In the Name of God

Journal of
Information Systems & Telecommunication
Vol. 9, No.3, July-September 2021, Serial Number 35

Research Institute for Information and Communication Technology
Iranian Association of Information and Communication Technology
Affiliated to: Academic Center for Education, Culture and Research (ACECR)

Print ISSN: 2322-1437
Online ISSN: 2345-2773
Publication License: 91/13216
Editorial Office Address: No.5, Saeedi Alley, Kalej Intersection., Enghelab Ave., Tehran, Iran,
P.O.Box: 13145-799 Tel: (+9821) 88930150 Fax: (+9821) 88930157
E-mail: info@jist.ir , infojist@gmail.com
URL: www.jist.ir

Indexed by:
- SCOPUS www. Scopus.com
- Index Copernicus International www.indexcopernicus.com
- Islamic World Science Citation Center (ISC) www.isc.gov.ir
- Directory of open Access Journals www.Doaj.org
- Scientific Information Database (SID) www.sid.ir
- Regional Information Center for Science and Technology (RICeST) www.ricest.ac.ir
- Iranian Magazines Databases www.magiran.com

Publisher:
Iranian Academic Center for Education, Culture and Research **(ACECR)**

This Journal is published under scientific support of
Advanced Information Systems (AIS) Research Group and
Telecommunication Research Group, ICTRC

Acknowledgement

JIST Editorial-Board would like to gratefully appreciate the following distinguished referees for spending their valuable time and expertise in reviewing the manuscripts and their constructive suggestions, which had a great impact on the enhancement of this issue of the JIST Journal.

(A-Z)

- Agahi, Hamed, Islamic Azad University of Shiraz, Iran
- Agarwal, Parul, Jamia Hamdard University, New Delhi, India
- Ahmadizad, Arman, University of Kurdistan, Kurdistan, Iran
- Ahmadifar,Hamidreza, University of Guilan, Rasht, Iran
- Alavi, Seyed Enayatallah, Shahid Chamran University, Ahvaz, Iran
- Alam Tabriz, Akbar, Shahid Beheshti University, Tehran, Iran
- Azizi, Sadoon, University of Kurdistan, Sanandaj, Iran
- Badie, Kambiz, Tehran University, Tehran, Iran
- Babaei, Shahram, Islamic Azad University of Tabriz, Iran
- Bahrepour, Davoud, Islamic Azad University, Mashhad, Iran
- Bayanati, mahmonir, West Tehran Islamic Azad University, Tehran, Iran
- Bhalaji N, Sri Sivasubramaniya Nadar College, Chennai, Tamil Nadu, India.
- Bouyer, Asgar Ali, Shahid Madani University, Azarbaijan, Iran
- Darmani, Yousef, K. N. Toosi University of Technology, Tehran, Iran
- Davarpanah, seyed hashem, Birjand University of Technology, South Khorasan, Iran
- Ebadati, Omid Mahdi, Kharazmi University,Tehran, Iran
- Ebrahimpour, Nader, Islamic Azad University of Mahabad, West Azerbaijan, Iran
- Farsi, Hassan, University of Birjand, South Khorasan, Iran
- Fatemi Khorasgani, Afsaneh, University of Isfahan, Isfahan, Iran
- Fekri Ershad, Shervan, Islamic Azad University, Najafabad, Iran
- Geetha R, Dayananda Sagar Academy of Technology, Karnataka, India
- Ghayoomi, Masood, Institute for Humanities and Cultural Studies, Tehran, Iran
- Ghaffari, Ali, Islamic Azad University, Tabriz Branch, Iran
- Ghasemzadeh, Mohammad, Yazd University, Yazd, Iran
- Haghighi, Hassan, Shahid Beheshti University, Tehran, Iran
- Kheirkhah, Fatemeh, ACECR, Tehran, Iran
- Mirroshandel, Seyed Abolghasem, University of Guilan, Rasht, Iran
- Mirzaei, Abbas, Islamic Azad University, Ardabil, Iran
- Minoofam, Seyed Amir Hadi, Qazvin Islamic Azad University, Qazvin, Iran
- Mir, Ali, Lorestan University, Iran
- Momtazi, Saeedeh, Amir kabir University, Tehran, Iran
- Mohammadi, Mohammad Reza, Iran University of Science and Technology, Tehran, Iran
- Mothku, Sai Krishna, National Institute of Technology Tiruchirappalli, Tamil Nadu, India
- Nangir, Mahdi, University of Tabriz, Tabriz, Iran

- Reshadat, Vahideh, Malek-Ashtar University of Technology, Tehran, Iran
- Rezvanian, Alireza, University of Science and Culture, Tehran, Iran
- Sarrafzadeh, Abdolhossein, Unitec Institute of Technology, Auckland, New Zealand
- Sekar, M, Raja, VNR VJIET, Institute of Engineering and Technology, Hyderabad, India
- Shirmarz, Alireza, Ale Taha Institute of Higher Education, Tehran, Iran
- Sohrabi, Babak, Tehran University, Tehran, Iran
- Solouk, Vahid, Urmia University of technology, Iran
- Vu Duc, Nghia, Chung-Ang University, Seoul, Korea
- Yaghoobi, Kaebeh, Ale Taha Institute of Higher Education, Tehran, Iran
- Zare, Hadi, Tehran University, Tehran, Iran

Table of Contents

Cost Benefit Analysis of Three Non-Identical Machine Model with Priority in Operation and Repair

Nafeesa Bashir*
Department of Statistics, University of Kashmir
nafeesabashir8@gmail.com
Raeesa Bashir
Mathematics and Quantitative Analysis, Amity University, Dubai, UAE
raisabashir52@gmail.com
J P Singh Joorel
Department of Statistics, University of Jammu
joorel@rediffmail.com
T R Jan
Department of Statistics, University of Kashmir
drtrjan@gmail.com

Received: 15/Aug/2020 Revised: 10/May/2021 Accepted: 07/June/2021

Abstract

The paper proposes a new real life model and the main aim is to examine the cost benefit analysis of Textile Industry model subject to different failure and repair strategies. The reliability model comprises of three units i,e Spinning machine (S), Weaving machine (W), Colouring and Finishing machine(Cf). The working principal of the model starts with spinning machine (S) where in unit S is in operative state while as weaving machine, Colouring and Finishing machine are in ideal state. Complete failure of system is observed when all three units of system i.e. S,W and Cf are in down state. Repairperson is always available to carry out the repair activities in the system in which first priority in repair is given to Colouring and Finishing machine followed by Spinning and weaving machine. The proposed model attempts to maximize the reliability of a real life system. Reliability measures such as Mean Sojourn time, Mean time to system failure, Profit analysis of system are examined to define the performance of the reliability characteristics. For concluding the study of such model, different stochastic measures are analyzed in steady state using regenerative point technique. The tables are prepared for arbitrary values of the parameters to show the performance of some important reliability measures and to check the efficiency of the model under such situations.

Keywords: Reliability Measures; Mean Sojourn Time; Laplace Transformation; Laplace -Stieltjes transformation.

1- Introduction

In the present scenario of global viable market, the basic objectives of industries is to reduce the production cost and enhance the productivity of manufacturing systems of processing Industries. To meet the increasing demand of the society, the industries must offer long term production. The efficiency and accessibility of industry can be made profitable in terms of higher production and lower maintenance costs. Availability of complex systems can be improved by considering different techniques of reliability like maintenance, inspection, repairs, priority to repair of the failed unit and so on. In addition, these techniques in industrial design can substantially strengthen operational dependability through better system design. A very trivial change in the technique of reliability can cause a profound change in the operational dependability, that is why it is vital to specify model reliability targets before any strategy plan is taken into consideration .

There are systems where upon failure in a complex system the repair of a component is given preference over the other component depending upon the significance of the component. Such systems are called priority systems. When a unit of lower priority is under repair and a unit of higher priority arrives for repair, the lower priority unit is pre-empted and higher priority unit is repaired. After completion of the service of higher priority unit, the lower higher priority unit is given service. In reliability modeling, it is found that accessibility and performance of failed and repairable systems can be enriched using redundancy technique. But due to the high cost, it is not always possible to keep the unit as spare. Also, it sometimes becomes essential to prioritize one unit over the other to increase performance, availability and hence the efficiency of the system. The idea of priority in repair discipline in reliability modeling has been

* Corresponding Author

considered by various authors researchers in order to make the system available for required service with least possible operational cost.

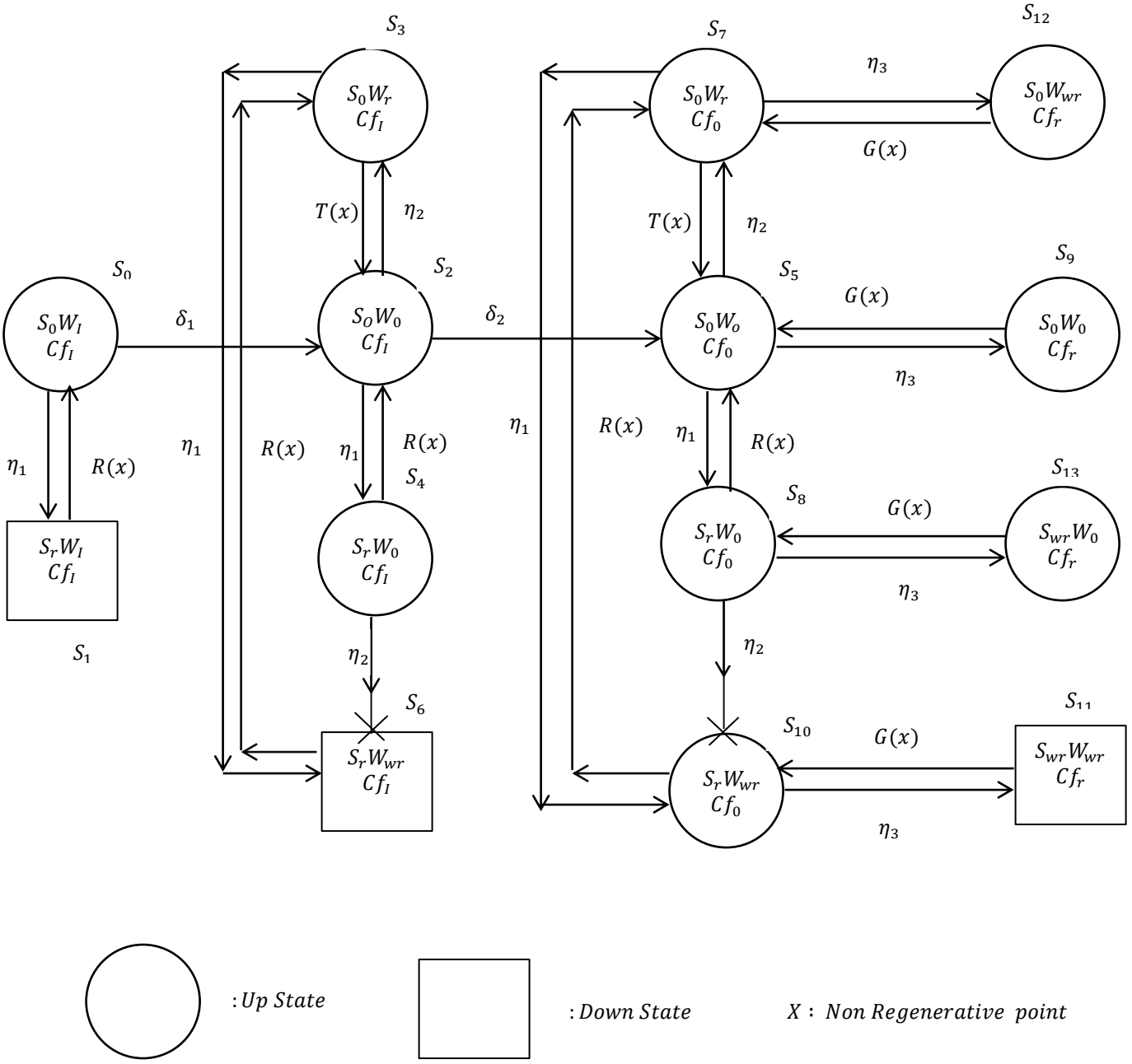
Kumar et al. [2] obtained reliability analysis of a warm standby model performing in normal/abnormal environment under certain assumptions where the distribution of failure time has been taken as negative exponential and the repair, inspection and replacement time distribution has been taken as arbitrary. Chander [11] discussed the two non-identical reliability models in which the failure and repair time of each unit were independent and uncorrelated random variables. The author obtained the various reliability measures using Regenerative Point Technique (R.P.T). Also, Zhang Wang [15] studies a cold standby system with priority in use and repair. Malik et al. [12] studied cost- benefit analysis of a model under and gave priority to repair and degradation. The numerical and graphical behaviour of important reliability measures has been obtained for arbitrary values of costs and parameter. Further, Kadyan et al. [6] considered reliability performance of model under priority to repair and degradation. The authors obtained the availability and profit of the system using Regenerative point technique (RPT). Sureria and Malik [5], Malik [13] analyzed computer system with priority to software replacement over hardware replacement and obtained the reliability and economic measures of computer system under certain assumptions. The model has been analyzed stochastically by using R.P.T and the results of the model has been compared with the existing model. Chib et al. [3] assumed two types of repair facilities for stochastic analysis of two unit priority system. Rathee and Chander [9] carried out the reliability analysis of parallel system in which he gave priority to repair over maintenance. Munday and Malik[14] obtained profit analysis of a Computer System with Priority for hardware redundancy and repair. Recently Kumar et al. [4] studied profit analysis of computer systems with priority and maximum operation time. The author gave priority to software upgradation over hardware repair and replacement and hence obtained the profit of the system. The profit of the system has been compared with some existing model in literature. Also Bashir et al. [7] used the technique of inspection of degraded unit and obtained cost analysis of two unit model. Gorjian Jolfaei [8] carried out reliability analysis of power generation engines in a wastewater treatment plant. Kumar and Malik [1] examines the weathering server system with identical element in warm standby and used the concept of first come first serve repair policy for redundant systems in weather conditions. The various reliability measures of the system has been obtained using R.P.T. The author proved that the concept of first come first serve repair policy proved to be economically beneficial. Sharma and joorel [10] developed a model for fruit juice manufacturing system for its reliability analysis, which consists of four non identical units viz. Fruit cutter, Fruit pulper, Sterlizer and Homogenizer. The author used the concept of emergency repair in which the team of experts is available on the failure of Sterlizer or Homogenizer, to carry out the failure activities. [16] Obtained the reliability of two unit cold standby model under the assumption that the system is operating under different weather conditions and gave priority to preventive maintenance over inspection. To determine the nature of the system, various system effectiveness has been obtained using regenerative point technique and semi markov process. [17] Analysed the concrete mixture plant which comprises of seven units subject to the coverage factor by considering profust reliability approach which is based on probability assumption as well as fuzzy state assumption. The performance analysis of the system has been done using Markov birth–death model and differential equations solved by Runge–Kutta method. The results of the proposed model have proved beneficial for system designers. [18] have made an effort to obtain the system performance of sewage treatment plant by using markovian birth-death process. Various system measures like reliability, availability, maintainability and obtainability of the system has been obtained. [19] explored the generators in steam turbine power plant and obtained the availability and profit of generators. Using supplementary variable technique amd markov birth-death process different system measures has been evaluated. Results of the system have been obtained for availability and profit that would be useful for system designers to enrich the reliability and thus performance of plant

The basic aim of the paper is to obtain the profit evaluation of textile industry with exponential failure time distribution subject to different repair strategies. The System consists of three non-identical components Spinning (S), Weaving (W), Coloring and Finishing (Cf). Initially unit S is in working mode whereas unit W and Cf are in idle state. Failure of the system occurs only if all the unit fails. Single server is available in the system to carry out repair activities and Cf gets first priority over repair followed by S and W. The failure time (FT) ~ exp. (η). The repair time distribution is taken as general.

