only executes multiple instance of original Snort in parallel and use load balancer that distributes task queues to dispatch traffic. After distributing tasks by load balancer, each processing thread does a job, such as decode, preprocess, and detect.

Yoshioka A. *et al.* [12] Approach generates a hash value using five protocol fields for each rule. The five protocol fields are source IP, destination IP, protocol type, source port, and destination port. Each hash value is stored in a single hash table. When a packet arrives, the same hash function is applied to the incoming packet and the hash value generated from the packet is used to search in the hash table for the matching rules. This approach searches for the matching rules multiple times against each packet to cover all possibilities.

3. Preliminaries

This section provides an introduction about snort NIDS (Subsection 3.1) and packet classification which appears in Subsection 3.2.

3.1 Snort as Intrusion Detection System

Snort logically consists of five components. These components work together to detect particular attacks and to generate output in a required format from the detection system [13]. The first component is the packet decoder which takes packets from network interface via the libpcap library, and determines which protocol is in use for a given packet. As a preprocessors plug-ins component, this component can be used with Snort to arrange or to modify data packets before the detection engine does some operation to find out if the packet is used by an intruder.

The third component is the detection engine. The packets are sent to detection engine by the decoder and preprocessor plug-ins components. These packets are compared against the Snort rules; if a packet matches any rule, the output plug-ins will activate, otherwise, the packet will be dropped. The detection plug-ins, as the fourth component, provides additional detection functions on the packets. Finally, there are many outputs plug-ins used by Snort to save generated output (logs/alerts) from the previous components.

Several reasons encouraged us to select Snort NIDS. First, Snort is one of the more mature open source packages, which is free of charge. Second, there is an ability to program and reconfigure this software. Third, bugs for Snort are virtually nonexistent due to

periodic industry maintains and support. Fourth, the preprocessors arrange or modify the network data flow in real time to increase the chances of a signature noticing a malicious packet. Fifth, Snort is very flexible in the ways it can be used with network as a packet sniffer, packet logger, or as IDS. Finally, industry support where the Snort open source community keeps Snort signatures up to date in order to identify the new attacks.

The NIDS can be evaded by flooding it using noise or dummy traffic. The flooding technique requires the NIDS to utilize valuable CPU and memory resources to analyze the dummy traffic. If the performance of the NIDS is not sufficient to handle the traffic generated by flooding, the actual intrusive activity performed by the intruder may be missed, due to the excessive dummy traffic generated. Even if the real attack is detected, the NIDS response may be significantly delayed due to the exhaustion of CPU and memory resources on the NIDS [14]. In this paper, we will increase the efficiency of Snort IDS to avoid the flooding problem.

3.2 Packet Classification

Packet classification is the process of categorizing the packets according to its header fields. This process is applied in the forwarding machine (like router) to identify the context of the packets and to perform important actions.

The criteria for classifying packet is called rule R , and the set of finite rules $R_1, R_2, \dots, R_n$ contained in forwarding machine is called rule database or a classifier [15]. The fields of rule and packet header are related; for example, the rules that implement IPv4 consist of 5 fields (source IP address, destination IP address, protocol type, source port, and destination port). The incoming packet to router matches specific rule if the distinct fields in the packet match the corresponding filed in that rule [16].

In this paper, we have used a packet classification algorithm which uses hash tables. The reason for choosing hash table based algorithm is that we have a small number of rules; in addition, it has time complexity $O(1)$.

4. Proposed System

As we said above, the main idea of this paper is to use packet classification algorithm to enhance processing speed of Snort IDS. This is done by distribute the packet flow over a set of Snort sensors. Figure 1 depicts the main components of the proposed system. The packet classifier online receives the packet flow and distributes the flow over snort sensors. These sensors monitor the flow for any suspicious activity and trigger an alert for every detected attack. The triggered alerts by these sensors will go to a central database which is monitored by the security analyst.

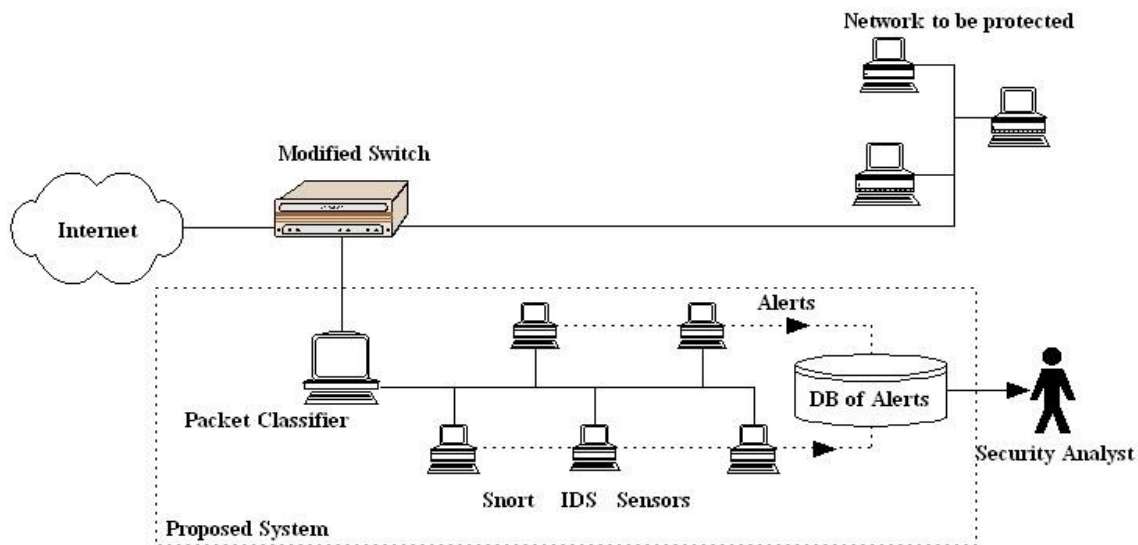


Figure 1: main components of the proposed system. The packet classifier online receives the packets and distributes the flow over snort sensors.

One critical point about the classifier is how to classify packets and guarantee that the packets of the same attack go to the same sensor. For this reason, we clustered the signatures by the type of protocol and distribute these signatures over Snort sensors. Every Snort sensor has to monitor partial set of the whole rules.

Algorithm 1 demonstrates the main steps of the proposed system. It receives a real-time

packet flow and produces a set of alerts. It composed of two parts. The first part (Lines 2-6) is related to packet classifier. The classifier decodes every packet to extract protocol type. According to protocol type, the classifier forwards the packet to the suitable Snort sensor. The second part of the algorithm (Lines 7-10) demonstrates the work of every Snort sensor.

Algorithm 1: Classification for detection Input: a real-time sequence of packets

Output: a sequence of alerts

Begin

1. duplicate network traffic and forward it to the packet classifier;
- // Packet classifier
2. foreach packet PKT received do{
3. P=protocol type extracted from PKT header;
4. using hash table to find IP address of snort sensor_i monitoring P;

```

5.      modify PKT header with new destination IP address;
6.      forward PKT to snort sensori;}
// Snort sensori; it works in parallel with packet classifier
7. foreach PKT received do{
8.      foreach S ∈ partial signature seti do {
9.          if PKT contains S then
10.             report alert;}}
End

```

5. Results

The experiments were implemented in academic CISCO lab at the University of Babylon under Linux operating system. Snort IDS software version 2.8.5.1 had been used in the experiments. The number of rules that we use are 3306 rules. In addition, we used DARPA datasets [17] to evaluate our system in real-time. We used this dataset to measure the load on CPU, RAM, and disk swapping for the Snort in a normal case and for the classifier that we built.