2- Notations and Symbols

- η_1, η_2, η_3= failure rates of unit S, W, Cf respectively.
- δ_1, δ_2= activation rates of unit W and Cf from idle to operation respectively
- $R(t), T(t), G(t) =$ cdf of repair time of the failed unit S, W, Cf respectively
- $*$ = Laplace transform i.e. $f^*(s) = \int_0^\infty e^{-st} f(t) dt$

TRANSITION DIAGRAM



~ = Laplace-Stieltjes transform i.e. $\tilde{F}(s) = \int_0^\infty e^{-st}\, dF(t)$.

- $S_0, W_0, Cf_0 = S, W, Cf$ are normal and operative.
- $S_r, W_r, Cf_r = S, W, Cf$ are under repair.
- $S_{wr}, W_{wr}, Cf_{wr} = S, W, Cf$ are waiting for repair.
- $W_I, Cf_I = W, Cf$ is in ideal state.
- S_0 = Total possible states of the system, $i = 0, 1, 2, 3 \ldots 13$

$$S_0 = [S_0, W_I, Cf_I]$$
$$S_1 = [S_r, W_I, Cf_I]$$
$$S_2 = [S_0, W_0, Cf_I]$$
$$S_3 = [S_0, W_r, Cf_I]$$
$$S_4 = [S_r, W_0, Cf_I]$$
$$S_5 = [S_0, W_0, Cf_0]$$
$$S_6 = [S_r, W_{wr}, Cf_I]$$
$$S_7 = [S_0, W_r, Cf_0]$$
$$S_8 = [S_r, W_0, Cf_0]$$
$$S_9 = [S_0, W_0, Cf_r]$$
$$S_{10} = [S_r, W_{wr}, Cf_0]$$
$$S_{11} = [S_{wr}, W_{wr}, Cf_r]$$
$$S_{12} = [S_0, W_{wr}, Cf_r]$$
$$S_{13} = [S_{wr}, W_0, Cf_r]$$

3- Transition Probabilities

Let $Q_{ij}(t)$ represents the transition probability from state i to j in transient state. Then the various transition probabilities, can be written as follows:

$$Q_{01}(t) = \eta_1 \int_0^t e^{-(\eta_1+\delta_1)u} du$$
$$Q_{02}(t) = \delta_1 \int_0^t e^{-(\eta_1+\delta_1)u} du$$
$$Q_{10}(t) = R(u)du$$
$$Q_{23}(t) = \eta_2 \int_0^t e^{-(\eta_1+\eta_2+\delta_2)u} du$$
$$Q_{24}(t) = \eta_1 \int_0^t e^{-(\eta_1+\eta_2+\delta_2)u} du$$
$$Q_{25}(t) = \delta_2 \int_0^t e^{-(\eta_1+\eta_2+\delta_2)u} du$$
$$Q_{32}(t) = \int_0^t e^{-(\eta_1)u} dtu$$
$$Q_{36}(t) = \eta_1 \int_0^t e^{-(\eta_1)u} \bar{T}(u) du$$
$$Q_{42}(t) = \int_0^t e^{-(\eta_2)u} dr(u)$$
$$Q_{57}(t) = \eta_2 \int_0^t e^{-(\eta_1+\eta_2+\eta_3)u} du$$
$$Q_{58}(t) = \eta_1 \int_0^t e^{-(\eta_1+\eta_2+\eta_3)u} du$$
$$Q_{59}(t) = \eta_3 \int_0^t e^{-(\eta_1+\eta_2+\eta_3)u} du$$
$$Q_{63}(t) = R(u)du$$
$$Q_{75}(t) = \int_0^t e^{-(\eta_2+\eta_3)u} du$$
$$Q_{7,10}(t) = \eta_1 \int_0^t e^{-(\eta_1+\eta_3)u} \bar{T}(u) du$$
$$Q_{7,12}(t) = \eta_3 \int_0^t e^{-(\eta_1+\eta_3)u} \bar{T}(u) du$$
$$Q_{8,13}(t) = \eta_3 \int_0^t e^{-(\eta_2+\eta_3)u} \bar{R}(u) du$$
$$Q_{8,11}(t) = \eta_2\eta_3 \int_0^t \int_0^v e^{-(\eta_2)u} du\, e^{-(\eta_3)v}\, \bar{R}(v) dv$$
$$Q_{85}(t) = \int_0^t e^{-(\eta_2+\eta_3)u} dR(u)$$
$$Q_{87}(t) = \eta_2 \int_0^t \int_0^v e^{-(\eta_2)u} du\, e^{-(\eta_3)v}\, dR(v) dv$$
$$Q_{95}(t) = \int_0^t G(u) du$$
$$Q_{10,7}(t) = \int_0^t e^{-(\eta_3)u} dR(u)$$
$$Q_{10,11}(t) = \int_0^t e^{-(\eta_3)u} \bar{R}(u) du$$
$$Q_{11,10}(t) = Q_{12,7}(t) = Q_{13,8}(t) = \int_0^t G(u) du$$

In transition probabilities by taking $t \to \infty$ we get the steady state probabilities i.e $p_{ij} = Q_{ij}(\infty) = \int_0^t Q_{ij}(t)$ and

it can be easily verified that $\sum_j p_{ij} = 1$ for all possible values of i, i.e

$p_{01} + p_{02} = 1$

$p_{10} = 1$

$p_{23}+p_{24} + p_{25} = 1$

$p_{32} + p_{36} = 1$

$p_{42} + p_{43}^{(6)} = 1$

$p_{57}+p_{58} + p_{59} = 1$

$p_{63} = 1$

$p_{75}+p_{7,10} + p_{7,12} = 1$

$p_{85} + p_{87}^{(10)} + p_{87}^{(11)} + p_{8,13} = 1$

$p_{95} = 1$

$p_{10,7} + p_{10,11} = 1$

$p_{11,10} = p_{12,7} = p_{13,8} = 1$

3.1 Mean Sojourn Times(MST)

The MST ϕ_i is the expected time taken by a system in a given state before transiting to any other state. If T_i is the sojourn time of S_i then MST ϕ_i is given by:

$$\phi_i = \int_0^\infty \mathrm{P}(\mathrm{T_i} > t)\mathrm{dt}$$

$$\phi_0 = \int_0^t e^{-(\eta_1+\delta_1)t}\, dt = \frac{1}{\eta_1+\delta_1}$$

$$\phi_1 = \int_0^\infty \bar{R}(t)dt$$

$$\phi_2 = \int_0^\infty e^{-(\eta_1+\eta_2+\delta_2)t}\, dt = \frac{1}{\eta_1+\eta_2+\delta_2}$$

$$\phi_3 = \int_0^\infty e^{-(\eta_1)t}\bar{T}(t)dt = \frac{1}{\eta_1}[1-\tilde{T}(\eta_1)]$$

$$\phi_4 = \int_0^\infty e^{-(\eta_2)t}\bar{R}(t)dt = \frac{1}{\eta_2}[1-\tilde{R}(\eta_1)]$$

$$\phi_5 = \int_0^\infty e^{-(\eta_1+\eta_2+\eta_3)t}\, dt$$

$$\phi_6 = \int_0^\infty \bar{R}(t)dt$$

$$\phi_7 = \int_0^\infty e^{-(\eta_2+\eta_3)t}\bar{T}(t)dt = \frac{1}{\eta_2+\eta_3}[1-\tilde{T}(\eta_2+\eta_3)]$$

$$\phi_8 = \int_0^\infty e^{-(\eta_2+\eta_3)t}\bar{R}(t)dt = \frac{1}{\eta_2+\eta_3}[1-\tilde{R}(\eta_2+\eta_3)]$$

$$\phi_9 = \phi_{11} = \phi_{12} = \phi_{13} = \int_0^\infty \bar{G}(t)dt$$

$$\phi_{10} = \int_0^\infty e^{-(\eta_3)t}\bar{R}(t)dt = \frac{1}{\eta_3}[1-\tilde{R}(\eta_3)]$$

4- Mean Time to System Failure

We define $\mathrm{T_i}$ as the time to system failure and $\pi_i(t)$ be the c.d.f. of the time to system failure for the first time when the system starts operation from state $\mathrm{S_i}$. We find the equations of $\pi_\mathrm{i}(\mathrm{t})$ for various values of i, by using regenerative point technique.

Therefore, MTSF for different states is given by:

$\pi_0(t) = Q_{01}(t) + Q_{02}(t)\circledR\pi_2(t)$

$\pi_2(t) = Q_{23}(t)\circledR\pi_3(t) + Q_{24}(t)\circledR\pi_4(t) + Q_{25}(t)\circledR\pi_5(t)$

$\pi_3(t) = Q_{32}(t)\circledR\pi_2(t) + Q_{36}(t)$

$\pi_4(t) = Q_{42}(t)\circledR\pi_2(t) + Q_{46}(t)$

$\pi_5(t) = Q_{57}(t)\circledR\pi_7(t) + Q_{58}(t)\circledR\pi_8(t) + Q_{59}(t)\circledR\pi_9(t)$

$\pi_7(t) = Q_{75}(t)\circledR\pi_5(t) + Q_{7,10}(t)\circledR\pi_{10}(t) + Q_{7,12}(t)\circledR\pi_{12}(t)$

$\pi_8(t) = Q_{85}(t)\circledR\pi_5(t) + Q_{8,10}(t)\circledR\pi_{10}(t) + Q_{8,13}(t)\circledR\pi_{13}(t)$

$\pi_9(t) = Q_{95}(t)\circledR\pi_5(t)$

$\pi_{10}(t) = Q_{10,7}(t)\circledR\pi_7(t) + Q_{10,11}(t)$

$\pi_{12}(t) = Q_{12,7}(t)\circledR\pi_7(t)$

$\pi_{13}(t) = Q_{13,8}(t)\circledR\pi_8(t)$

Taking L.S.T of above equations, and simplifying for $\tilde{\pi}_0(s)$,

$\tilde{\pi}_0(s) = \frac{N_1(s)}{D_1(s)}$

On taking $s \to 0$ in above equation and $\widetilde{\mathrm{Q}}_{\mathrm{ij}}(s) \to \mathrm{p_{ij}}$, as $s \to 0$, we have

$\tilde{\pi}_0(s) = \frac{N_1(s)}{D_1(s)} = 1$ implies that $\pi_0(t)$ is a proper cdf.

Therefore, MTSF is given by

$$E(T) = -\frac{d\tilde{\pi}_0(s)}{ds}\Big|_{s=0} = \frac{D_1'(0)-N_1'(0)}{D_1(0)}$$

Where,

$$D'_1(0) - N'_1(0) = \left[\phi_0\{1 - p_{23}p_{32} - p_{24}p_{42}\} + \phi_2 p_{02} + \phi_3 p_{23}p_{02} + \phi_4 p_{02}p_{24}\right] + \left[\left(1 - p_{7,10}p_{10,7} - p_{7,12}\right)\right]\left[\left(1 - p_{8,13}\right) - \{p_{58}p_{85} + p_{59}\left(1 - p_{8,13}\right)\}\right] - \left[p_{75}\left(p_{57}\left(1 - p_{8,13}\right) + p_{58}p_{8,10}p_{10,7}\right)\right] + \phi_5 p_{75}\left(1 - p_{8,13}\right)\left(1 - p_{23}p_{32} - p_{24}p_{42}\right) + \phi_7\{p_{57}\left(1 - p_{8,13}\right) + p_{58}p_{8,10}p_{10,7}\}\left(1 - p_{23}p_{32} - p_{24}p_{42}\right) + \phi_8 p_{58}\left(1 - p_{7,10}p_{10,7} - p_{7,12}\right)\left(1 - p_{23}p_{32} - p_{24}p_{42}\right) + \phi_9 p_{59}\left(1 - p_{8,13}\right)\left(1 - p_{7,10}p_{10,7} - p_{7,12}\right)\left(1 - p_{23}p_{32} - p_{24}p_{42}\right) + \phi_{10}\left[p_{7,10}\left[\left(1 - p_{8,13}\right) - \{p_{58}p_{85} + p_{59}\left(1 - p_{8,13}\right)\}\right] + p_{58}p_{8,10}\right]$$ (1- $p_{23}p_{32} - p_{24}p_{42}$)+ $\phi_{12}p_{7,12}\left[\left(1 - p_{8,13}\right) - \{p_{58}p_{85} + p_{59}(1 - p_{8,13})\}\right](1 - p_{23}p_{32} - p_{24}p_{42}) + \phi_{13}\left[(p_{8,13} - p_{59}p_{8,13})\left(1 - p_{7,10}p_{10,7} - p_{7,12}\right) - p_{57}p_{75}p_{8,13}\right](1 - p_{23}p_{32} - p_{24}p_{42})$

$$D_1(0) = (1 - p_{23}p_{32} - p_{24}p_{42})\left[\{1 - p_{7,10}p_{10,7} - p_{7,12}p_{12,7}\}\left\{\left(1 - p_{8,13}p_{13,8}\right) - \{p_{58}p_{85} + p_{59}(1 - p_{8,13})\}\right\} - \{p_{75}(p_{57}(1 - p_{8,13}) + p_{58}p_{8,10}p_{10,7})\}\right]$$

5- Availability Analysis

We define availability, $A_i(t)$ as the probability that the system is in upstate and is available to serve its required purpose when it initially start from regenerative state S_I

The availability equations for different states is given as:

$$A_0(t) = M_0(t) + q_{01}(t) \copyright A_1(t) + q_{02}(t) \copyright A_2(t)$$
$$A_1(t) = q_{10}(t) \copyright A_0(t)$$
$$A_2(t) = M_2(t) + q_{23}(t) \copyright A_3(t) + q_{24}(t) \copyright A_4(t) + q_{24}(t) \copyright A_4(t)$$
$$A_3(t) = M_3(t) + q_{32}(t) \copyright A_2(t) + q_{36}(t) \copyright A_6(t)$$
$$A_4(t) = M_4(t) + q_{42}(t) \copyright A_2(t) + q_{43}^{(6)}(t) \copyright A_3(t)$$
$$A_5(t) = M_0(t) + q_{01}(t) \copyright A_1(t) + q_{02}(t) \copyright A_2(t)$$
$$A_6(t) = q_{63}(t) \copyright A_3(t)$$
$$A_7(t) = M_7(t) + q_{75}(t) \copyright A_5(t) + q_{7,10}(t) \copyright A_{10}(t) + q_{7,12}(t) \copyright A_{12}(t)$$
$$A_8(t) = M_8(t) + q_{85}(t) \copyright A_5(t) + q_{87}^{(10)}(t) \copyright A_7(t) + q_{8,11}^{(10)}(t) \copyright A_{11}(t) + q_{8,13}(t) \copyright A_{13}(t)$$
$$A_9(t) = M_9(t) + q_{95}(t) \copyright A_5(t)$$
$$A_{10}(t) = M_{10}(t) + q_{10,7}(t) \copyright A_7(t) + q_{10,11}(t) \copyright A_{11}(t)$$
$$A_{11}(t) = q_{11,10}(t) \copyright A_{10}(t)$$
$$A_{12}(t) = M_{12}(t) + q_{12,7}(t) \copyright A_7(t)$$
$$A_{13}(t) = M_{13}(t) + q_{13,8}(t) \copyright A_8(t)$$

Taking the LT of above equations, we get the set of linear equations in $A_i^*(s)$ and solving for $A_0^*(s)$, we get

$$A_0^*(s) = \frac{N_2(s)}{D_2(s)}$$

The steady state availability is given as:

$A_0 = \lim_{s\to 0} s\, A_0^*(s) = N_2(0)/D_2'(0)$

where,

$$N_2(0) = \left[\{p_{32}(1 - p_{23}) - p_{24}p_{32}\}\phi_0 + \{p_{02}((\phi_2 + p_{24}\phi_4)p_{32} + (p_{23}p_{24}p_{43}))\}\right]\left[\{p_{10,7} - p_{7,10}p_{7,12}\}\{(1 - p_{8,13})(1 - p_{59}) - p_{58}p_{85}\} - \{p_{57}(1 - p_{8,13}) + p_{58}p_{87}^{(10)}\}(p_{75}p_{10,7})\right] + p_{02}p_{25}\left[\{(\phi_5 + p_{59}\phi_9)(1 - p_{8,13}) + p_{58}(\phi_8 + p_{8,13}\phi_{13})\}\left(p_{10,7}(1 - p_{7,12})\right) + \{p_{57}(1 - p_{8,13}) + p_{58}p_{87}^{(10)}\}\{(\phi_7 + p_{7,12}\phi_{12})(1 - p_{7,10})p_{10,7} + p_{7,10}\{\phi_{10} + p_{10,7}(\phi_7 + p_{7,12}\phi_{12})\}\}\right] \quad (1)$$

$$D_2'(0) = \left[\phi_5(1 - p_{8,13}) + \phi_8 p_{58}\right]p_{02}p_{25}p_{32}p_{75}p_{10,7} + \left[\phi_7\{p_{57}(1 - p_{8,13}) + p_{58}p_{87}^{(10)}\} + \phi_9 p_{59}p_{75}(1 - p_{8,13})\right]p_{02}p_{25}p_{32}p_{10,7} + \phi_{10}\left[(1 - p_{7,12})\{(1 - p_{8,13})(1 - p_{59}) - p_{58}p_{85}\} - p_{75}\{p_{57}(1 - p_{8,13}) + p_{58}p_{87}^{(10)}\}\right] p_{02}p_{32}p_{25} + \phi_{11}\left[p_{10,11}(1 - p_{7,12})\{(1 - p_{8,13})(1 - p_{59}) - p_{58}p_{85}\} - p_{75}p_{10,11}\{p_{57}(1 - p_{8,13}) + p_{58}p_{87}^{(10)}\}\right]p_{02}p_{32}p_{25} + \left[\phi_{12}\{(1 - p_{8,13})(1 - p_{59}) - p_{58}p_{85}\}p_{7,12} + \phi_{13}p_{75}p_{8.13}p_{58}\right]p_{10,7}p_{02}p_{25}p_{32} \quad (2)$$

6- Busy Period Analysis (BPA)

It is the probability that the repairman is occupied in repairing the failed component at time 't' given that the system has started from regenerative state S_i at $t = 0$. The BPA for different states is given by

$$B_0(t) = q_{01}(t) \copyright B_1(t) + q_{02}(t) \copyright B_2$$
$$B_1(t) = M_1 + q_{10}(t) \copyright B_0(t)$$
$$B_2(t) = q_{23}(t) \copyright B_3(t) + q_{24}(t) \copyright B_4 + q_{25}(t) \copyright B_5$$
$$B_3(t) = M_3 + q_{32}(t) \copyright B_2(t) + q_{36}(t) \copyright B_6(t)$$
$$B_4(t) = M_4 + q_{42}(t) \copyright B_2(t) + q_{43}^{(6)}(t) \copyright B_3(t)$$

$$B_5(t) = M_5 + q_{57}(t)\copyright B_7(t) + q_{58}(t)\copyright B_8(t) + q_{59}(t)\copyright B_9(t)$$

$$B_6(t) = M_6 + q_{63}(t)\copyright B_3(t)$$

$$B_7(t) = M_7 + q_{75}(t)\copyright B_5(t) + q_{7,10}(t)\copyright B_{10}(t) + q_{7,12}(t)\copyright B_{12}$$

$$B_8(t) = M_8 + q_{85}(t)\copyright B_5(t) + q_{87}^{(10)}(t)\copyright B_{10}(t) + q_{8,11}^{(10)}(t)\copyright B_{11}(t) + q_{8,13}(t)\copyright B_{13}(t)$$

$$B_9(t) = M_9 + q_{95}(t)\copyright B_5(t)$$

$$B_{10}(t) = M_{10} + q_{10,7}(t)\copyright B_7(t) + q_{10,11}(t)\copyright B_{11}(t)$$

$$B_{11}(t) = M_{11} + q_{11,10}(t)\copyright B_{10}(t)$$

$$B_{12}(t) = M_{12} + q_{12,7}(t)\copyright B_7(t)$$

$$B_{13}(t) = M_{13} + q_{13,8}(t)\copyright B_8(t)$$

Taking the LT of above equations and solving for $B_0^*(s)$,

$$B_0^*(s) = \frac{N_3(s)}{D_3(s)}$$

In steady state, BPA is given by

$$B_0 = lim_{s\to 0}\, sB_0^*(s)\frac{N_3(0)}{D_3'(0)}$$

$$N_3(0) = \phi_8 p_{58}p_{02}p_{25}p_{32}p_{75}p_{10,7} + \left[\phi_7\left\{p_{57}(1 - p_{8,13}) + p_{58}p_{87}^{(10)}\right\}\right]p_{02}p_{25}p_{32}p_{10,7} + \left[\phi_{10}(1 - p_{7,12})\left\{(1 - p_{8,13})(1 - p_{59}) - p_{58}p_{85}\right\} - p_{75}\left\{p_{57}(1 - p_{8,13}) + p_{58}p_{87}^{(10)}\right\}\right]p_{02}p_{32}p_{25} + \phi_{11}\left[p_{10,11}(1 - p_{7,12})\left\{(1 - p_{8,13})(1 - p_{59}) - p_{58}p_{85}\right\}p_{75}p_{10,11}\left\{p_{57}(1 - p_{8,13}) + p_{58}p_{87}^{(10)}\right\}\right]p_{02}p_{32}p_{25} + \left[\phi_{12}\left\{(1 - p_{8,13})(1 - p_{59}) - p_{58}p_{85}\right\}p_{7,12} + \phi_{13}p_{75}p_{8,13}p_{58}\right]p_{02}p_{25}p_{32}p_{10,7}$$

And $D_3'(0) = D_2'(0)$ given by (2)

7- Expected No. of Visits by Repairman

We denote $V_i(t)$ as the expected no. of visits by the repairperson to repair the failed unit. The expected no of visits for different states is given as:

$$V_0(t) = q_{01}(t) \copyright V_1(t) + q_{02}(t) \copyright V_2(t)$$

$$V_1(t) = q_{10}(t) \copyright V_0(t)$$

$$V_2(t) = q_{23}(t) \copyright V_3(t) + q_{24}(t)[1 + V_4(t)] + q_{25}(t) \copyright V_5(t)$$

$$V_3(t) = q_{32}(t) \copyright V_2(t) + q_{36}(t) \copyright V_6(t)$$

$$V_4(t) = q_{42}(t) \copyright V_2(t) + q_{43}(t) \copyright V_3(t)$$

$$V_5(t) = q_{57}(t)[1 + V_7(t)] + q_{58}(t)[1 + V_8(t)] + q_{59}(t)[1 + V_9(t)]$$

$$V_6(t) = q_{63}(t) \copyright V_3(t)$$

$$V_7(t) = q_{75}(t) \copyright V_5(t) + q_{7,10}(t) \copyright V_{10}(t) + q_{7,12}(t) \copyright V_{12}(t)$$

$$V_8(t) = q_{85}(t) \copyright V_5(t) + q_{8,13}(t) \copyright V_{13}(t) + q_{87}^{(10)}(t) \copyright V_7(t) + q_{8,11}^{(10)}(t) \copyright V_{11}(t)$$

$$V_9(t) = q_{95}(t) \copyright V_5(t)$$

$$V_{10}(t) = q_{10,7}(t) \copyright V_7(t) + q_{10,11}(t) \copyright V_{11}(t)$$

$$V_{11}(t) = q_{11,10}(t) \copyright V_{10}(t)$$

$$V_{12}(t) = q_{12,7}(t) \copyright V_7(t)$$

$$V_{13}(t) = q_{13,8}(t) \copyright V_8(t)$$

Taking the LT of above equations and solving for $V_0^*(s)$,

$$V_0^*(s) = \frac{N_4(s)}{D_4(s)}$$

In steady state, the no. of visits per unit time is given by

$$\mathrm{V_0(0) = \lim_{s\to 0} sV_0^*(s) = N_3(0)/D_4'(0)}$$

Where,

$$N_4(0) = \left[p_{01}\left\{(1 - p_{24}p_{42})p_{32} - \left\{p_{23} + p_{24}p_{43}^{(6)}\right\}p_{32}\right\} + p_{02}(p_{23} + p_{24} + p_{25})p_{32}\right]\left[p_{10,7}p_{7,5}\left(1 - p_{85}p_{58} - p_{59}(1 - p_{8,13})\right)\right] - \left\{p_{57}(1 - p_{8,13}) + p_{58}p_{87}^{(10)}\right\}p_{75}p_{10,11} + \left\{p_{02}p_{25}p_{32}(p_{57} + p_{58} + p_{59})(1 - p_{8,13})\right\}p_{10,7}p_{75}$$

And $D_4'(0) = D_2'(0)$ given in (2)

8- Profit Analysis

The profit encountered to the model in steady state by considering different costs and revenues, is calculated as:

$$P_1 = K_0A_0 - K_1B_0 - K_2V_0$$

Where

$\mathrm{K_0}$ = Revenue per unit up time of the system,
$\mathrm{K_1}$ = Cost per unit time for which the repair is busy,
$\mathrm{K_2}$ = Cost per unit visits by the repairman.

9- Behavior of Reliability Measures:

In this section, for practical illustration, the behavior of some important reliability measures has been examined for some specific values of parameters.

Table 1.Performance of η_1 and fixed parameters η_2,η_3, δ_1, δ_2,α_1, α_2 and $\alpha_3$on MTSF.

η_1	MTSF		
	$\eta_2 = 0.10$ $\eta_3 = 0.04$ $\delta_1 = 0.74$ $\delta_2 = 0.60$ $\alpha_1 = 0.47$ $\alpha_2 = 0.24$ $\alpha_3 = 0.33$	$\eta_2 = 0.08$ $\eta_3 = 0.02$ $\delta_1 = 0.79$ $\delta_2 = 0.69$ $\alpha_1 = 0.34$ $\alpha_2 = 0.39$ $\alpha_3 = 0.37$	$\eta_2 = 0.15$ $\eta_3 = 0.06$ $\delta_1 = 0.64$ $\delta_2 = 0.50$ $\alpha_1 = 0.37$ $\alpha_2 = 0.34$ $\alpha_3 = 0.39$
0.1	3.9904	5.2570	2.8029
0.2	2.9559	3.9690	2.3256
0.3	2.3585	3.2605	2.0005
0.4	1.9677	2.8004	1.7637
0.5	1.6917	2.4719	1.5834
0.6	1.4863	2.2227	1.4416
0.7	1.3276	2.0256	1.3271
0.8	1.2013	1.8648	1.2328
0.9	1.0985	1.7307	1.1537
1.0	1.0065	1.6612	1.0653

Table 2. Performance of α_1 and fixed parameters η_2, η_3, δ_1,δ_2,η_1, α_2 and $\alpha_3$on MTSF.

α_1	MTSF		
	$\eta_2 = 0.13$ $\eta_3 = 0.25$ $\delta_1 = 0.31$ $\delta_2 = 0.33$ $\eta_1 = 0.27$ $\alpha_2 = 0.03$ $\alpha_3 = 0.33$	$\eta_2 = 0.15$ $\eta_3 = 0.18$ $\delta_1 = 0.34$ $\delta_2 = 0.43$ $\eta_1 = 0.29$ $\alpha_2 = 0.04$ $\alpha_3 = 0.33$	$\eta_2 = 0.11$ $\eta_3 = 0.21$ $\delta_1 = 0.31$ $\delta_2 = 0.33$ $\eta_1 = 0.23$ $\alpha_2 = 0.02$ $\alpha_3 = 0.33$
0.1	1.8571	1.6701	2.1038
0.2	2.0716	1.8292	2.4664
0.3	2.1878	1.9227	2.6476
0.4	2.2646	1.9869	2.7667
0.5	2.3221	2.0350	2.8596
0.6	2.3680	2.0727	2.9397
0.7	2.4054	2.1019	3.0123
0.8	2.4351	2.1230	3.0792
0.9	2.4563	2.1352	3.1401
1.0	2.4677	2.1369	3.1938

Table 3. Performance of η_1 and fixed parameters η_2,η_3, δ_1, δ_2,α_1, α_2, α_3, k_0, k_1 and k_2 on Profit.

η_1	Profit		
	$\eta_2 = 0.81$ $\eta_3 = 0.95$ $\delta_1 = 0.91$ $\delta_2 = 0.39$ $\alpha_1 = 0.56$ $\alpha_2 = 0.32$ $\alpha_3 = 0.33$ $k_0 = 1000$ $k_1 = 100$ $k_2 = 50$	$\eta_2 = 0.79$ $\eta_3 = 0.93$ $\delta_1 = 0.91$ $\delta_2 = 0.39$ $\alpha_1 = 0.55$ $\alpha_2 = 0.35$ $\alpha_3 = 0.33$ $k_0 = 1000$ $k_1 = 120$ $k_2 = 90$	$\eta_2 = 0.89$ $\eta_3 = 0.97$ $\delta_1 = 0.81$ $\delta_2 = 0.32$ $\alpha_1 = 0.54$ $\alpha_2 = 0.33$ $\alpha_3 = 0.33$ $k_0 = 950$ $k_1 = 120$ $k_2 = 90$
0.1	2.5482	2.4729	2.8455
0.2	2.1945	2.2126	2.4681
0.3	1.8936	2.0004	2.1495
0.4	1.6221	1.8157	1.8639
0.5	1.3656	1.6459	1.5954
0.6	1.1151	1.4828	1.3339
0.7	0.8643	1.3212	1.0724
0.8	0.6091	1.1574	0.8061
0.9	0.3463	0.9888	0.5318
1.0	0.0737	0.8134	0.2468

Table 4. Performance of α_1 and fixed parameters η_2,η_3, δ_1, δ_2,η_1, α_2, α_3, k_0, k_1 and k_2 on Profit.