The proposed system is divided into two stages, the first stage is concerned with how to distribute Snort rules over some nodes, and the second stage is concerned with how to test that system; these two stages are stated in the following two subsections, respectively.

5.1 Snort Rules' Distribution

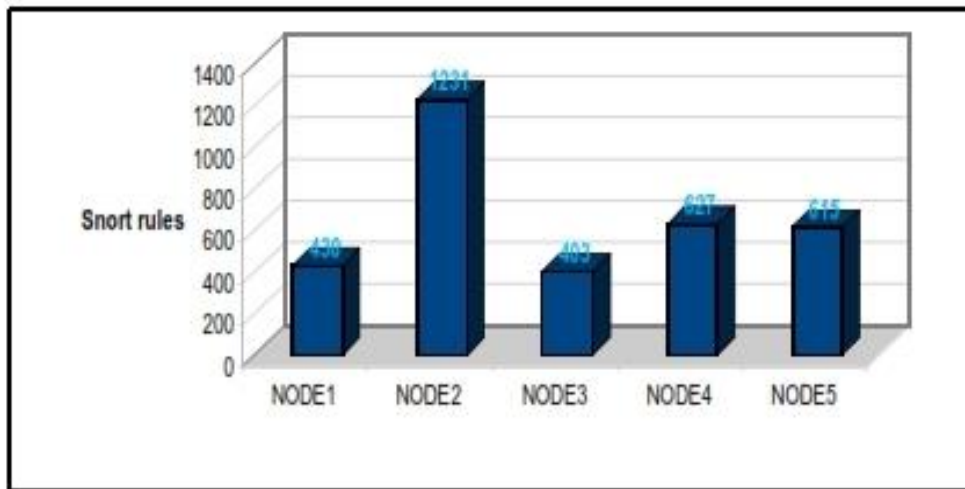
The DARPA 1999 [17] is an off-line dataset to evaluate the intrusions detection system. These dataset includes data for five weeks. Three weeks of

training data with background traffic and labeled attacks and two weeks of test data with background traffic and unlabeled attacks. A rich variety of background traffic that looks as if it were initiated by hundreds of users on thousands of hosts is generated in the test bed.

Instead of using one Snort IDS sensor with complete rules, we used five Snort sensor nodes with partial rule-set. In this case, we have to distribute Snort rules in order to process the network traffic in parallel fashion. Table 1 shows the protocols distribution over the five sensor nodes, whereas Figure 2 show as the Snort rules distribution according to its protocols for system to sensor nodes.

Table 1: Rules distribution according to its protocols.

Sensor node <i>j</i>	Monitored protocol(s)
1	IP,UDP,ICMP
2	HTTP
3	SMTP, ORACLE
4	FTP, NETBIOS, SQL, MYSQL, EPMAP, MICROSOFT DS, SSH
5	TELNET, POP, TIME, IRC(CHAT), IDENT, FINGER, SNMP, IMAP, DNS, NNTP, Others Prot.

**Figure 2: The suggested Snort rules' distribution.**

In Figure 2, we show that the Node2 has the largest amount of Snort rules because of its protocol distribution in DARPA 1999 dataset. In spite of the fact that node2 has the most number of rules, these rules should be placed in one sensor for detection purposes.

5.2 Resources Consumption Experiments

We use the TCPREPLAY program for replaying the DARPA 1999 dataset. Because of the large size of inside.tcpdump file, it takes very long time in replaying. Thus, we replayed it with 10MB/Sec because this speed is considered as a suitable speed since it is not

lost in packets or slowly in packets read, and the time required to replay inside.tcpdump file is about from 240 to 250 second.

At same time we replayed the dataset, we had computed the load on the PC resources using ATOP benchmark software. We use ATOP program to calculate the consumption on the CPU, memory, and disk swapping on a Linux system. By applying ATOP command line with `-C` option, we computed the average CPU consumption for the Snort in normal case, and the classifier as shown in Figure 3.

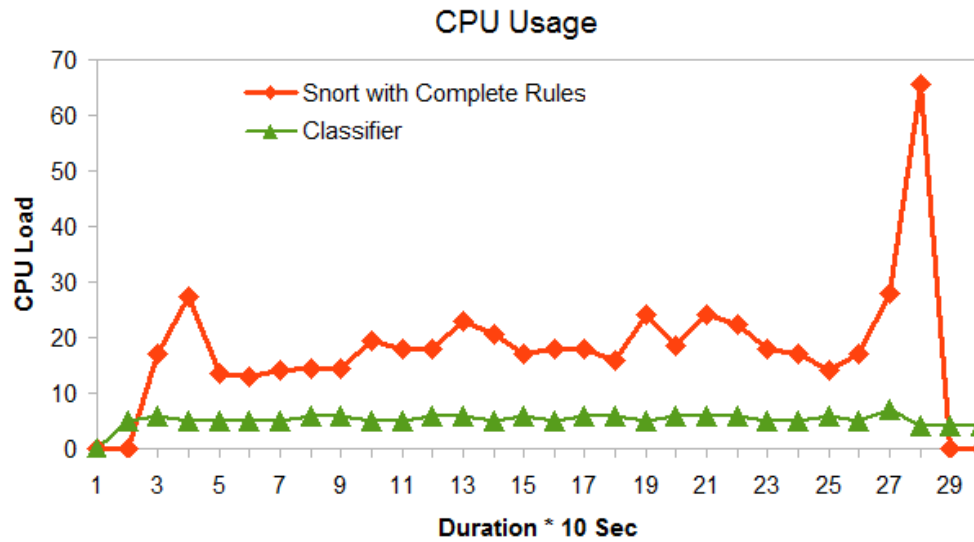


Figure 3: CPU usage for the classifiers compared with the Snort in normal case.

In the Figure 3, we note that the classifier has less CPU consumption than the Snort IDS in normal case (Snort sensor that have the complete rule- set). It should be noted that Snort IDS in normal case suffered from

flooding in the 290 second and became out of service.

Figure 4 explains the CPU consumption on sensor nodes which belong to our system.

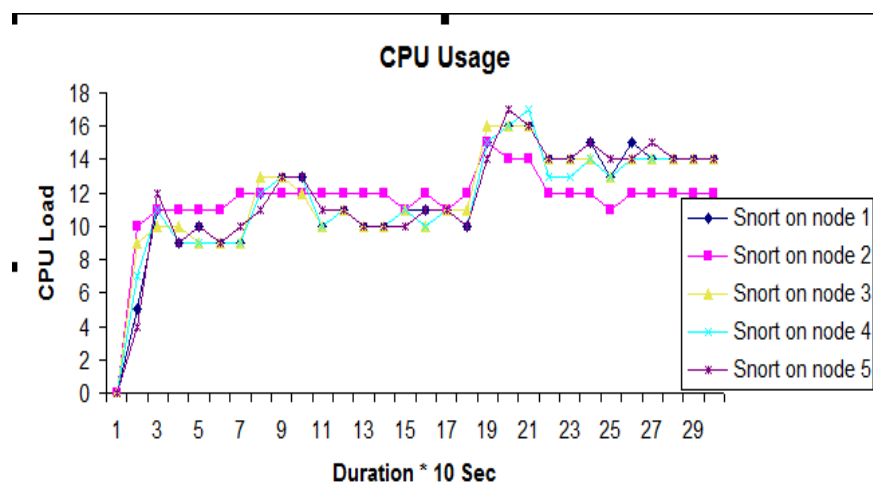


Figure 4: CPU consumption for the sensor nodes associated with our system.