α_1	Profit		
	$\eta_2 = 0.65$ $\eta_3 = 0.59$ $\delta_1 = 0.65$ $\delta_2 = 0.49$ $\eta_1 = 0.47$ $\alpha_2 = 0.69$ $\alpha_3 = 0.55$ $k_0 = 950$ $k_1 = 120$ $k_2 = 90$	$\eta_2 = 0.69$ $\eta_3 = 0.62$ $\delta_1 = 0.55$ $\delta_2 = 0.49$ $\eta_1 = 0.45$ $\alpha_2 = 0.72$ $\alpha_3 = 0.56$ $k_0 = 900$ $k_1 = 150$ $k_2 = 80$	$\eta_2 = 0.61$ $\eta_3 = 0.55$ $\delta_1 = 0.54$ $\delta_2 = 0.39$ $\eta_1 = 0.35$ $\alpha_2 = 0.75$ $\alpha_3 = 0.59$ $k_0 = 1000$ $k_1 = 100$ $k_2 = 50$
0.1	3.59827	4.17309	3.80843
0.2	3.67789	4.37047	3.85116
0.3	3.83764	4.68783	3.99023
0.4	4.08536	5.15737	4.23971
0.5	4.44261	5.84607	4.63524
0.6	4.95729	6.89995	5.2584
0.7	5.739	8.68624	6.31344
0.8	7.08058	12.4277	8.45213
0.9	10.0689	25.9994	15.3405
1.0	11.1245	30.3421	20.4563

From Table 1, it has been observed that as failure rate "η_1" increases, keeping the other parameters η_2 , η_3 , δ_1 ,

δ_2 , α_1, α_2 and α_3 fixed, mean time to system failure decreases. This implies that expected life of the system decreases as failure rate increases. Thus we conclude that mean time to system failure can be increased by reducing failure rate. From Table 2, it has been seen that as repair rate "α_1" increases, keeping the values of other parameters η_2,η_3, δ_1, δ_2,$\alpha_1, \eta_1$and α_3 fixed, the mean time to system failure increases. This implies that expected life of the system increases with increasing repair rate. Further, from Table 3, it has been observed that profit of the model decreases as failure rate "η_1" increases, irrespective of the other parameters η_2,η_3, δ_1, δ_2,α_1, α_2, α_3, k_0, k_1 and k_2. Hence it can be concluded that expected life of the system can be increased by reducing failure rate. Also from Table 4, it has been noticed that the profit of the model increases as repair rate "α_1" increases, keeping the other parameters η_2 , η_3 , δ_1 , δ_2 , α_1, η_1 and α_3 fixed. This implies that expected life of the system increases as repair rate increases.

10- Conclusion

In this paper, the profit analysis of the model comprising of three non-identical units has been obtained. The reliability technique of priority of one unit over other has been used. Expressions for various important reliability measures have been obtained using regenerative point technique. The numerical illustration reveals that MTSF and profit of the system decreases with increase in the failure rate. Further MTSF and profit of the model increases as repair rate increases. Thus we conclude that expected lifetime of the model can be enriched by either increasing the repair rate of the unit or by reducing the failure rate of the unit which in turn will enhance the performance and hence efficiency of the model. The study advocates that some preventive measures should be taken to avoid failure of the system, for that the priority strategy of reliability measure should be taken into consideration.

The results obtained in this research paper are useful for system planners who are engaged in designing highly reliable system. This is significant because it considers an actual problem and provides a unique conceptual model as a starting point for organizations to foster higher levels of engagement and check the efficiency and effectiveness of the system. It is Novel Research and good topic for research for manufacturers, designers and users.

References

[1] A. Kumar and S. C. Malik, "Reliability Analysis of a Redundant System with 'FCFS' Repair Policy Subject to Weather Conditions", International Journal of Advanced Science and Technology ,Vol. 29, No. 03, 2020, pp. 7568 – 7578.

[2] A. Kumar, D. Pawar, and S. C. Malik, "Profit analysis of a warm standby non-identical unit system with single server performing in normal/abnormal environment," Life Cycle Reliab Saf Eng., doi 10.1007/s41872-019-00083-2. 2019.

[3] Chib, Rakesh, J.P. S. Joorel, and V. Sharma "Profit Analysis of a Two Unit Priority System with Two Types of Repair Facility", International Journal of Statistics and Reliability Engineering, Vol. 1, No. 1, 2014, pp. 69-81

[4] I. kumar, A. Kumar, and M. Saina, "Analysis of performance measures of computer systems with priority and maximum operation time," Advances in intelligent systems and computing,Springer Singapore, vol 933, 2019.

[5] J.K. Sureria and S.C. Malik "Reliability Measures of a Computer System with Arrival Time of the Server and Priority to H/w Repair over S/w Up-gradation", International Journal of Computer Applications, Vol. 65, No.13, 2013, pp. 44-48.

[6] M.S. Kadyan, S.C.Malik and J.Kumar, "Cost-Analysis of a System under Priority to Repair and Degradation" International Journal of Statistics and Systems, Vol. 5, No. 1, 2010, pp. 1-10.

[7] N. Bashir, R. Bashir and T R Jan, "Effectiveness Analysis of Two Non-Identical Unit System Model with Priority Unit Subject to Degradation and Inspection Facility", IEEE International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 2019, pp. 40-45. ISBN 978-1-7281-3778-0

[8] N. Gorjian Jolfaei , B. Jin , L. van der Linden , I. Gunawan and N. Gorjian , "Reliability modelling with redundancy—A case study of power generation engines in a wastewater treatment plant", Qual Reliab Engng Int. 2019, pp. 1–13. https://doi.org/10.1002/qre.2573

[9] R. Rathee and S. Chander "A Parallel System with Priority to Repair over Preventive Maintenance Subject to Maximum Operation and Repair Times", International Journal of Statistics and Reliability Engineering, Vol. 1, No. 1, 2014, pp. 57-68.

[10] R. Sharma and J.P.S. Joorel, "Reliability modelling and cost benefit analysis of fruit juice manufacturing system". Mathematics in Engineering, Science and Aerospace, Vol. 6, No. 2, 2015, pp. 281-293.

[11] S.Chander "Reliability models with priority for operation and repair with arrival time of server", Pure and Applied Mathematika Sciences, Vol. XI, No. 1-2, 2005, pp. 9-22.

[12] S.C.Malik, M.S. Kadian, and J. Kumar , "Cost-Analysis of a System under Priority to Repair and Degradation", International Journal of Statistics and Systems, Vol. 5,No.1, 2010, pp. 1 – 10.

[13] S.C. Malik, "Reliability Modeling of a computer System with Preventive Maintenance and Priority Subject to Maximum Operation and Repair Times", International Journal of System Assurance Engineering and Management, Vol. 4(1), 2013, pp. 94-100.

[14] V.J. Munday and S.C. Malik, "Reliability Measures of a Computer System with Priority for Repair and Hardware Redundancy" , International Journal of Computer Applications, Vol. 125, No.8, 2015, pp. 32-37.
[15] Y.L.Zhang and G.J. Wang, "A Geometric Process Repair Model for a Repairable Cold Standby System with Priority in Use and Repair", Journal of Reliability Engineering and System Safety, Vol. 94, 2009, pp. 90-100.
[16] M.S. Barak , Neeraj, and S.K. Barak "Proft analysis of a two-unit cold standby system model operating under different weather conditions", Life Cycle Reliability and Safety Engineering, Springer, 2018, https://doi.org/10.1007/s41872-018-0048-6
[17] O. Dahiya, A. Kumar, and M. Saini, "Modeling and analysis of concrete mixture plant subject to coverage factor and profust reliability approach", Life Cycle Reliability and Safety Engineering, Springer, 9, 2020, pp. 273–281, https://doi.org/10.1007/s41872-019-00104-0
[18] D. Goyal, A. Kumar, M. Saini, and H. Joshi. "Reliability, maintainability and sensitivity analysis of physical processing unit of sewage treatment plant", SN Applied Sciences, Springer, 2020. https://doi.org/10.1007/s42452-019-1544-7
[19] N. Gupta, M. Saini, and A. Kumar, "Operational availability analysis of generators in steam turbine power plants" SN Applied Sciences, Springer, 2020. https://doi.org/10.1007/s42452-020-2520-y

Ms. Nafeesa Bashir received Masters degree in Statistics from University of Kashmir, J&K, India. Currently she is a Ph.D. research Scholar in University of Kashmir. She has to her credit several research publications both in national and international journals of repute. Her research interest include Reliability Modeling and Probability Distributions.

Dr. Raeesa Bashir (Ph.D. Statistics) from University of Jammu, India. She is an Assistant Professor, in Mathematics and Quantitative Analysis in the Engineering and Management Department at Amity University Dubai with many years of experience in teaching and research. She has to her credit several articles and research papers in reputed journals. She has previously worked in government and private organizations in Kingdom of Saudi Arabia and UAE.

Prof J P Singh Joorel is a Professor of Statistics in the Faculty of Mathematical Sciences at University of Jammu, Jammu, India and presently heading the INFLIBNET Centre (An Autonomous Inter University Centre of University Grants Commission, Govt of India), Gandhinagar, Gujarat as Director. He received his Doctorate degree in Statistics from Institute of Social Sciences, Agra University, Agra, India in 1989. He has published work in international and national journals of repute. His areas of interest are Reliability Theory, Sample Surveys and Statistical Inference.

Dr. Tariq Rashid Jan is an Associate Professor in the Department of Statistics, University of Kashmir, J&K, India. He did his M. Phil. in Statistics from Department of Statistics and Operations Research, Aligarh Muslim University, Aligarh, India. He received his Ph. D. degree in Statistics from Department of Statistics, University of Kashmir, Srinagar, India in 2004. He has published work in international and national journals of repute. His current research interests are in Reliability Theory, Bio –Statistics and Generalized Distributions.

DeepFake Detection using 3D-Xception Net with Discrete Fourier Transformation

Adeep Biswas
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore
adeep.biswas2016@vitstudent.ac.in
Debayan Bhattacharya
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore
debayan.bhattacharya2016@vitstudent.ac.in
Kakelli Anil Kumar*
Associate Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore
anilsekumar@gmail.com

Received: 18/Aug/2020 Revised: 28/March/2021 Accepted: 01/June/2021

Abstract
The videos are more popular for sharing content on social media to capture the audience's attention. The artificial manipulation of videos is growing rapidly to make the videos flashy and interesting but they can easily misuse to spread false information on social media platforms. Deep Fake is a problematic method for the manipulation of videos in which artificial components are added to the video using emerging deep learning techniques. Due to the increase in the accuracy of deep fake generation methods, artificially created videos are no longer detectable and pose a major threat to social media users. To address this growing problem, we have proposed a new method for detecting deep fake videos using 3D Inflated Xception Net with Discrete Fourier Transformation. Xception Net was originally designed for application on 2D images only. The proposed method is the first attempt to use a 3D Xception Net for categorizing video-based data. The advantage of the proposed method is, it works on the whole video rather than the subset of frames while categorizing. Our proposed model was tested on the popular dataset Celeb-DF and achieved better accuracy.

Keywords: Computer Vision; DeepFake Detection; Xception Net; Video Manipulation.

1- Introduction

In recent times, the usage of videos has increased rapidly for different purposes such as marketing, news, and entertainment [1]. Along with the growing popularity of video-based content, some major social media platforms have come up as well and while these platforms have many benefits, they do not regulate or verify the videos being posted on their platform to a large extent [2]. Due to limited regulations and control systems, the users can easily manipulate videos artificially for malicious purposes such as spreading false political propaganda or causing disruptions in the financial market through spreading false rumours and fake information about military and research organizations by manipulating the satellite videos [3]. Due to such possibilities of malicious use, detecting such videos has become a serious issue [4]. One of the most common types of artificial videos is DeepFake. DeepFakes are videos or images which have been generated artificially using deep learning models [5]. DeepFakes rely on two types of artificial manipulation of videos mainly face swapping and facial re-enactment. Face swapping involves the replacement of a person's face in an image or a video with another face [6]. On the other hand, facial re-enactment is the process of creating the artificial head, lips, or any other facial feature movement to create a false narrative about a person's speech or expressions [7]. Both are artificial manipulations that are very harmful to social media users.

Deepfake generation technique is perfect and has some weaknesses that can be exploited to detect them. It is essential to determine various factors that can be used to differentiate a deepfake video from a genuine one. Some factors or features include the frequency of a blinking of eyelids in the video or even the anomalies in the head movement of a person [8]. The deepfake generation techniques have grown to be more sophisticated and accurate as well. The existing research work has focused to determine more robust detection techniques for deepfakes and accordingly, various newer techniques have been suggested such as the analysis of the colour hues in the video, the difference in the neural activation behaviour of the detection model, and the discrepancies in the convolutional traces of the video among other things [5].

* Corresponding Author

Although such newer detection techniques have achieved better accuracy results, it is essential to introduce advanced methods for better detection techniques continues. Due to the constantly evolving nature of deepfake generation, it is essential to introduce emerging detection techniques to overcome the challenges of the future [4]. The proposed work has proposed a new deepfake detection technique-3D Inflated Xception Net with Discrete Fourier Transformation which was able to achieve state-of-the-art accuracy results. To validate the performance of the proposed model, the publicly available deepfake benchmark dataset called Celeb-DF was used [9, 24]. Celeb-DF is a large-scale and high-quality deepfake dataset containing 5639 videos generated using various deep generation techniques [8, 9]. This data set is useful for the performance evaluation of various proposed algorithms/techniques for DeepFake detection and analysis. The main contribution of the method proposed work is it uses a 3D convolutional neural network model which takes the whole video as an input rather than extracting a subset of features from videos and using them as input parameters for categorization. Xception is proposed, and able to outperform most other pre-existing models in terms of accuracy as well as computational costs [10]. Despite having such promising results, its architecture has remained two-dimensional and relies on specific image frames extracted from a video for its categorization. The main drawback of such an approach is if the right video frame is not selected for the input, the video may get incorrectly categorized since artificial manipulations don't need to be done to all frames of the video when a deepfake is generated. This problem has been addressed in our proposed model by converting the 2-dimensional architecture of Xception net into 3-dimensions and initializing the network by pre-training it on static videos generated from a subset of images of the ImageNet dataset [25]. Furthermore, our model takes a two-stream approach and combines the results of the 3D Xception net with the results of a Discrete Fourier Transformation based classifier to account for all the parameters present in the spatial, temporal as well as frequency domains to achieve better accuracy results [26, 27].

The rest of the paper is arranged in the following manner- section 3 discusses the relevant research work which has already been done in this domain, section 4 illustrates details of the deepfake detection model proposed in this paper, section 5 specifies the configuration of our experimental setup and the results that were achieved by the proposed model and finally in section 6 we provide the conclusion and discuss the further scope for research work in this domain in the future.

2- Related Work

Significant breakthroughs have been made over the years now [28], there is still a lot of scope of improvements to handle the challenges in the domain of deepfake detection [29].

2-1- GAN Based Deepfake Generation

Generative Adversarial Networks (GAN) were introduced in 2014 and one of the most popular methods of generating deepfake videos [11, 30]. The main reason why GAN-based deepfake generation techniques have become important is they can adapt automatically and overcome any biases or weaknesses present in the network's generation process [12, 31].

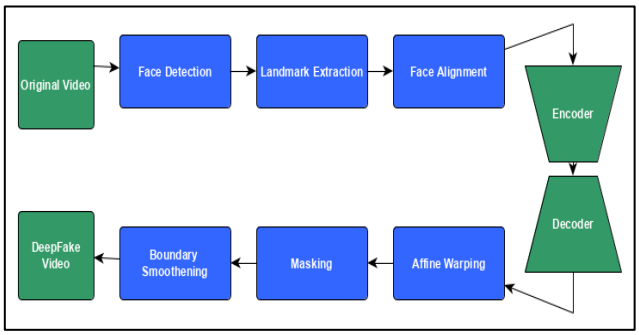


Fig. 1. DeepFake Synthesis Process

GAN has two major components, a generator, and a discriminator, both of which compete against each other for best results [32]. This works on any deepfake video outputted from the generator is passed to the discriminator which would try to determine whether the video inputted is fake or real and subsequently give feedback to the generator [13]. If the discriminator can classify the videos with high accuracy, it means that the generator function has some weaknesses that can be exploited. Based on this, the generator function would change its architecture and produce better deepfakes. On the other hand, if the discriminator is not able to make predictions accurately, then the generated deepfakes are of high quality and not easily distinguishable. Due to this constant process of feedback and improvement, GAN can overcome most detection methods eventually, causing those detection techniques to become obsolete [4]. Some such popular GAN methods proposed by researchers include AttGAN, StarGAN, StyleGAN, and PGGAN [11]. Due to the ability of GANs to change their generative architecture, traditional deepfake detection methods do not work on it and researchers are slowly moving towards a more forensic-based approach to counter GANs using the difference in color cues and pixel distribution [13]. The faces are targeted and detected in GAN Based Deepfake

Generation from the input video. The faces are aligned to the defined configuration using Landmark extraction and face alignment techniques as shown in figure 1. The encoder is used to detect the similarity of the facial expressions of the images or videos w. r.to target images. The GAN encoder and decoder are trained models, automated, the decoder can decode the target the facial expressions and also limit the re-construction errors. The targeted faces are wrapped, masking and boundary smoothing to the predefined configurations of original faces of the input images or video.

2-2- Existing DeepFake Detection Algorithms

The development of robust deepfake detection techniques has gained momentum in recent years due to their growing need. Some advanced deepfake detection methods have been presented to show the advancement and the variety of the methods. Li et al [14] proposed the detection of artificial manipulations in a video by locating the blending boundary between the combined real and fake potions of the video frames. Amerini et al [15] suggest a method of deepfake detection that relies on computing the optical flow of the given video. Fernando et al [16] put forth a technique of detecting deepfakes by utilizing memory networks to replicate human-like cognition of the social context of the video. Another interesting approach suggested by Venkatesh et al [17] is to detect a deepfake video by concentrating on the presence of morphing within its video frames. For this purpose, a form of context aggregation network is put forward. Jeon et al [18] have prioritized the computational cost of deepfake detection over its accuracy by proposing a lightweight neural network architecture that can utilize pre-existing and pre-trained classifier models. Zhang et al [19] has introduced a unique approach to tackling the deepfake problem by analyzing the difference in image compression ratios of the multiple video frames or images that are blend together through error level analysis. Dang et al [20] demonstrate a way of increasing the accuracy scores of deepfake detection classifiers by using eXtreme Gradient Boosting algorithm. Guera et al [35] has proposed a machine learning-based tool for automatic detection of manipulation traces in videos using CNN. Recurrent neural network (RNN) has used feature extraction and classification to find video manipulation. Afchar et al [36] has proposed an efficient automatic deep learning method for detect facial tampering in videos using Face-2-face and Deepfake. The work has achieved 95 to 98% accuracy using the above-mentioned methods. Sohrawardi et al [37] have proposed an efficient and robust system using artificial intelligence techniques for Deepkfake detection. Albahar et al [38] have analyzed the impact of the Deepfake in society hence they have introduced new techniques based on digital watermarking, facial detection techniques, and convolutional neural networks (CNNs). Using machine learning techniques, the method has achieved better accuracy in the detection of Deepfake videos. The proposed mechanism has achieved better results in terms of accuracy and efficiency in the detection of Deepfake. Therefore, from the review of these existing works, the problem of deepfake detection can be solved by using various techniques or parameters. The main challenge lies in determining these parameters, and other ways of improving the already existing detection technique.

This work deals with a modified version of Xception and the original Xception architecture proposed by Google researchers in 2017 [10]. Xception is a particular format of architecture or a particular way of arrangement of the different layers of activation present in a convolutional neural network model. This Xception model was created by modifying the previously benchmark CNN model called Inception and replacing its inception modules with depth-wise separable convolutions which made the network architecture more efficient and allowed a higher level of accuracy for the same number of input parameters. The main advantage of Xception over other deepfake detection techniques can achieve comparable and better accuracy results than the other existing detection techniques. The Xception is tested on popular datasets like FaceForenics [21] and DeeperForensics [22] through a 2-dimensional network in contrast to 3-dimensional or involving computationally expensive recurrent layers [33, 34]. Hence, Xception is an efficient algorithm and has the scope for better accuracy scores to make a perfect model for future purposes [37, 38].

3- Proposed Methodology

In this paper, a new algorithm for the detection of deepfake videos is proposed which involves 3D Inflated Xception Net with Discrete Fourier Transformation. The model takes in the whole video as an input and categorizes it in one of the two possible classifications namely fake or real based on the combined results of the 3D Xception stream and the 3D DFT stream. This allows the network to capture the overall spatial as well as frequency domain representations of the video from which local features. The local features are extracted subsequently in the convolutional layers and facilitating the classification while considering all parameters [39, 40]. The complete pipeline of the proposed algorithm including the data pre-processing and the 3D Xception and 3D DFT streams are presented in detail in the subsequent subsections.

3-1- Data Pre-Processing

The data pre-processing involved inputting the raw video into the 3D models to convert the videos into a uniform format with fixed dimensions for inputting into the model automatically one by one. It is necessary because the architecture of the 3D CNN used in our model needs inputs of fixed size. The raw video needed to be converted into a format that could be readable by the CNN and allow it to perform convolutional and padding operations. Initially, all video frames were extracted from the videos. Any video having lesser than 30 frames was discarded since they were too short. For the remaining videos, 30 frames were selected from the video. After that, each video frame is cropped into height and width dimensions of 299 by 299 pixels respectively. The cropped video frames are then converted into Numpy arrays and appended with each other to create 3-dimensional matrices of size 30 x 299 x 299 which could finally be inputted into the main classifier models.

3-2- Network Architecture

The model consists of two parallel streams- one consisting of the Xception model while the other one consisting of the Fourier transformation-based classifier. The overall structure is as shown in the figure 2.

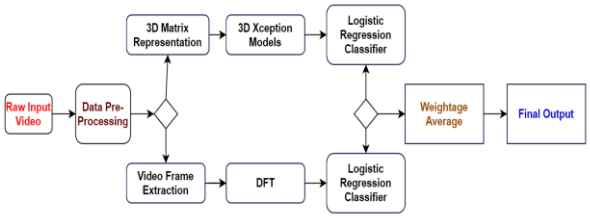


Fig. 2. Proposed Pipeline for DeepFake Detection

The Xception stream forms the first part of the pipeline which contains the 3D Xception model. A modified version of the Xception model is chosen because the original Xception architecture is a benchmark model for deepfake detection and outperforms other CNN architectural configurations for deepfake detection purposes. Furthermore, it showed as 2-dimensional, it showed great potential for being capable of being further enhanced by adding a third dimension to account for the temporal attributes of a video. The configuration of this model is the same as specified in the original Xception [2]. The only change is to the configuration instead of using 2-dimensional layers inside the CNN, they have been replaced with 3-dimensional convolutional layers by adding an extra dimension for the depth of the video to the already existing height and width dimensions. The newly constructed model was initialized randomly and pre-trained on inflated static videos formed from a subset of the ImageNet dataset. ImageNet dataset was chosen because it is one of the largest and most exhaustive datasets of images and has a demonstrated history of being a suitable option for pre-training models without imparting any biases to them. Furthermore, it needs to be noted that since ImageNet consists of only 2-dimensional images, these images had to first be converted into inflated videos by appending the same image 30 times to create a static video from the inflation of those given images. These inflated videos were then used to pre-train the Xception model to impart better weight initializations to the model than simple random initializations. The deepfake dataset called Celeb-DF was introduced to the model for training and testing purposes after completing pre-training of the model.

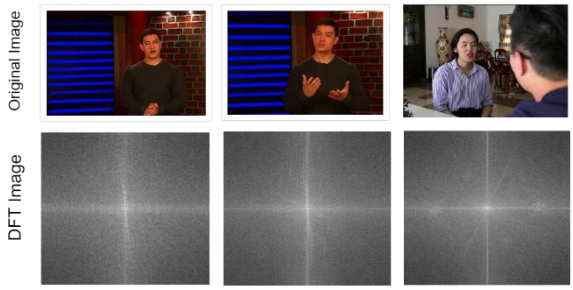


Fig. 3. Video Frames extracted from the Celeb-DF and their corresponding DFT Visualizations

The second stream consists of the Discrete Fourier Transformation module which is introduced to do a parallel frequency domain analysis on the video. The reason behind using a separate stream for frequency analysis of the videos is the most artificially generated videos can replicate the spatial properties of a video or an image. The replicating frequency and amplitude distribution of a genuine video is more difficult and often unaccounted the generation model and can be exploited to determine whether the video for artificially manipulated or not. For performing DFT transformation on the input videos, each of the video frames extracted from the raw videos during the data pre-processing phase is passed through a 2D discrete Fourier transformer and the output of these transformations is then appended to get an overall frequency domain representation. As shown in figure 3. Finally, this representation is compressed into a 2D matrix and then fed into a Logistic Regression algorithm-based classifier to get the output from the DFT stream. The logistic regression model is the best fit model as an exemplary classifier for binary and multi-class classification. The results of the proposed method have

shown that our hybrid classifier DFT with logistic regression model can differentiate the video samples of variable sizes with multiple features as real or fake. Once the output is given by the two streams, their outputs are combined to get the final result. The combination is done by taking a weighted average of the probability of the video being fake as predicted by the Xception stream and the DFT stream. Higher weightage is given to the output of the Xception model since it has a higher accuracy performance when used individually. The final probability is obtained from taking the weighted average of the individual probabilities of the two streams determines whether the video is fake or real by comparing it with a pre-determined threshold probability value.

4- Experimental Results

In this section, the experiments which are conducted to determine the performance of our proposed methodology are discussed in detail. This involves the overall experimental setup and resources used to achieve our results, the analysis of the outcome of the testing done on our proposed methodology as well as our model's performance comparison to other deepfake detection algorithms.

4-1- Experimental Setup

Due to the large size of video data as well as the high number of convolutional layers involved in the 3D Xception model, the training process of the prediction model was highly computationally expensive and hence, a 16GB GPU is used to run the code in our system. The rest of the configurations on which the code was run includes i7 processor, 8GB in-built RAM along an Ubuntu operating system environment. The complete implementation of the model is done on Python and Pytorch library is used for the implementation of the main 3D Xception module. The model is trained over 30 epochs to attain the final accuracy results. Besides the running environment, a publicly available benchmark dataset called Celeb-DF is used to test the performance of our proposed model. This particular dataset is chosen because it has a relatively large size containing 5639. Furthermore, the deepfake videos present in this dataset are of higher quality than those in previously available datasets and involve subjects of vast variations in terms of gender, age, and ethnicity, making this dataset suitable for replicating real-world deepfakes. A sample of this dataset is shown in Figure 4 where the original video and the deepfake were generated from it by performing artificial face-swapping.

Fig. 4. Real Video and the corresponding DeepFake generated

4-2- Results and Analysis

The performance of the model is measured using Area Under Curve (AUC) score of the Receiver Operating Characteristics graph since it is the standard parameter used for measuring the performance of the previous deepfake detection algorithms as well. Three different configurations of the proposed methodology are tested and their corresponding ROC curve and accuracy scores are presented in figure 5 and Table 1 respectively. The first configuration involves using the 3D Xception module directly without pre-training it on the ImageNet dataset or adding the DFT module to it. In the second case, the 3D Xception model is first pre-trained on ImageNet before training and testing it using the Celeb-DF dataset. As seen in the graph in Figure 5, the pre-training of the model significantly improves the performance of the model. Lastly, the pre-trained 3D Xception module is combined with the DFT module to further enhance the performance of the overall pipeline. Lastly, we compare the result achieved by the proposed methodology with the best AUC score of other prominent deepfake detection algorithms on the same Celeb-DF dataset [23]. From Table 2, our model's AUC score is among the best and comparable to some of the most advanced deepfake detection techniques. An important aspect here is that methods can perform 3D computations and those which are introduced to the Celeb-DF dataset during their training phase perform significantly better than the 2-dimensional algorithms.

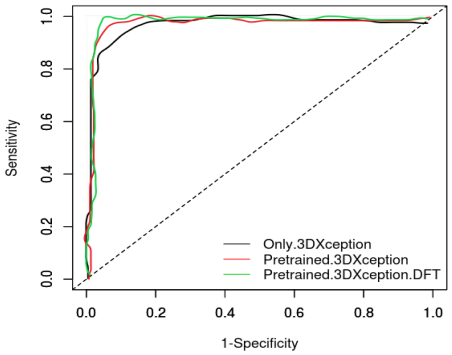


Fig. 5. ROC Curve for different configurations

Table 1. ROC-AUC % and Accuracy % of different configurations of the proposed methodology

Method	**ROC-AUC%**	**Accuracy %**
3D Xception	94.67	93.29
3D Xception (Pre-Trained with Image Net)	98.13	96.91
3D Xception- DFT (Pre- Trained with Image Net)	98.81	97.66

Table 2. ROC-AUC % Comparison of the Proposed Algorithm with other State-of-the-art DeepFake Detection Algorithms

Method	**Dimension**	**CALEB-DF Training**	**ROC-AUC %**
Two Stream	2D	NO	53.8
MESO4	2D	NO	54.8
MESOInception4	2D	NO	53.6
HEADPose	2D	NO	54.6
FWA	2D	NO	56.9
VA-MLP	2D	NO	55
VALogReg	2D	NO	55.1
Xception-raw	2D	NO	48.2
Xception-c23	2D	NO	65.3
Xception-c40	2D	NO	65.5
Mutli task	2D	NO	54.3
Capsule	2D	NO	57.5
DSP-FWA	2D	NO	64.6
DFT	3D	YES	66.8
X ception-Metric-Learning	3D	YES	99.2
RCN	3D	YES	74.87
R2Plus1D	3D	YES	99.43
I3D	3D	YES	97.59
MC3	3D	YES	99.3
R3D	3D	YES	99.73
3D Xception-DFT (The Proposed)	**3D**	**YES**	**98.81**

5- Conclusion and Future Scope

In this paper, we presented a new model for enhancing the performance of the pre-existing Xception algorithm. This was achieved by converting the whole architecture from being 2-dimensional into 3-dimensional space which is more suitable for handling the additional time-based dimension of videos. Furthermore, the paper also proposed a pipeline to combine the 3D Xception module with a frequency domain-based Fourier transformation model to achieve better results in terms of accuracy. Therefore, the model proposed in this paper accounts for all spatial, temporal as well as frequency domain-based parameters and hence can achieve results comparable to the existing state-of-the-art deepfake detection algorithms. The limitation of the proposed methodology is it increases the computational cost and complexity of the deepfake detection process. Thus, a common problem of the trade-off between efficiency and accuracy is created, each having its own merits for a particular use case. The future scope of this work would involve finding a way to make the proposed model more efficient while retaining the same level of accuracy. In real-time applications, speed is an important factor for the model to detect fake videos.

References

[1] Kumar, P., Vatsa, M., & Singh, R. Detecting face2face facial reenactment in videos. In The IEEE Winter Conference on Applications of Computer Vision, IEEE, 2020, pp. 2589-2597.

[2] Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., & Natarajan, P. Recurrent convolutional strategies for face manipulation detection in videos. Interfaces (GUI), 2019, vol 3(1).

[3] Nguyen, T. T., Nguyen, C. M., Nguyen, D. T., Nguyen, D. T., & Nahavandi, S. Deep learning for deepfakes creation and detection, 2019, arXiv preprint arXiv:1909.11573.

[4] Lyu, S. Deepfake detection: Current challenges and next steps. In 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), IEEE, 2019, pp. 1-6.

[5] Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. Deepfakes and beyond: A survey of face manipulation and fake detection. 2020, arXiv preprint arXiv:2001.00179.

[6] Bitouk, D., Kumar, N., Dhillon, S., Belhumeur, P., & Nayar, S. K. Face swapping: automatically replacing faces in photographs. In ACM SIGGRAPH 2008 papers, 2008, pp. 1-8.

[7] Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., & Nießner, M. Face2face: Real-time face capture and reenactment of rgb videos. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016 pp. 2387-2395.

[8] Tolosana, R., Romero-Tapiador, S., Fierrez, J., & Vera-Rodriguez, R. DeepFakes Evolution: Analysis of Facial Regions and Fake Detection Performance, 2020, arXiv preprint arXiv:2004.07532.