By ATOP command line with -M option, we computed the average memory consumption

for the Snort in normal case, and the classifier as shown in Figure 5.

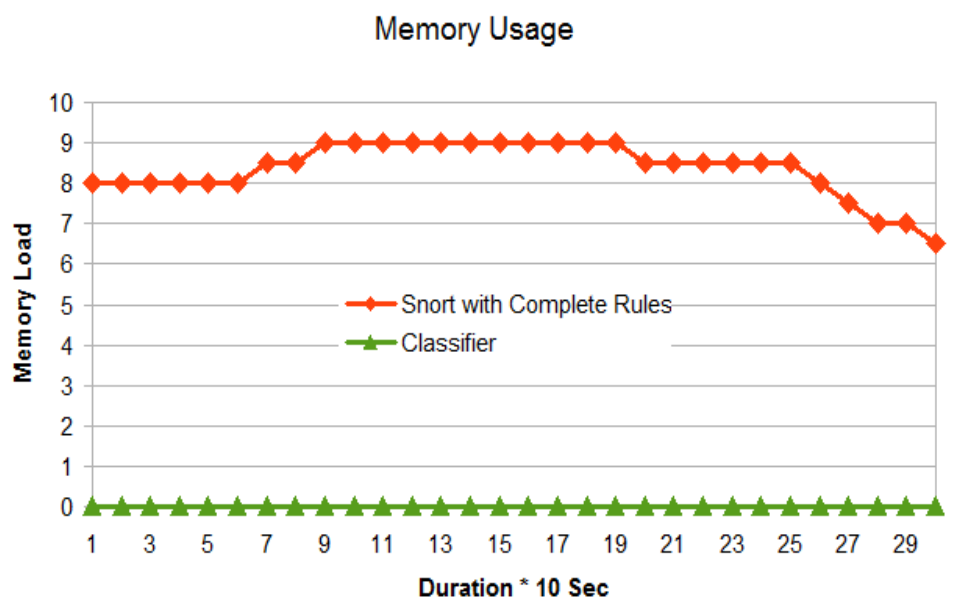


Figure 5: Memory usage for the classifier compared with the Snort in normal case

By applying the ATOP command line with `-D` option, we computed the average disk swapping

consumption for the Snort in normal case, and the classifier as shown in Figure 6.

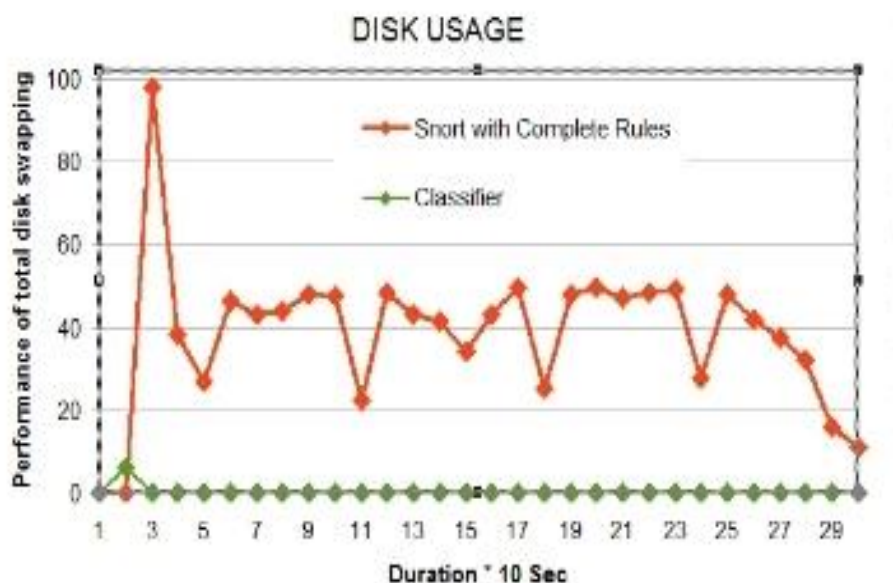


Figure 6: Disk usage for the classifiers compared with Snort in normal case

Table 2 explains the packets throughput by the Snort IDS in normal case compared with the classifier with different dataset replay speed as shown in Table 2.

Table 2: Average number of packets processed by the classifiers compared with Snort in normal case

No.	Software	Dataset file replaying speed	Average number of packet processed
1	Snort in normal case	10Mbps	418
		92Mbps	3731
3	Classifier	10Mbps	4370
		92Mbps	9832

6. Conclusions

Several researchers have worked on enhancing Snort IDS using different techniques. However, that process is a trade off between speed and detection accuracy. The aim of this paper is to improve efficiency of Snort IDS by distributing Snort rule-set over five Snort sensors and a classifier. We have concluded that the efficiency of the Snort IDS has been increased by about 30% after applying our system on DARPA 1999 dataset. A little increasing in efficiency is expected when applying the proposed system with real packet flow.

References

- [1] Endorf C., Schultz E., and Mellander J., "Intrusion Detection & Prevention", McGraw-Hill, 2004.
- [2] Advanced Intrusion Detection Environment (AIDE), [aide. sourceforge. net], 2011.
- [3] Ossec IDS, www.ossec.net, 2011.
- [4] Suricata Intrusion Detection and Prevention Engine, redmine.openinfosecfoundation.org/projects/show/suricata, 2011.
- [5] Prelude Hybrid Intrusion Detection System, prelude-ids.org, 2011.
- [6] The Bro Network Security Monitor, www.bro-ids.org, 2011.
- [7] Roesch, M., "Snort-lightweight intrusion detection for networks," In Proceeding of USENIX LISA Conference, pp. 229–238 (1999)
- [8] Wheeler P., and Fulp E. W., "A taxonomy of parallel techniques for intrusion detection", In Proceedings of the 45th Annual Southeast ACM Regional Conference, Winston-Salem, North Carolina, USA, P.P 278-282, 2007.
- [9] Kopek C. V, Fulp E. W, and Wheeler

P. S., "Distributed Data Parallel Techniques for Content-Matching Intrusion Detection Systems", In Proceeding of MILCOM 2007 IEEE Military Communications Conference (2007), PP 1-7, 2007.

[10] Sanz-Bobi M. A., Castro M., and Santos J., "IDSAI: A Distributed System for Intrusion Detection Based on Intelligent Agents," In Proceedings of the Fifth International Conference on Internet Monitoring and Protection, pp.1-6, 2010.

[11] Schuff D. L., Choe Y. R., and Pai V. S., "Conservative vs. optimistic parallelization of stateful network intrusion detection," in proceeding of the 12th ACM SIGPLAN symposium on Principles and Practice of parallel programming, 2007

[12] Yoshioka A., Shaikot S. H., and Kim M. S., "Rule Hashing for Efficient Packet Classification in Network Intrusion Detection," In Proceeding of the

[13] Beale J., Foster J. C., Jeffrey Posluns, and Brian Caswell, "Snort 2.0 Intrusion Detection", Syngress Publishing, 2003.

[14] Menga J., and Timm C., "CCSP Secure Intrusion Detection and SAFE Implementation Study Guide", SYBEX, 2004.

[15] Madhi D., and Ramasamy K., "Network routing algorithms, protocols, and architectures", Morgan Kaufmann, USA, pp. 1-957, 2007.

[16] Varghese G., "Network Algorithmics An Interdisciplinary Approach to Designing Fast Networked Devices", Morgan Kaufmann, USA, pp. 1-491, 2005.

[17] Lippmann R., Haines J. W., Fried D. J., Korba J., and Das K., "Analysis and Results of the 1999 DARPA Off-Line Intrusion Detection Evaluation", Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection, p.162-182, October 02-04, 2000.