[9] Li, Y., Yang, X., Sun, P., Qi, H., & Lyu, S. Celeb-df: A new dataset for deepfake forensics, 2019, arXiv preprint arXiv:1909.12962.

[10] Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251-1258.

[11] Huang, Y., Juefei-Xu, F., Wang, R., Xie, X., Ma, L., Li, J., ... & Pu, G. FakeLocator: Robust Localization of GAN-Based Face Manipulations via Semantic Segmentation Networks with Bells and Whistles, 2020, arXiv preprint arXiv:2001.09598.

[12] Nirkin, Y., Keller, Y., & Hassner, T. FSGAN: Subject agnostic face swapping and reenactment. In Proceedings of the IEEE international conference on computer vision, 2019, pp. 7184-7193.

[13] McCloskey, S., & Albright, M. Detecting gan-generated imagery using color cues, 2018, arXiv preprint arXiv:1812.08247.

[14] Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., & Guo, B. Face x-ray for more general face forgery detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5001-5010.

[15] Amerini, I., Galteri, L., Caldelli, R., & Del Bimbo, A. Deepfake video detection through optical flow based cnn. In Proceedings of the IEEE International Conference on Computer Vision Workshops, 2019.

[16] Fernando, T., Fookes, C., Denman, S., & Sridharan, S. Exploiting human social cognition for the detection of fake and fraudulent faces via memory networks. 2019, arXiv preprint arXiv:1911.07844.

[17] Venkatesh, S., Ramachandra, R., Raja, K., Spreeuwers, L., Veldhuis, R., & Busch, C. Detecting morphed face attacks using residual noise from deep multi-scale context aggregation network. In The IEEE Winter Conference on Applications of Computer Vision, 2020, pp. 280-289.

[18] Jeon, H., Bang, Y., & Woo, S. S. FDFtNet: Facing Off Fake Images using Fake Detection Fine-tuning Network, 2020, arXiv preprint arXiv:2001.01265.

[19] Zhang, W., Zhao, C., & Li, Y. A Novel Counterfeit Feature Extraction Technique for Exposing Face-Swap Images Based on Deep Learning and Error Level Analysis. Entropy, 2020, vol 22(2), no. 249. ##

[20] Dang, L. M., Min, K., Lee, S., Han, D., & Moon, H. Tampered and computer-generated face images identification based on deep learning. Applied Sciences, 2020, vol 10(2), no. 505.

[21] Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. Faceforensics: A large-scale video dataset for forgery detection in human faces, 2018, arXiv preprint arXiv:1803.09179.

[22] Jiang, L., Li, R., Wu, W., Qian, C., & Loy, C. C. DeeperForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2889-2898.

[23] de Lima, O., Franklin, S., Basu, S., Karwoski, B., & George, A. Deepfake Detection using Spatiotemporal Convolutional Networks, 2020, arXiv preprint arXiv:2006.14749.

[24] Li, Y., Yang, X., Sun, P., Qi, H., & Lyu, S. Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3207-3216.

[25] Chollet, F. Xception: deep learning with depthwise separable convolutions, in: 2017 IEEE conference on computer vision and pattern recognition CVPR, 2017.

[26] Kaiser, L., Gomez, A. N., & Chollet, F. Depthwise separable convolutions for neural machine translation, 2017, arXiv preprint arXiv:1706.03059.

[27] Rahimian, E., Zabihi, S., Atashzar, S. F., Asif, A., & Mohammadi, A. XceptionTime: Independent Time-Window Xceptiontime Architecture for Hand Gesture Classification. In ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 1304-1308.

[28] Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. Deepfakes and beyond: A survey of face manipulation and fake detection, 2020, arXiv preprint arXiv:2001.00179.

[29] Guarnera, L., Giudice, O., & Battiato, S. DeepFake Detection by Analyzing Convolutional Traces. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 666-667.

[30] Zhang, J., Salehizadeh, M., & Diller, E. Parallel pick and place using two independent untethered mobile magnetic microgrippers in IEEE International Conference on Robotics and Automation, 2018.

[31] Bau, D., Zhu, J. Y., Strobelt, H., Zhou, B., Tenenbaum, J. B., Freeman, W. T., & Torralba, A. Visualizing and understanding generative adversarial networks, 2019, arXiv preprint arXiv:1901.09887.

[32] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. Generative adversarial networks: An overview. IEEE Signal Processing Magazine, vol 35(1), 2018, pp 53-65.

[33] Kietzmann, J., Lee, L. W., McCarthy, I. P., & Kietzmann, T. C. Deepfakes: Trick or treat?. Business Horizons, 2020, vol 63(2), pp 135-146.
[34] Wang, J., Liu, A., & Xiao, J. Video-Based Pig Recognition with Feature-Integrated Transfer Learning. In Chinese Conference on Biometric Recognition, Springer, Cham, 2018, pp 620-631.
[35] Güera, D., & Delp, E. J. Deepfake video detection using recurrent neural networks. In 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE,2018, pp. 1-6.
[36] Afchar, D., Nozick, V., Yamagishi, J., & Echizen, I. Mesonet: a compact facial video forgery detection network. In IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, 2018, pp. 1-7.
[37] Sohrawardi, S. J., Chintha, A., Thai, B., Seng, S., Hickerson, A., Ptucha, R., & Wright, M. Poster: Towards robust open-world detection of deepfakes. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2613-2615.
[38] Albahar, M., & Almalki, J. Deepfakes: Threats and countermeasures systematic review. Journal of Theoretical and Applied Information Technology, vol 97(22), 2019, pp 3242-3250.
[39] Maksutov, A. A., Morozov, V. O., Lavrenov, A. A., & Smirnov, A. S. Methods of Deepfake Detection Based on Machine Learning. In 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), IEEE, 2020, pp. 408-411.
[40] Korshunov, P., & Marcel, S. Deepfakes: a new threat to face recognition assessment and detection, 2018, arXiv preprint arXiv:1812.08685.

Adeep Biswas is a graduate in B. Tech in Computer Science and Engineering from Vellore Institute of Technology. He looks forward to pursuing his Graduate Degree in information and communication technologies. His research interests include Image Processing, Information retrieval, Recommendation systems, Network Security, Web development.

Debayan Bhattacharya is a graduate in B. Tech in Computer Science and Engineering from Vellore Institute of Technology. He looks forward to pursuing his Graduate Degree in information and technology. His research interests include Image Processing, Network Security, Web development and computer vision.

Kakelli Anil Kumar is an Associate Professor of the School of Computer Science and Engineering at the Vellore Institute of Technology (VIT), Vellore, TN, India. He earned his Ph.D. in Computer Science and Engineering from Jawaharlal Nehru Technological University (JNTUH) Hyderabad in 2017, and graduated in 2009 and under-graduated in 2003 from the same university. He started his teaching career in 2004 and worked as an Assistant Professor, and Associate Professor and HOD in various reputed institutions of India. His current research includes wireless sensor networks, internet of things (IoT), cyber security and digital forensics, Malware analysis, block-chain and crypto-currency. He has published over 40 research articles in reputed peer reviewed international journals and conferences.

An Efficient Method for Handwritten Kannada Digit Recognition based on PCA and SVM Classifier

Ramesh G*
Department of Computer Science & Engineering, University Visvesvaraya College of Engineering, Bengaluru, India.
rameshmg6308@gmail.com
Prasanna G B
Department of Computer Science & Engineering, University Visvesvaraya College of Engineering, Bengaluru, India.
Prasannabhagwat98@gmail.com
Santosh V Bhat
Department of Computer Science & Engineering, University Visvesvaraya College of Engineering, Bengaluru, India.
Santoshbhat1998@gmail.com
Chandrashekar Naik
Department of Computer Science & Engineering, University Visvesvaraya College of Engineering, Bengaluru, India.
Canaik24@gmaol.com
Champa H.N
Department of Computer Science & Engineering, University Visvesvaraya College of Engineering, Bengaluru, India.
champahn@yahoo.co.in

Received: 16/Oct/2020 Revised: 01/May/2021 Accepted: 22/May/2021

Abstract
Handwritten digit recognition is one of the classical issues in the field of image grouping, a subfield of computer vision. The event of the handwritten digit is generous. With a wide opportunity, the issue of handwritten digit recognition by using computer vision and machine learning techniques has been a well-considered upon field. The field has gone through an exceptional turn of events, since the development of machine learning techniques. Utilizing the strategy for Support Vector Machine (SVM) and Principal Component Analysis (PCA), a robust and swift method to solve the problem of handwritten digit recognition, for the Kannada language is introduced. In this work, the Kannada-MNIST dataset is used for digit recognition to evaluate the performance of SVM and PCA. Efforts were made previously to recognize handwritten digits of different languages with this approach. However, due to the lack of a standard MNIST dataset for Kannada numerals, Kannada Handwritten digit recognition was left behind. With the introduction of the MNIST dataset for Kannada digits, we budge towards solving the problem statement and show how applying PCA for dimensionality reduction before using the SVM classifier increases the accuracy on the RBF kernel. 60,000 images are used for training and 10,000 images for testing the model and an accuracy of 99.02% on validation data and 95.44% on test data is achieved. Performance measures like Precision, Recall, and F1-score have been evaluated on the method used.

Keywords: Computer Vision; Dimensionality Reduction; Handwritten Digit Recognition; Kannada-MNIST Dataset; PCA; SVM.

1- Introduction

Machine learning and deep learning assumes a significant function in computer technology and artificial intelligence. With the utilization of machine learning, human exertion can be diminished in recognizing, learning, predicting and lot more regions. It is a fast-growing field of computer science that is making its way into all other domains. A significant space in this field is efficient and generic handwritten digit recognition. The handwritten digit recognition has many potential real applications such as marks digitization, banking utilities, reading postal code and tax form. The isolated handwriting recognition process can be broken down into three stages: pre- processing, feature extraction and classification. The important role in Feature extraction in getting high accuracy rates. However, along with this proper pre-processing of data also contributes to high accuracy. Many research activities are made in this regard for English numerals and impressive outputs are obtained. However, there is room for more improvement when it comes to Kannada numerals.

Computer vision is the field of computer engineering that centers around duplicating portions of the unpredictability of the human vision framework and empowering computer to recognize and deal with objects in pictures similarly that people do. On a specific level Computer Vision is about example pattern recognition. So one approach to prepare a computer how to understand

* Corresponding Author

visual information is to include the image, bunches of images on the off chance that conceivable that have been named, and afterwards subject those to different programming strategies, or calculations, that permit the computer to chase down examples in all the components that identify with those labels. Machine learning and computer vision are two fields that have gotten firmly identified with each other. Machine learning has improved computer vision about recognition and following. It offers successful strategies for acquisition, image processing, and object focus are utilized in computer vision. In turn, computer vision has broadened the scope of machine learning. PC vision has expanded the extent of AI. Machine learning is utilized in computer vision in the translation phase of digital image recognition.

A pictorial representation for the steps involved in image classification and recognition can be seen in fig. 1 Image Acquisition is the first step and is basically capturing an image and generally involves pre-processing of image, such as scaling, de-skewing etc. Following the acquisition step is image enhancement where the noise in the image are removed and also necessary enhancements such as increasing/decreasing contrast etc. is done in order to improve the image quality. Image Restoration forms the very next step and mainly targets improving appearance of an image by considering the image blur and hence reducing it using mathematical or probabilistic models. Compression as the name suggests involves reducing the size of the image using few well known techniques without losing much of the image quality. Segmentation is one of the important phases which involves separating the data into distinct groups. The member of the segment is different to other and they are similar from of the well-defined segment to other Representation and Description involves representing the data in various forms in an N-Dimensional space. Description or Feature Selection helps in extracting useful information. Recognition is the final process where labels are attached to the images based on the feature matching and classification

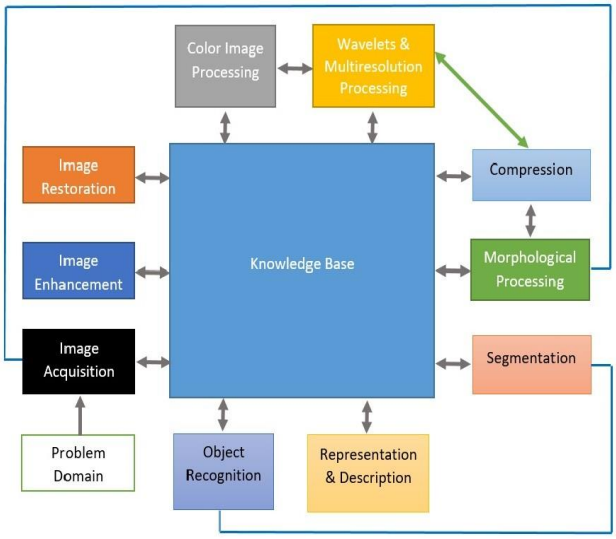


Fig. 1. Steps involved in image classification and recognition [23].

Support Vector Machines (SVM) is widely used for classification of numbers in handwritten digit recognition due to high accuracy. But when Principal Component Analysis (PCA) is used as a pre-processing step along with SVM much higher accuracy rate is seen. PCA reduces the number of features, and then they use some Principal Components (the eigenvectors of the covariance matrix) as the new features. This in turn removes the non-predictive features and gives much better results. This work presents recognizing the manually written Kannada digits (0 to 9) from the K-MNIST dataset, looking at SVM classifier and the cascade of SVM-PCA on RBF kernel. Various performance factors like the accuracy, precision, re- call, F1 score have been compared for the kernel and the two classifiers.

1-1- Motivation

Kannada numerals have a very long and rich history. The earliest inscription having all 9 Kannada numerals have been engraved in the Gudnapur Inscription which dates back to the time of Kadamba Ravivarma (485 A.D. to 519 A.D.). The symbols used to represent digits from 0 to 9 in the language are different from the well-known and modern Hindu-Arabic numerals. Even today, people of Karnataka use Kannada digits for day-to-day affairs. Kannada numerals also got itself a full- fledged Kannada-MNIST dataset in 2019. There have been numerous works around Kannada digits in ML before this. However, Kannada-MNIST data provides sufficient amount of data for training and testing. The state-of-the-art classifiers like SVM along with PCA have been used for recognition of handwritten digits in various languages. However, the method has not been implemented yet on a standard native

language dataset. This motivated us to use the SVM classifier along with PCA on the Kannada-MNIST dataset.

1-2- Contribution

We have used the Kannada-MNIST dataset to first train our model only using SVM. In the next phase we applied PCA before applying SVM. Based on the validation result we fine- tuned the parameters such as n-component value for PCA. After this we again trained our model using SVM. And this model was used to predict labels in the test data.

1-3- Organization

The work is unfurled over the pages in 7 segments. section I gives the introduction to the proposed work. An understanding about the past work is done on the difficult explanation is spread out in Section 2. section 3 depicts obviously the issue explanation we have drawn closer to settle. The proposed framework is described in Section 4 and Section 5 gives an itemized portrayal of the proposed framework. section 6 contains the test consequences of the proposed framework while Section 7 is an end to the work.

2- Literature Survey

The work on Kannada handwritten digit recognition is very limited and less research has been taken place in this field. There is scope for improvement in the techniques used till date for recognition of Kannada numerals. The recognition of isolated Kannada characters was first explored by Kunte [1] where wavelet features were extracted from the character contour and used as features. A character recognition accuracy of 56% was achieved using a Multi-layer feedforward neural network with one hidden layer. In Rajput et al., [2] binary images of numerals are created by scanning and a size of 40 x 40-pixel image is created after normalization. The line between the object pixels and the background (the crack) is computed and these are termed as Crack Codes. These codes are then represented in complex plane and features computed from 10 dimensional Fourier descriptors are used. The experiment is carried out using five-fold cross validation method with SVM as classifier. The work states that an accuracy of 99.76% and 95.22% has been obtained for printed and handwritten numerals, respectively. Segmenting and recognizing arbitrarily connected and superimposed handwritten numeral recognition in one-stroke finger gestures has been a problem and a solution to this has been proposed by Chiang et al., [3]. The method has two phases, key numeral spotting (KNS) phase and recognition by concatenation (RBC) phase. For recognizing key numerals in gesture, dynamic time warping (DTW) algorithm which is an endpoint detection method is used. The proposed solution achieves 94% precision.

Ramappa et al., [4] presented by the continuous exploration in Optical Character Recognition Systems with center around various techniques for division, segmentation, feature extraction and for classification. They are considering eight different features computed from zonal extraction, radon transform, fan beam projections, image fusion, discrete fourier transform, run length count, directional chain code and curvelet transform along with 10 different classifiers like Euclidean distance, Chebyshev distance, Manhattan distance, K-NN, K-medoids, Linear classifier, Cosine distance, Artificial Immune system, K-means and Classifier fusion are considered. They conclude by observing that a maximum recognition rate of 98.5% is achieved by zoning features with Artificial Immune System and that it outperforms all the other combinations.

A Zone based features is employed for recognition of handwritten and printed digits by Dhandra et al., [5]. 64 zones are formed form a digit image each zone is compared for pixel density. This procedure is sequentially repeated for entire zone. For classification and recognition 64 features are selected from the previous step. The value of zone row/column with empty foreground pixels in the feature vector is zero. By using KNN and SVM classifiers the work concludes saying that an accuracy of 97.32% and 98.30% respectively was achieved for mixed handwritten and printed Kannada digits. Karthik et al., [6] present a Histogram of Oriented Gradients (HOG) based method for the recognition of handwritten kannada numerals. HOG descriptors are considered one of the best descriptors for character recognition problem since they are invariant to geo-metric transformation. Multi-class Support Vector Machines (SVM) has been used for the classification. The proposed algorithm has achieved an average accuracy of 95% when experimented on 4,000 images of isolated handwritten Kannada numerals. Zone and Distance metric based feature extraction approach has been carried out by Rajashekararadhya [7] where character centroids are computed for each character image. The image is further divided into an equal zones and an average distance from the character centroid to each pixel present in the zone is computed. The classification and recognition is carried out using a feed forward back propagation neural network a recognition accuracy of 98% and 96% are obtained for Kannada and Telugu numerals respectively. The problem of recognizing printed and handwritten numerals seen in various documents has been addressed by Rajput et al. [8] where a Support Vector Machine based classification is implemented. Scanned numerals are converted to binary image and normalized to a size of 40 x 40. The boundaries are traced and chain codes are calculated. These codes are then represented in complex plane and features computed from 10 dimensional Fourier descriptors are used. These

are input to a multi-class SVM for recognition of class. An accuracy of 97.76% has been achieved for a data set size of 5000 mixed numerals image.

Hallur et al., [9] proposes a Holistic based approach for the recognition of offline handwritten numeral recognition. Due to varied shapes of kannada digits several features are considered carefully. Initial digits recognition varies after choosing such a set of features. Gradient features are extracted from gray scale image (value ranges from 0 to 255). Quadratic classifiers are used for classification purpose and an overall recognition accuracy of 95.98% is achieved on a dataset size of 1470 contributed by 147 people. U. Pal et al. [10] propose a quadratic classifier-based scheme for recognition of offline handwritten numerals of six popular Indian script. Devnagari, Bangla, Telugu, Oriya, Kannada and Tamil scripts are considered for the experiment. The bounding box of a numeral is segmented into blocks and the directional features are computed in each of the blocks as part of feature computation. Gaussian filter is then used to down sample these blocks and the features obtained are fed to a modified quadratic classifier for recognition.

Bovolo et al., [11] proposes a classifier that incorporated fuzzy logic for generalizing the properties of SVMs for the identification and modeling of many classes in mixed pixels. The results of this is a fuzzy-input fuzzy-output support vector machine classifier. This classifier processes fuzzy information given to as input to the classification algorithm in the learning phase of the classifier for modeling the subpixel information. It also provides a fuzzy modeling of the classification results, allowing a relation many-to-one between classes and pixels. Maloo et al., [12] propose a method for the recognition of Gujarati handwritten numerals which is basically an SVM based recognition scheme. Morphological operations are considered during the pre-processing stage of this method. Each isolated numeral is segmented into blocks in order to compute features. These blocks then create base for four sets of features. These sets of features are then used to obtain affine invariant moments as features which are fed as input to SVM classifier. The work mentions that the method has obtained a recognition rate of 91% approximately. Sajedi et al., [13] proposes a standardization of research works on OCR in Persian language. A database named PHOND, Persian Handwritten Optical Numbers & Digits is used for classification purpose. The proposed method uses K-Nearest Neighbour (KNN), and Support Vector Machines (SVM) with RBF, Linear and Polynomial kernels employed in SVM for measuring the effectiveness of the extracted features. The proposed method is said to achieve a higher recognition rate up to 99%.

A multiclass classifier-based approach is mentioned in [14] where PCA of HOG is utilized for exact and quick recognition of handwritten digits. HOG is known as a powerful element descriptor, while PCA brings about quick multiclass recognition. By joining PCA with HOG, the PCA-of-HOG based classier is reported to have achieved a recognition rate of 98.39% when applied on 10,000 test data from standard MNIST database. A very recent work on kannada MNIST can be inferred from [15] where the authors have used a skip architecture of CNN to skip some layers of the neural network and the output of previous layer is fed as the input to the next or some other layer. Experimental results show that they have achieved 97.53% accuracy on Kannada MNIST and 85.02% accuracy on Dig-MNIST Dataset. The Generative Adversarial Networks (GAN) is one of the technique that doesn't need earlier information on the possible variabilities that exist across guides to make novel artificial models [16]. On account of a training dataset with a low number of named models, which are portrayed in a high dimensional space, the classifier may sum up ineffectively. Hence, we target advancing information bases of images or signals for improving the classifier execution by planning a GAN for making artificial images. The proposed and evaluated generative adversarial networks as an information expansion procedure for the grouping of manually written digits of various contents (Latin, Bangla, Devanagari, and Oriya). The outcomes propose that such a methodology offers a significant increase in the accuracy.

The Deep Convolutional Self-organizing out Maps (DC-SOM) network contains a course of convolutional SOM layers prepared [17] successively to speak to various levels of features. The 2D SOM network is normally utilized for either information perception or feature extraction. this work utilizes high dimensional map size to make another deep network. A lot of experiments utilizing MNIST manually written digit information base and every one of its variations are directed to assess the robust representation of the proposed DCSOM network. Test results uncover that the presentation of DCSOM outperforms state-of-the-art techniques for noisy digits and accomplish a comparable performance with other deep learning architecture for other image varieties.

Cocotte et al., [19] presented by a novel dynamic learning strategy for the classification of manually written digits. The proposed technique depends on a k-closest neighbor graph got with an image deformation model, which considers nearby deformations. During the dynamic learning system, the user is first approached to labels the vertices with the highest number of neighbors. Label propagation function is performed to consequently label the models. The system is repeated until all the pictures are marked, they are evaluating the performance of the strategy on four database bases relating to various contents (Latin, Bangla, Devnagari, and Oriya) and they show that it is possible to label just 332 pictures in the MNIST

training database to acquire an accuracy of 98.54% on this same database base (60000 images).

Table I Summary of Literature Survey

Author Year	Title	Attribute
Kunte et al., 2000	170wavelet features based on-line recognition of handwritten	A character recognition accuracy of 56% was achieved using a Multi-layer feedforward neural network with one hidden layer.
Rajput et al., 2010	Printed and handwritten kannada numeral recognition using crack codes and fourier descriptors plate	Five-fold cross validation method with SVM as classifier is used to get an accuracy of 99.76% and 95.22% for printed and handwritten numerals, respectively.
Chiang et al., 2017	Recognizig arbitrarily connected and superimpoed handwritten numerals in intangible writing interfaces	Dynamic time warping (DTW) algorithm which is an endpoint detection method is used. The proposed solution achieves 94% precision.
Ramappa et al., 2013	A comparative study of different feature extraction and classification methods for recognition of hand- written kannada numerals	Maximum recognition rate of 98.5% is achieved by zoning features with Artificial Immune System and it outperforms all the other combinations
Dhandra et al., 2011	Zone based features for handwritten and printed mixed kannada digits recognition	KNN and SVM classifiers are used to get an accuracy of 97.32% and 98.30% for mixed handwritten and printed Kannada digits
Karthik et al., 2015	Handwritten kannada numerals recognition using histogram of oriented gradient descriptors and support vector machines	Histogram of Oriented Gradients(HOG) descriptors are used for character recognition and Multi-class Support Vector Machines (SVM) has been used for the classification. Accuracy of 95% achieved on 4,000 images of isolated handwritten Kannada numerals
Rajasheka raradhya et al., 2018	Neural network based handwritten numeral recognition of kannada and telugu scripts	The classification and recognition is carried out using a feed forward back propagation neural network a recognition accuracy of 98% and 96% are obtained for Kannada and Telugu numerals respectively
Rajput et al., 2010	Printed and handwritten mixed kannada numerals recognition using svm	Scanned numerals are converted to binary image and normalized to a size of 40 x 40. The boundaries are traced and chain codes are calculated. Features computed from 10 dimensional Fourier descriptors are used. These are input to a multi-class SVM for recognition of class. An accuracy of 97.76% has been achieved for a data set size of 5000 mixed numerals image
Hallur et al., 2014	Offline kannada handwritten numeral recognition: Holistic approach	Gradient features are extracted from gray scale image (value ranges from 0 to 255). Quadratic classifiers are used for classification purpose and an overall recognition accuracy of 95.98% is achieved on a dataset size of 1470 contributed by 147 people
U.Pal et al., 2007	Handwritten numeral recognition of six popular indian scripts	The bounding box of a numeral is segmented into blocks and the directional features are computed in each of the blocks as part of feature computation. Gaussian filter is then used to down sample these blocks and

		the features obtained are fed to a modified quadratic classifier for recognition
Bovolo et al., 2010	A novel technique for subpixel image classification based on support vector machine	Fuzzy logic classifier used to processes fuzzy information given to as input to the classification algorithm in the learning phase of the classifier for modeling the subpixel information and provides a fuzzy modeling of the classification results.
Maloo et al., 2011	Support vector machine based gujarati numeral recognition	Morphological operations are considered during the pre-processing stage of this method. Later SVM classifier is used to get a recognition rate of 91% approximately
Sajedi et al., 2016	Handwriting recognition of digits, signs, and numerical strings in persian	PHOND, Persian Handwritten Optical Numbers & Digits is used for classification purpose. The proposed method uses K-Nearest Neighbour (KNN), and Support Vector Machines (SVM) with RBF, Linear and Polynomial to achieve a recognition rate up to 99%.
Cocotte et al., 2016	Active graph based semi-supervised learning using image matching: application to handwritten digit recognition	K-closest neighbor graph got with an image deformation model ND Label propagation function is performed to consequently label the models to acquire an accuracy of 98.54% on MNIST training database

3- Problem Statement and Objectives

This work mainly focuses on solving the recognition of handwritten Kannada digits, by using the state-of-the-art classifier, which gives very high accuracy along with reliability. This effort is focused on using SVM with PCA on the Kannada-MNIST dataset to recognize the digits accurately. The goal of this work is to solve Recognition of Handwritten Kannada Digits, which is a very challenging topic for the researchers in recent years. This lets the computer to understand Kannada digits that is written manually by the user using PCA and SVM. This work likewise analyses the limitations in different methods that are being used as the solution for the issue.

4- System Architecture

The architecture followed in the proposed method is neatly laid out in Fig. 2 the dataset comprises of images of hand-written numerals in Kannada with 60,000 image in preparing set and 10,000 image in the test set with a size of 28x28 for each image. Pre-processing of the data is carried out in the subsequent step where data is de-skewed first to correct the alignment of the data. After this step, we scale the data for normalization. We also reduce the noise in the data by filling out the missing data or ignore the less important data. Following the pre-processing stage, feature extraction is carried out where PCA is used to map the high dimensionality space of input features to lower dimensions by throwing away some columns. It selects some columns of features, which has low standard deviation and deletes it from the feature matrix. The data points with the highest correlation will be retained after applying this method and has good impact on further training process.

One of the major parts of the system architecture include classification using SVM, which classifies the digits based on the features of the array of grayscale images. The features considered here are the Mean and standard deviation of each digit. Multiclass SVM classifiers are trained using these features which effectively separates different classes of digits by finding the hyperplanes with maximum margins. The performance of the model can be scrutinized by visualizing the output using pictorial elements such as charts, graphs, and maps. Data visualization tools provide assistance in understanding trends, outliers, deviation and patterns in data. By this, we can compare the expected and predicted output of the models. We can also diminish the performance of different algorithms by representing data in a pictorial form. Output analysis is the essential step to evaluate the performance of the model. A good model should give satisfactory results for the input data. We can use Confusion Matrix, F1 Score, classification accuracy, mean squared errors etc. to evaluate the performance of the model.

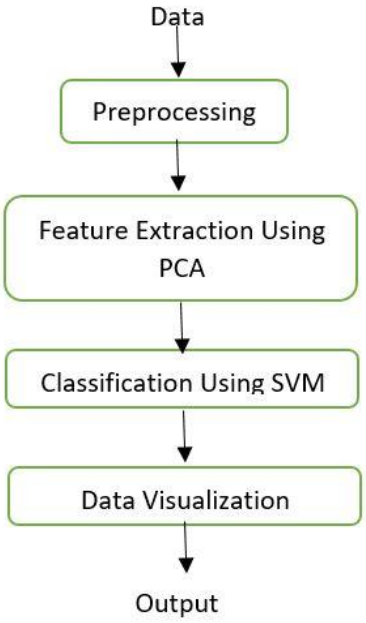


Fig. 2. System Architecture.

5- Proposed PCA-SVM Classifier

5-1- Kannada Numerals

Kannada is the official and the most spoken language in Karnataka- a state of India. It is one of the oldest Dravidian languages of India just like Tamil. The earliest inscription having all 9 Kannada numerals have been engraved in the Gud napur Inscription which dates back to the time of Kadamba Ravivarma (485 A.D. to 519 A.D.). The symbols used to represent 09 as shown in Fig. 3 are distinct from the modern Hindu-Arabic numerals.

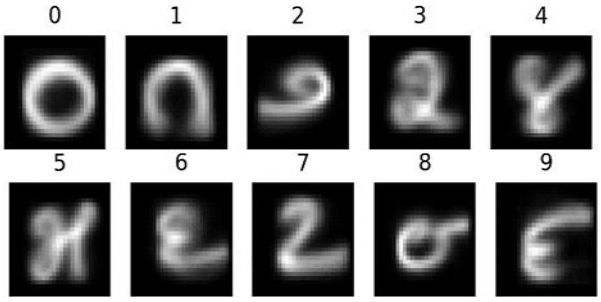


Fig. 3. Handwritten digits in the Kannada MNIST dataset.

5-2-Principal Component Analysis

PCA is a strategy, which utilizes orthogonal transformation change to convert connected information into uncorrelated information known as principal components. It produces a simple representation of a data set by eliminating the columns containing less significant features and thus reduces the dimensionality of the data as shown in Fig. 4. An arrangement of linear combination of the factors that have greatest change and are commonly uncorrelated [21]. Apart from finding, the features with major significance it also used for data visualization.

$$Z_1 = \Phi_{11}X_1 + \Phi_{21}X_2 + \ldots + \Phi_{p1}X_p \quad (1)$$

In Equation 1, a set of features X_1, X_2, ..., X_p is converted into a single principal component Z_1. Here Z_1 is the largest variance of normalized linear combination of the features X_1, X_2, ..., X_p . And Φ_{11}, Φ_{21}, ..., Φ_{p1} are the loadings of the first principal component. Eliminating the correlated data minimizes the loss of information.