International Conference on Computer Commu

تحسين أنظمة اكتشاف الهجمات باستخدام البيئة الموزعة

أنيس محسن هادي

كلية العلوم للبنات

جامعة بابل

وسام سمير بهيه

كلية تكنولوجيا الحاسبات

جامعة بابل

صفاء عبيس مهدي

كلية تكنولوجيا الحاسبات

جامعة بابل

أصبحت دراسة أنظمة اكتشاف الاختراقات من الثوابت المهمة في أمنية الشبكات . حالما يكشف نظام الاختراقات أي هجوم ، فإنه يقوم بتوليد مجموعة من التحذيرات التي تشير الى وجود خرق أمني في الشبكة . برنامج Snort المفتوح المصدر هو احد أنظمة اكتشاف الاختراقات المجانية ولكنه يعاني من مشكلة البطء نسبيا اذا ما تم استخدامه مع الشبكات السريعة . لحل هذه المشكلة ، نقترح استخدام عدة نسخ من برنامج Snort مع مصنف حزم . كل نسخة من هذا البرنامج تحتوي على جزء من المجموعة الكلية للهجمات المراد مراقبتها . أما دور مصنف الحزم فهو توزيع الحزم على النسخ المستخدمة من برنامج Snort وكذلك لضمان عدم ارسال الحزم الخاصة بهجوم واحد الى اكثر من نسخة من برنامج Snort وبالتالي تجنب الاكتشاف . النتائج العملية مع بيانات DARPA 1999 بينت ان كفاءة البرنامج Snort قد تحسنت بنسبة 30% .

توليد شفرة السمات الدلالية للأسماء آليا في المعجم المحوسب

سلمى محمود عبد الباقي

يسرى مالك ضمد

جامعة البصرة /كلية العلوم/قسم علوم الحاسبات

المستخلص:

تعد حوسبة المعجم من أهم مجالات علم اللغة الحاسوبي وأكثرها تلبية للمتطلبات العلمية والثقافية في الدول المتقدمة في العالم المعاصر، إذ هناك معاجم حاسوبية وإلكترونية متنوعة في الغرب تساعد على السرعة والدقة في إنجاز الكثير من التطبيقات. على حين تعدّ حوسبة المعجم العربي مهمة جدا" في وضعه وتنظيمه إلا أنه يفتقر الى من يقوم بدراسته، لان من يبحث في الاتجاه المعجمي قليل ونادر في العالم العربي وذلك لأسباب جوهرية من أهمها أنه لا يخفى على أحد ان معالجة اللغة آليا" مع مراعاة نظام القواعد والاستعمال اللغوي والتطور التاريخي تلقي بثقلها على حوسبة المعجم.

بني أنموذج المعجم الدلالي الحاسوبي المقترح على الأسس الآتية:

- 1- اعتماد أصناف الكلام الرئيسية للغة العربية وميزات كل صنف.
- 2- اعتماد أنواع العلاقات الدلالية بين المفردات والعمل بميزات كل علاقة.
- 3- اعتماد تصنيف الأفعال بطريقة دلالية ساعدت على وضع الأطر العامة لها.
- 4- جمع المفردات المجردة تحت صنف واحد هو صنف الصفات لعدم تشابه ميزاتها مع ميزات الأسماء على الرغم من كونها أسماء.
- 5- اعتماد هياكل بيانية متنوعة لتمثل كل صنف حسبما يتطلبه ميزات الصنف.
- 6- إنشاء قاعدة بيانات كمكان للخرن الثابت تحتوي على جميع مفردات المعجم مع المعلومات الخاصة بكل مفردة.
- 7- بناء آلية فهرسة لكل هيكل بياني لضمان الوصول الى المدخل بأقل وقت وجهد.

تناول البحث الحالي حل مشكلة السمات الدلالية التي تعتبر من أكثر مشاكل المعاجم المحوسبة أهمية نتيجة الجهد المبذول في تزويد قاعدة بيانات المعجم بجميع المعلومات لكل مفردة من مفرداته. إذ ان المستخدم ليس له علاقة بالسمات الدلالية للمفردات المدخلة بل تنتج آليا وهذا ما يسهل عمل الباحثين في معالجة اللغات والتي تعتبر هذه النقطة هي حجر العثرة لديهم في إدخال مفردات معجمهم ولهذا يلجأون إلى تحديد المجال (طبي ، صناعي ، ...) لتقليل عدد المفردات وللسيطرة على إدخال سماتها الدلالية .

بالنظر لميزة هيكل الأسماء الهرمي التدرجي المعتمد على أنواع خاصة من العلاقات الدلالية ظهرت آلية التوارث التي من خلالها حصل على السمات الدلالية للأسماء آليا مسجلين بذلك خطوة باتجاه العمومية وهي ميزة مهمة من ميزات المعجم المقترح.

الكلمات المفتاحية: معالجة اللغات، المعجم، النظرية الدلالية

المقدمة:

ومحاصرة ظواهره بالوسائل التقليدية اليدوية في بناءه

وتحديثه واستخدامه. لهذا أصبحت مكتنة المعجم مطلباً

إن المعجم ليس مجرد قائمة مفردات بل هو هيكل معقد

ومتشابه وهو غاية في التعقيد يتعذر معه فهم أدائه

سلسلة من البحوث والانتاجات وكان آخرها هو معجم العهد الجديد (Greek New Testament) والذي يقوم على أربعة تصنيفات رئيسية.

- 1- الموجودات Entries.
 - 2- الأحداث Events.
 - 3- المجردات Abstracts.
 - 4- العلاقات Relations.
- تحت كل قسم هناك أقسام فرعية أخرى وهكذا ..

أساسياً تفرضه طبيعة المعجم وتنظيمه وتحديثه لما لها من أهمية في حياة البشر. [1,2,3]

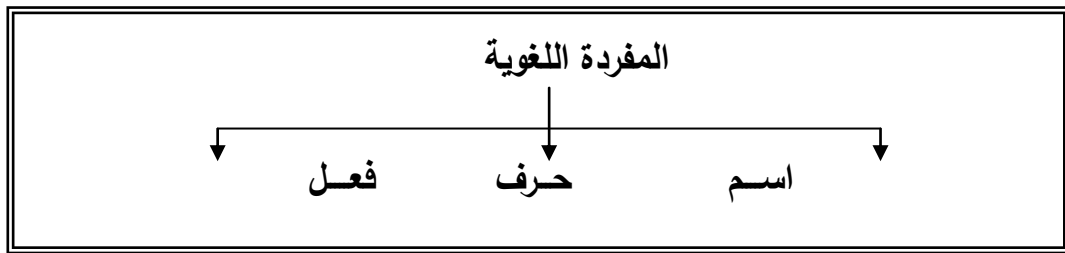
وبما أن المعجم يزخر بشبكة من العلاقات التي تربط بين مفرداته وبين فصائل المفردات وحقولها الدلالية وسماتها التفصيلية ، لهذا جاء هذا البحث في دراسة مستفيضة للعلاقات في الحقول الدلالية لاستغلالها في بناء هيكل معجم دلالي معرفي والتركيز على هيكل الأسماء. [4,5]

نظرية الحقول الدلالية: [6,7,8,9]

يمكن توضيح معنى الحقل الدلالي بأنه مجموعة من الكلمات التي ترتبط دلالاتها وتوضع عادة تحت لفظ عام يجمعها، فلفهم معنى كلمة يجب أن نفهم مجموعة الكلمات المتصلة بها دلالياً. لهذا يجب دراسة العلاقات بين المفردات داخل الحقل أو الموضوع الفرعي ومن هذا فإن معنى الكلمة هو محصلة علاقاتها بكلمات الأخرى في داخل الحقل المعجمي في هذه النظرية ظهر معجم الحقول الدلالية المصنف الذي جاء بعد

مداخل المعجم اللغوي

يحتوي المعجم اللغوي العربي على العديد من الأصناف منها الأسم والفعل والحرف ... وغيرها من الأصنافوكما موضح في الشكل(1) ، لكل صنف من هذه الأصناف تمثيله الخاص وعلاقاته الخاصة مع بقية مكونات المعجم.



الشكل(1) الأصناف الرئيسية للمفردات اللغوية داخل المعجم اللغوي العربي

الاسم :

- علاقة الاشتمال و التضمين.
- علاقة الكل والجزء.
- علاقة الترادف.
- علاقة التضاد.

يعرف الاسم على أنه كلمة تدل على معنى مستقل بالفهم وليس الزمن جزءاً منه [12,11,10]. يتميز صنف الأسماء بمجموعة ميزات التي تربطها داخل المعجم اللغوي هي:

1- تعكس الأسماء هيكلاً هرمياً تدرجياً ضمنياً داخل المعجم ، تبدأ من مفاهيم ذات عمومية إلى مفاهيم أقل عمومية وهكذا نحو التخصصية. كما موضحة في الشكل (2).

ومن خلال هذه الميزات يمكننا الحصول على جملة من الاستنتاجات التي أسهمت في تصميم وبناء هيكل الاسم في المعجمالمقترح هي:

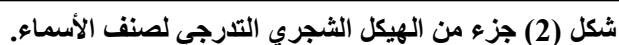
2- تربط هذه المفاهيم بعلاقات متنوعة محددة ضمن الهيكل الهرمي وأهم هذه العلاقات:

1- من هيكل الهرمية التي يتم فيها التدرج من المفاهيم العامة الى المفاهيم الخاصة يتولد لدينا مايسمى بنظام التوارث ، حيث يتم وراثته جميع صفات المفهوم الاعم الى المفهوم الخاص

3- [13,15,16,17,18,19]

2-إن العلاقة الرئيسية التي تربط بين المفاهيم وتسهم في بناء الهيكل الهرمي هي علاقة الاشتمال والتضمين وأما العلاقة الأخرى التي تُسهم في بناء جزء من الهرم هي علاقة جزئه ب كله

-4



فلضخامة المعجم اللغوي سيكون تخزين كل بيانات المداخل

أمراً غير مجدٍ لذا أرتئينا خزن البيانات الأساسية لمداخل المعجم والتي تمثل بيانات ثابتة للمفردات اللغوية في هيكل تخزيني دائم هو قاعدة البيانات ، بينما يتم تجميع وتكملة بقية بيانات المداخل في أثناء المعالجة في هيكل شجري يعكس تمثيل صنف الأسماء في المعجم اللغوي ويوضح طبيعة العلاقات الدلالية التي تربطها ويضمن التوارث وتكاملية بيانات المداخل خلال المعالجة. ولضمان سهولة التعامل مع المداخل وسرعة الوصول الى أي منها في أثناء المعالجة

- اسم المدخل (أو أسم العقدة في الهيكل الشجري فيما بعد): وهو عبارة عن الكلمة الرئيسية أو الأسم الممثل للمفردة اللغوية.
- أذن سيكون لدينا مجموعة من الهياكل البيانية المترابطة لتمثيل صنف الأسماء هي :
 - أ. قاعدة البيانات.
 - ب. الهيكل الشجري.
 - ج. ملف فهرسة
- في الفقرات التالية شرح لمكونات كل هيكل من هذه الهياكل وكيفية الارتباط بينها.
- **أ. قاعدة البيانات**

يحتوي جدول قاعدة البيانات على الحقول التالية التي تمثل البيانات الأساسية الثابتة لأي مدخل أسمى في المعجم:

 - **الاسم** : قائمة بجميع الكلمات المرادفة لأسم العقدة الرئيسي.
 - **التضاد** : قائمة بجميع الكلمات المضادة لأسم العقدة الرئيسي .
 - **الصفة المميزة** : وتمثل الصفة التي تميزها عن العقدة الأب .
 - **المعنى** : وهو عبارة عن شرح لأسم العقدة.

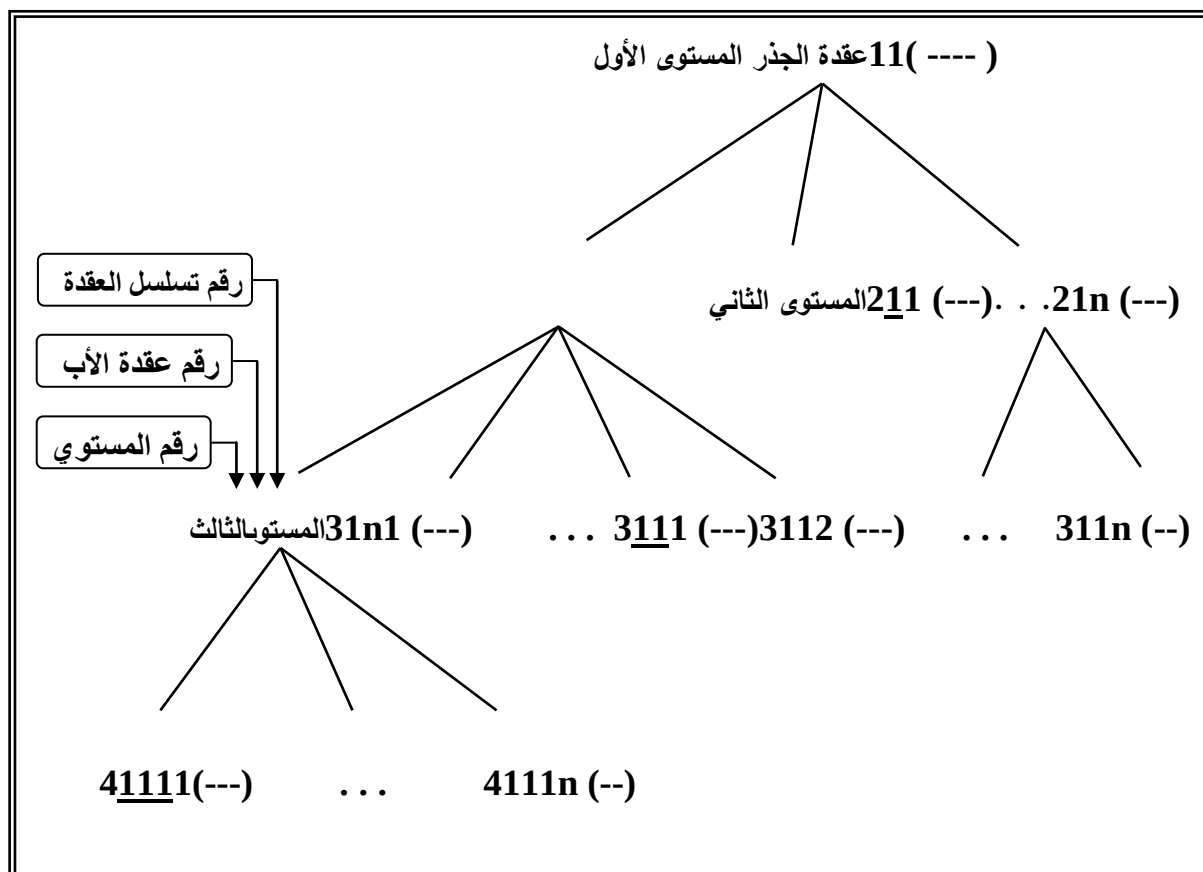
وكما موضح في الجدول (1):

الجدول (1) جزء من جدول الاسماء

الاسم	الأب	العلاقة	الصفة المميزة	المرادف	التضاد	المعنى
كائن			ملموس	موجود		
حي	كائن	الضمين	له روح		غير حي	
غير حي	كائن	الضمين	ليس له روح	جامد	حي	
حيوانات	حي	الضمين	محرك			
نبات	حي	الضمين	غير محرك	زرع		
حروان	حيوانات	الضمين	غير عاقل			
إنسان	حيوانات	الضمين	عاقل	بشر، آدمي		
أنفة	حروان	الضمين	أخصابية		مفترسة	
مؤحضة	حروان	الضمين	مؤذية		أنفة	

- ب. **الهيكل الشجري**
- تمثل حقول قاعدة البيانات الحقول الثابتة المميزة لطبيعة ومميزات المفردة اللغوية أو المدخل المعجمي، تنقل في أثناء
- التنفيذ إلى الهيكل الشجري المقترح مع حقول أخرى تُحسب في أثناء التنفيذ والمعالجة وكما موضح في أدناه :

- رقم العقدة التسلسلي.. ويمثل رقم المستوى في الشجرة ورقم
تسلسل العقدة في المستوى حيث يصبح رقم تسلسل العقدة هو
- رقم العقدة الاب للعقدة التالية كما موضح في الشكل (3).
- عناوين عقد الاطفال.. ليتسنى ربط عقد الاطفال بالعقدة الاب
- عنوان عقدة الاب .. ليتم من خلاله ربط العقدة الحالية بالعقدة الاب.
- عدد عقد الاطفال .. يمثل عدد العقد المرتبطة بالعقدة الحالية.



الشكل (3) الرقم التسلسلي للعقدة

بحيث ضمنت التحرك الى الاعلى أو نحو العمومية (علاقة الاشتمال/ الكل) والتحرك نحو الاسفل أو نحو الخصوصية (التضمنين /الجزء).

تكون الحقول المنقولة من قاعدة البيانات والحقول المحسوبة في أثناء التنفيذ مكونات عقدة المدخل المعجمي في الهيكل الشجري. يتميز الهيكل الشجري المقترح بسهولة إضافة حقول أخرى مثل الجذر، الوزن ، أصل الكلمة،... وغيرها ، وحسب متطلبات البحث ونوع التطبيق الحاسوبي والتي أهملت هنا لمحاولة تقليل التفاصيل التي قد تكون غير ضرورية لمتطلبات البحث الحالي في الوقت الحاضر.

أضيفت أسماء الأعلام غير المشتقة مثل (زينب ، زهراء ، محمد ، حسين،...) الى شجرة الاسماء كمفهوم متضمن في

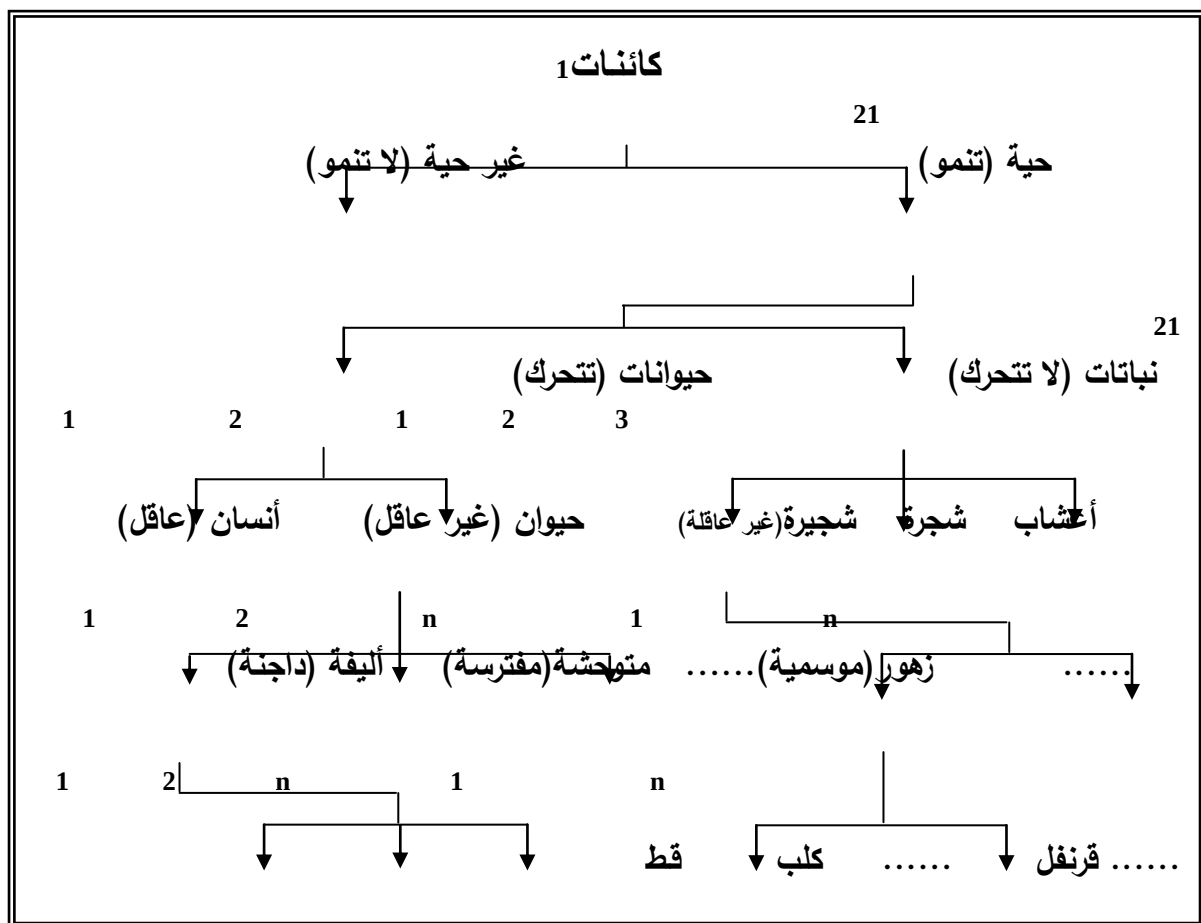
وعلى هذا الاساس بُني هيكل صنف الأسماء بشكل شجرة متعدد الأطفال. أعتمدنا في بناء هيكل الشجرة عدة مبادئ هي :-

- 1- لكل عقدة طفل عقدة أب واحدة (فقط) لمنع حدوث حالات (الدائرية).
- 2- هيئة الكلمات أو أشباه الجمل يجب أن تكون واضحة وخالية من الغموض .
- 3- يجب أن تخضع عقدة الأب لأحدى العلاقات الدلالية مع الأب (الاشتمال والتضمنين أو الكل والجزء) في التنظيم الهرمي.
- 4- هناك ربط ثنائي بين عقدتي الأب والأبن ، سهلت عملية الربط الثنائي هذه الحركة الثنائية على شجرة الاسماء ،

السابقين المرتبط معهم بعلاقات ويضاف إليها صفة واحدة على الأقل تميزه عن أسلافه.

تعد السمة الدلالية للاسم ضرورية جداً بل أساسية في أي معالجة للغة الطبيعية ولهذا فقد يضطر الباحثون في هذا المجال بإدخال كل السمات الدلالية مع كل أسم في أي نظام معالجة يتم تصميمه، وهذه العملية بعد ذاتها عملية شاقة حيث تتطلب وقتاً وجهداً ودقة بالإضافة الى عدم قدرة بعضهم على تحديد وأعطاء السمات الصحيحة للاسم، لذا غالباً ما يلجأ الباحثون الى تحديد المجال الذي يعالجونه كأن يكون علمياً، أدبياً، صناعياً، أو غير ذلك ليتسنى لهم حصر مجموعة المفردات اللغوية الخاصة بذلك المجال. في البحث الحالي ، تُحسَبُ قيمة عددية تمثل السمة الدلالية لكل أسم ألياً من خلال هيكل الشجرة بدون أن يتقل كاهل المستخدم وكما موضح في الشكل(4).

1. تنفيذ آلية توارث الصفات من الجذر الى العقد في المستويات الأدنى من الشجرة. هذه الآلية ساعدت في إيجاد السمات الدلالية للأسماء والتي هي عبارة عن مجموعة الصفات التي تميز الكلمات عن بعضها، إذ تُجمَع هذه الصفات للاسم الحالي عن طريق وراثتها عن أسلافه



الشكل (4) كيفية تمثيل السمات الدلالية للاسماء في هيكل الشجرة.

نقطة التقاء هاتان الكلمتان عند الجد الرابع الا وهو الكائنات الحية.

3. العمومية والخصوصية : يقصد بالعمومية هي الصفات العامة للمفهوم والتي يرثها عن طريق أسلافه أما الخصوصية فهي الصفات التي تميزه عن أسلافه. في بعض الحالات التي تواجهنا مثلاً في الجملة التالية: "نمى الأظفر" هل يمكن تحديد صحة هذه الجملة دلاليّاً ألياً ؟ هل يمكن أن يعد الأظفر كائناً حياً وبالتالي يمتاز بصفة النمو؟ . كما هو معروف بأن الأظفر ليس كائناً حياً بل هو جزء منه. أذن هل يمكن أن تعتبر هذه الجملة خاطئة دلاليّاً ؟ . من أيجاد أسلاف كلمة الأظفر نستطيع أثبات صحة هذه الجملة .

جزء من ← جزء من

بنفس السمات الدلالية تدعى عندها الكلمتان بالمتراذفتين، إذ جُمِعَتْ بعض الاسماء المترادفة في مجموعات مترادفة وأعطيت لها السمات الدلالية نفسها، بحيث يمكن استخدام أي من هذه الاسماء ضمن المجموعة الواحدة في الوصول الى المدخل المعجمي الخاص بالمجموعة. أن تقليل عدد المداخل بمقدار عدد المرادفات الموجودة في صنف الأسماء قلل من حجم المعجم وهذا الهدف يسعى له العاملون في مجال الصناعة المعجمية وهي تقليل عدد المداخل المعجمية الى أقل حد ممكن لتحقيق المعادلة (تزايد حجم المعجم والضغط الاقتصادي عليه).

5. المعنى والاسلاف : من المشاكل التي واجهت البحث الحالي وجود العديد من الكلمات التي ليس لها معنى مباشر، فمثلاً لا نجد معنى مباشر لكلمة "تمر" في معجم مختار الصحاح[18].

من الهيكل المقترح يمكن الحصول على معنى هذه الكلمة من خلال تتبع أسلافها وكما يأتي:

ويمكن توضيح ذلك بمثال على الشكل (4):

رمز السمة الدلالية لكلمة كلب هي 121212 ترمز للسمات (ينمو، يتحرك، غير عاقل، داجنة).

رمز السمة الدلالية لكلمة قرنفل هي 122111 ترمز للسمات (تنمو، لا تتحرك، غير عاقلة، موسمية).

2. حساب درجة قرابة الاسماء : تساعد القيمة العددية الممثلة للسمة الدلالية للأسماء في أيجاد قرابة الاسماء (بمعنى الصفات المشتركة بينها) من عدمها ، وذلك عن طريق معرفة أسلافها وعند أي جد يلتقيان. وللمزيد من التوضيح ، في المثال السابق

السمة الدلالية لكلمة الكلب هي 121212 ، والسمة الدلالية لكلمة القرنفل هي 122111 ، يمكن عندئذ تحديد ومعرفة

الأظفر الأصبع ← جزء من ← القدم اليد ، الساق ← جزء من

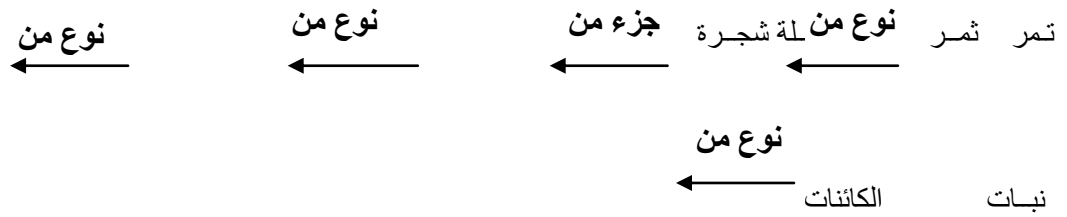
الجسم ← جزء من ← الكاء نوع من

في حالات أخرى مثلاً الجملة "مشى الأظفر" ، هل يمكن أتباع الأسلوب نفسه أعلاه بتتبع وأيجاد أسلاف الأسم الأظفر لأثبات صحة الجملة دلاليّاً أم لا ؟ في مثل هذه الحالة نلاحظ بأن العلاقة التي تربط بين الأسلاف والتي لها تأثير كبير على المفردات. الجواب بكل تأكيد أن الجملة غير صحيحة دلاليّاً وذلك لأن حكم علاقة "نوع من" غير حكم علاقة "جزء من"، حيث تعتبر الأولى مستقلة أما الثانية فتكون غير مستقلة ، بمعنى أوضح أن علاقة "الجزء من" يكون ارتباطها قوياً بالمفهوم العام الاقرب لها فتأخذ بذلك الصفة المميزة له لذا تكون الجملة "مشى الأظفر" غير صحيحة. بينما تعد الجملة "مشى القط" صحيحة دلاليّاً وذلك

قط نوع من حيوان أليف ← نوع من حيوان نوع من حيوانات نوع من كائن حي

هذا بالإضافة الى أن تحديد صحة الجملة يعتمد على إطار الفعل الدلالي الذي يحدد السمة الدلالية للفاعل .

4. الوصول المباشر وتقليل عدد المداخل المعجمية من معنى الترادف أنه تشابه معنى كلمتين ، بمعنى اشتراك كلمتان



ج. ملف الفهرسة

أن الوصول لأي عقدة أو مدخل في الهيكل الشجري لأي سبب كأن يكون ربط عقدة جديدة معها، حذف عقدة ، الحصول على معلومات عنها،... الخ يتطلب بحثاً تتابعياً للوصول الى العقدة المطلوبة ومن ثم أستهلاك الكثير من الزمن والمعالجة، ذلك لكون عقد الشجرة مرتبطة بمؤشرات مع بعضها مما يستوجب البحث من الجذر وحتى العقدة المطلوبة ولاسيما أن أسماء العقد غير مرتبة أبجدياً. للتغلب على مساويء البحث التتابعي تم تبني آلية فهرسة تتلخص بخلق ملف فهرسة يحتوي على اسم المفردة بالاضافة الى موقعها في الشجرة أو عنوانها، بحيث تم درج

جميع الاسماء أو المداخل المعجمية مع المرادفات في مداخل مستقلة في الملف، علماً أن المرادفات تحمل عنوان الاسم الرئيس نفسه للمدخل المعجمي للأسماء. من أهم ميزات هذه الآلية هي:

- إمكانية الوصول المباشر للمدخل المعجمي عن طريق عنوانه وهذا يعني زمن وصول أقل.
 - إمكانية الوصول الى المدخل عن طريق الاسم الرئيسي له أو أحد مرادفاته.
- والشكل (5) يوضح تعريف عقدة المدخل في الهيكل الشجري

```

Node = ^nod
nod = Record
Name :
Syn :
Relation:
Featuer:
Mean:
Father_Add : pointer to nod
Fath_nam :
Level :
Aton :
Serialno :
Child : Array of pointer to nodes

```

الشكل (5) تعريف هيكل العقدة

خوارزمية (1) توضح كيفية بناء الشجرة التي تجلب معلوماتها من جدول الأسماء أما الخوارزمية (2) فتوضح كيفية إضافة وربط العقد الجديدة في شجرة الأسماء.

```

Algorithm Build_Tree
Getdat ( from Table_name );
Search for father_name in index_file
If found Then
Search for name in index_file
If not found Then
Create new node
Insert to tree (Algorithm Joined_Nod_Tree)
Insert Address node to index_file
Else
Error ( old entry )
End If
Else
Error ( father name is not found )
End If
End Algorithm Build_Tree

```

خوارزمية (1) بناء شجرة التصنيف الهيكلي للأسماء

```

Algorithm Joined_Nod_Tree
Get Address Of Father Name
Put Address of node in Array Children Address To father node and give index of node
Put Address of father node in father address of node
End Joined_Nod_Tree

```

خوارزمية (2) إضافة وربط العقد في الشجرة

العمل المستقبلي:

- 1- توسيع قاعدة العلاقات المعجمية وبإضافة واستخدام علاقات دلالية (التضاد بأنواعه ، التنافر ، ... وغيرها) كمحاولة لتوسيع تطبيق المعجم واستخدامه لمعالجات مختلفة.
- 2- يمكن استخدام المعجم المقترح في تطبيقات متعددة منها الترجمة الآلية بأضافة المفردات اللغوية للغات أخرى غير العربية، ومعالجة أستخلاص النصوص ، وغيرها نتيجة مميزات هيكل الاسماء.

المصادر

1. آل ياسين، الشيخ محمد حسن، 1992، "المعجم الذي نطمح إليه"، المعجمية العربية أبحاث الندوة التي عقدها المجمع العلمي العراقي، مطبعة المجمع العلمي العراقي.
2. الاثري، محمد بهجت، 1992، "أصالة المعجمية العربية"، المعجمية العربية أبحاث الندوة التي عقدها المجمع العلمي العراقي، مطبعة المجمع العلمي العراقي.
3. ظاظا، د.حسن، 1976، "كلام العرب من قضايا اللغة العربية"، دار النهضة العربية للطباعة والنشر، بيروت، لبنان.
4. علي، د.نبيل، 1988، "اللغة العربية والحاسوب (دراسة بحثية)"، تعريب للطباعة والنشر، القاهرة، مصر.
5. Kontos, J., Malagardi, I., Bouligarak, M., 2000, "Semantics of technical lexicons", department of methodology, history and theory of science, University of Athens.
6. جواد، أحمد، 2002، "الحقول الدلالية وأشكالها المعنى"، مجلة المورد، العدد الثاني، المجلد الثلاثون، بغداد، العراق.
7. عمر، د.أحمد مختار، 1982، "علم الدلالة"، مكتبة دار العروبة للنشر والتوزيع، جامعة الكويت، الكويت.

8. عبد الجليل، منقور، 2001، "علم الدلالة أصوله ومباحثه في التراث العربي"، منشورات اتحاد الكتاب العرب، دمشق، سوريا.
9. محمود، سلمى عبد الباقي، 2001، "معجم عربي معرفي لفهم اللغة العربية"، أطروحة دكتوراه، جامعة البصرة، كلية العلوم، قسم علوم الحاسبات، البصرة، العراق.
10. الدحاح، أنطوان، 1989، "معجم قواعد اللغة العربية في جداول ولوحات"، مكتبة لبنان، بيروت، لبنان.
11. زوين، د.علي، 1986، "منهج البحث اللغوي بين التراث وعلم اللغة الحديث-دراسات"، دار الشؤون الثقافية العامة للطباعة والنشر، بغداد، العراق.
12. شامي، د.يحيى، 1993، "القواعد العربية الميسرة"، دار الفكر العربي، بيروت، لبنان.
13. الماشطة، مجيد عبد الحميد، فالح، حليم حسين، باقر، كاظم حسين، 1980، "علم الدلالة"، ترجمته لكتاب "Introduction to theoretical linguistic" للمؤلف "John Lyons"، 1968، كلية الاداب، جامعة البصرة، البصرة، العراق.
14. علي، د.نبيل، 1988، "اللغة العربية والحاسوب (دراسة بحثية)"، تعريب للطباعة والنشر، القاهرة، مصر.
15. Banerjee, S. & Pedersen, T., 2001, "An adapted Lesk algorithm for word sense disambiguation using wordnet", University of Minnesota, Duluth, MN 55812 USA, <http://www.d.umn.edu/~bane0025/~tpederse>.
16. Dym, C., L. & Levitt, R., E., 1991, "Knowledge-base systems in engineering", McGraw-Hill, USA.
17. Elkateb, S. & Black, B., 2000, "English-Arabic dictionary for translators", department of computation, UMIST, Po box 88, Manchester M60 1QD, UK.

19. Rich, E., 1983, "Artificial Intelligence", McGraw-Hill, Singapore.
18. Hirst, G., 2003, "Ontology and lexicon", department of computer science, University of Toronto, Canada.

Generating automatically for semantic features code in computation lexicon

Salma M. Abd Al-B

Yusra M. Dummad

Basra University / Sciences College / Computer Sciences Dept.

Abstract

The computational lexicon is one of the most important fields in the computational languages science and is the most compliance to the scientific and cultural needs in the developed countries of the modern world; there are different computational and electrical lexicons in the west helps in speeding up and the accuracy in accomplishing of many applications. The computational Arabic lexicon is very important in its creation and organization but its wants who studied it, because the researches in the lexicon direction are few and infrequent in the Arabic world, that is because of the substantial reasons such that the mechanism language with the consideration of the grammar system, the language use and the historical development, which irritate the lexicon computations.

The suggested computational semantic lexicon model is built up upon the following basics:-

1. Using the main speech categories in the Arabic language and the properties of every category.
2. Using the semantic relations kinds between the vocabulary and the work with the properties of every relation.
3. Using the semantic way in classifying verbs which helps us to put the general frames to it.
4. To bring together the mere vocabulary under one category named the properties category because the unlike of its properties with the noun properties in spite of it is also nouns.
5. Using different data structures to represent every category depending on the properties of the category.
6. Building a data base as a place to the fixed storage consisting all of the lexicon vocabularies with the information related with every vocabulary.

-
7. Building an indexing mechanism to every data structure to insure the access to the entrance with a shortest time and less efforts.

This study solve the problem of semantic features which consider most difficult problem in the computation lexicons field because of hardly accommodation database of lexicon to all information for any particular from vocabulary and quality uses.

For facility researcher work in natural languages processing field which consider this point the misstep stone with them in input their lexicon vocabulary so they go to the domain limitation (medical, manufacture,.....) for redundancy number vocabulary and for their semantic features inputs control.

Regarded to the hierarchal property of the noun structure depends on a special semantic relation, the inheritance mechanism appears by which we acquire the semantic features to the nouns automatically recording by that a success step towards the generality.