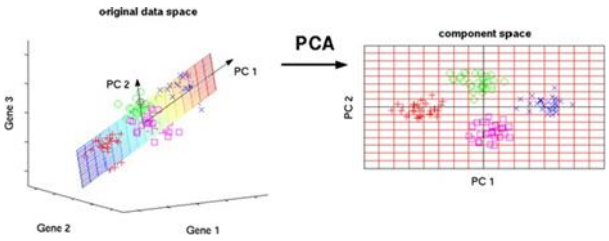


Fig. 4. Dimensionality reduction from 3D to 2D by finding the common hyperplane.

5-3-Support Vector Machine

The SVM is used as the classification algorithm and is a supervised learning model. It finds a hyperplane in an N-dimensional space (N- Number of features) and such hyper- plane contains all the unique features of the data in Fig. 5. The objective is to find the best-fit hyperplane of all the possible hyperplanes. The plane is called best fit when it has the maximum distance between data points of different classes in Fig. 6. Support vectors are the points, which are close to the hyperplane and impact on the locality and inclination of the hyperplane. We can maximize the distance of the data points, which lie in different classes using these support vectors in Fig.7.

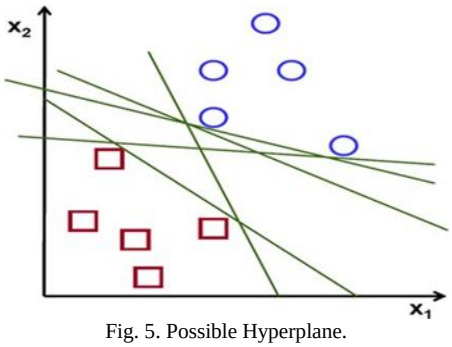


Fig. 5. Possible Hyperplane.

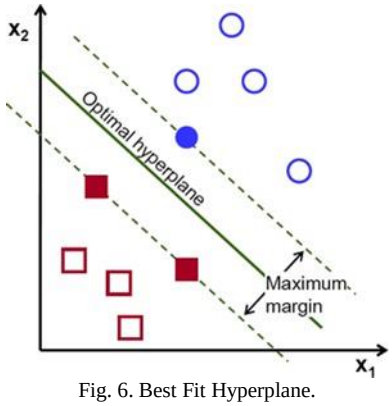


Fig. 6. Best Fit Hyperplane.

Hyperplane that separates the features of different classes is defined by the equation:

$$Y_i\,(W * X_i + b) \geq 1 \text{for} 1 \leq i \leq n \qquad (2)$$

where X_i are the data points shown as single dimensional matrix in a d-dimensional space, 'n' is the count of data points, Y_i -1,1 are classes of respective occurrences. 'w' and 'b' are the parameters of hyperplane. Hyperplane has to lie at a large distance as much as possible from data points of both the classes. This distance can be increased using $\frac{2}{\|w\|}$

Real world information contains noise and outliers that can be eliminated using the equation:

$$Y_i\,(W * X_i + b) \geq 1 - \varepsilon_i,\ \varepsilon_i \geq 0,\ 1 \leq i \leq n \qquad (3)$$

Equation 3 is the improvised equation of 2. It has the slack variable 'ε' to separate the noise and outliers in the data. We can represent the model using different kernels like sigmoid, polynomial and RBF kernels to make it linearly separable.

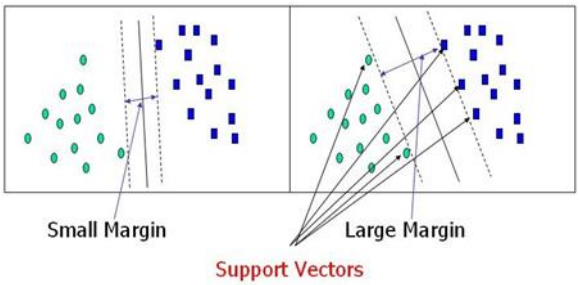


Fig. 7. Illustration of support vectors.

5-4- RBF Kernel

In the proposed method, we have used RBF kernel function as the pattern of distribution is found to be radial and it leads to impressive classification accuracy for the digits. RBF kernel function is defined as:

$$K\,(X_i, X_j) = \exp\,(-\gamma\|X_i - Xj\|)\,2 \qquad (4)$$

where X_i X_j is the Euclidean distance between X_i and X_j. $J_\gamma J$ is the parameter of kernel function. This parameter has effect on the standard or caliber of classifier, so adjusting the value of this parameter is a significant part.

5-5-Flow Diagram

Our system uses the following algorithm to solve the problem statement. The methods followed by the algorithm has been neatly laid out in Fig. 8 below.

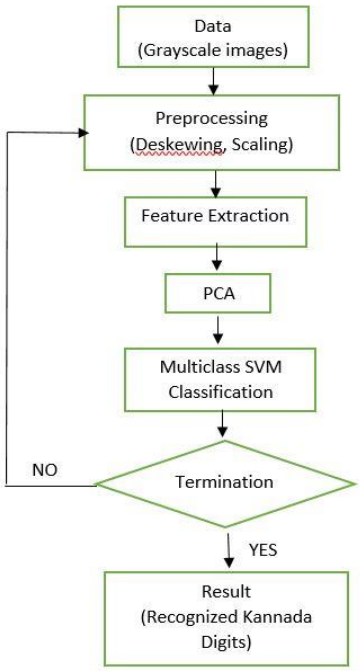


Fig. 8. Flow chart of SVM-PCA merge algorithm.

5-6- Implementation

In order to tackle the challenge in handwritten digit recognition we underwent following steps to increase the accuracy

- De-skewing: Different people write at different angles on paper or any surface. This leads to inclination of numbers written. Thus, the numbers appear to be skewed. Human eye can find similarities even though the images are variations of one another. But computer distinguishes such variations as different images. Hence de-skewing becomes necessary. De-skewing is the method of making a handwritten crooked image which has been scanned staright by changing the inclination of the image. To be specific, de-skewing is defined as an affine transformation. It is assumed that when the image was made initially (that is the skewed image), the image was some affine skew transformation on the de-skewed image.

$$\text{Image}^{j} = \text{A(Image)} + \text{b} \quad (5)$$

Where Image' is the skew corrected version of original image Image. b gives the histogram of the gradient orientation of the original image. To find out to what extent the image has to be offset, the center of mass of the image has to be found. After this the covariance matrix of the image pixel intensities can be figured out (for this approximate the skew of the angle can be used). The following formula shows this,

$$\begin{pmatrix} 1 & 0 \\ a & 1 \end{pmatrix}$$

Where

$$a = \frac{Cov(X,Y)}{V\,ar(X)} \quad (6)$$

- Data Normalization: If the values for different features vary largely from one another then the understanding capacity of the model fails significantly. It will take long time to complete the learning process and affect the model accuracy adversely. So, to achieve fast convergence we need to normalize the data and squash them into [0,1] in the preprocessing step.
- N Component Analysis: It is one of the necessary parameter to apply PCA on the feature space. It is the value of variance we need to keep before reducing the dimension of feature space in order to get the peak value of accuracy. The accuracy of the model has been analyzed along the Y-axis with the variance along the X- axis using the line graph shown in Fig. 9. It is found that the accuracy obtained is maximum at 0.7. Then the value of N component to our PCA algorithm has been fetched. The model kept only those features which have the variance value ¿=0.7. It removed those data points of less variance (less information) and made it easy for relational mapping of features by reducing the dimensions. PCA reduced the dimension of the feature matrix from 784 to 57 by eliminating the features having less variance than 0.7 which wouldn't contribute to the performance of the model.

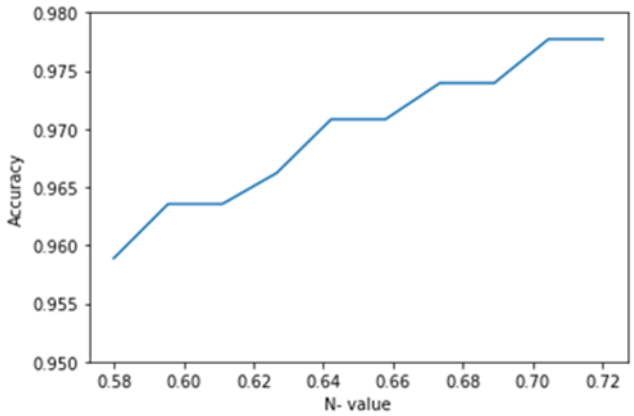


Fig. 9. Accuracy Vs Variance (N-value).

1)SVC parameters: Support Vector Classifier attempts to find the best fit hyperplane to distinguish the different classes by increasing the distance between sample points and the hyperplane. In order to demo the best fitting hyperplane fine tuning of the parameters is very important. The parameters which are tuned to get the robust model is illustrated below.
1) Regularization Parameter (C): It is utilized to compromise correct classification of preparing models against maximization of the decision functions margin. Small margin will be accepted for greater values of C if there is better classification of all training points correctly by the decision function. Larger margin will be encouraged by a lower C. Thus we get a simpler decision function, at the expense of training accuracy.
2) Kernel: To classify the nonlinear data we should apply kernel functions. These functions convert linearly independent data into linearly dependent ones. On each data instance, the kernel functions are applied which maps the initial non-linear observations into a higher-dimensional space in which they become independent. Our model reacts positively to the RBF kernel function and the value is set to rbf
3) Gamma: It is used to define the reach of a single training example's influene. Here far means low values and close means high values. It is clear that the gamma parameters are the inverse of the radius of influence of samples selected as support vectors by the model. Here, we set the gamma value as auto

which uses 1/n-features. The gamma parameter largely influences the behavior of the model. A large value of gamma will lead to the radius of the area of influence of the support vectors to include only the support vector itself. And overfitting cannot be prevented with any amount of regularization with C. On the other side, the model becomes very much constrained with a smaller value of gamma. Thus model cannot capture the shape or complexity of the data. Any selected support vector's region of influence would contain the entire training set. The model created will act in same way as a linear model which has a set of hyperplanes that differentiates high density centers of any pair of two classes.

4) Degree: It is the degree of the polynomial function. The default value of this is 3. It will be ignored by other kernels.

6- Experimental Results

The KMNIST dataset for training as well as testing the SVM+PCA model. The data is preprocessed using methods like normalization and de-skewing to increase the accuracy of the model. Then the n-component value for PCA is deter- mined. The hyper parameters of SVM are also tuned to get the best accuracy before using the model on test data.

6-1-Dataset Analysis

The proposed methodology has been applied to a dataset [18] consisting of 70000 samples of unique handwritten Kannada numbers. For experimentation, 60000 samples are used for model training and the remaining 10000 samples are used for model testing. The size of image considered is 28 * 28 pixel. In the preprocessing step these images undergone de-skewing and min-max normalization methods to remove the noisy and irregular data. The results of the evaluation are carried out in two-fold, first without applying the PCA and then with PCA. Of the various kernel functions available in SVM such as linear, RBF and polynomial, due to the nonlinearity of the data, RBF kernel appeared to be the best fit and with higher accuracies than the other functions.

6-2-SVM + PCA Model

The model is first trained on 42,000 digits of the dataset. Then, it is validated using 18,000 digits. After this, PCA is applied on the data to reduce the dimension of data. For this purpose, we need to choose the number of components for reducing dimension. First, a random value of 0.6 is chosen as the n-component value. This reduces the dimension from 784 to 57. Then, SVM classifier is used on the data. The accuracy obtained was 98.8%. To improve accuracy, n-component analysis is done to get the optimal value of n-components. The accuracy change with change of N-component value can be seen in Fig. 9. The accuracy curve flattens after reaching n-components value of 0.7. This value is chosen for applying in PCA, as this is the optimal value. For n-component value of 0.7, we get accuracy of 99.02%. Based on validation results, fine tuning of the model is done and the model is then tested again on 10,000 digits. as mentioned in Table II. The difference in accuracies can be clearly seen between only SVM and SVM+PCA. Reducing the dimensionality of the data helps in increasing the accuracy in both validation data as well as test data.

Table II Analyzing Accuracy for SVM and SVM + PCA

ACCURACY	SVM	SVM + PCA
Validation data(18000 digits)	96.33%	99.02 %
Test data (1000 digits)		95.44%

6-3-Classification Report

The model is first trained only using SVM classifier without applying PCA and the accuracy obtained is 96.35%. After the application of PCA on the data, accuracy of the model increased to 99.02%. The measure of performances accuracy, precision, recall and f1 score which has been calculated for the model for the given data set. The models performance for the data set can be studied using these measures and proper tuning of parameters can be done. The accuracy obtained for individual digits is shown in Table III.

Accuracy: Accuracy is closeness of the measurements to a specific value. In classification problems, it is the proportion of number of correct predictions to the total number of input tests. Based on Confusion matrix given in Fig. 14. and the above definition we can calculate the accuracy of individual digits. This is shown in Table III and Fig. 10.

$$\text{Accuracy} = \frac{TP+TN}{FP+FN+TP+TN} \quad (7)$$

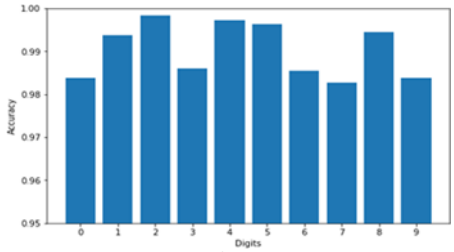


Fig. 10. Accuracy for Individual Digits.

Table III Individual Digit Accuracy

Digit	Accuracy
0	98.38%
1	99.38%
2	99.83%
3	98.61%
4	99.72%
5	99.64%
6	98.55%
7	98.27%
8	99.44%
9	98.38%

Precision: It is the closeness of measurements to each other. In classification problems, it is the fraction of appropriate occurrences among the extracted occurrences. The precision value for individual digits calculated for the SVM + PCA model and visualized in Fig. 11.

$$\text{Precision} = \frac{TP}{FP+TP} \quad (8)$$

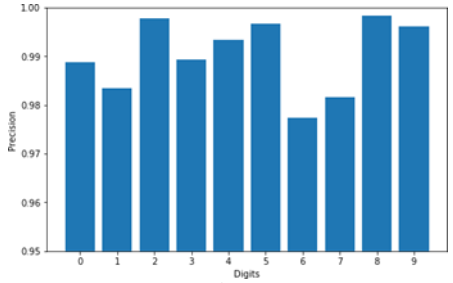


Fig. 11. Precision for Individual Digits.

Recall: It is the fraction of the total amount of appropriate occurrences that were actually extracted. It is also known as sensitivity. The recall value for individual digits calculated for the SVM + PCA model and visualized in Fig. 12.

$$\text{Recall} = \frac{TP}{FN+TP} \quad (9)$$

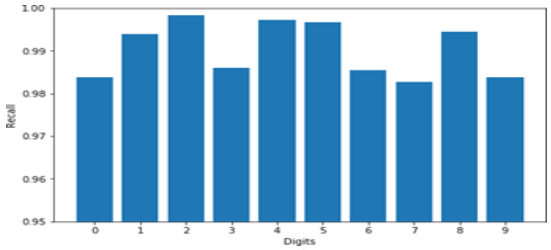


Fig. 12. Recall for Individual Digits.

F1 Score: It shows the balance between recall and precision. It is also defined as the harmonic mean of the recall and precision. The F1 Score value for individual digits calculated for the SVM + PCA model and visualized in Fig. 13.

$$\text{F1 Score} = 2^* \left(\frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}\right) \quad (10)$$

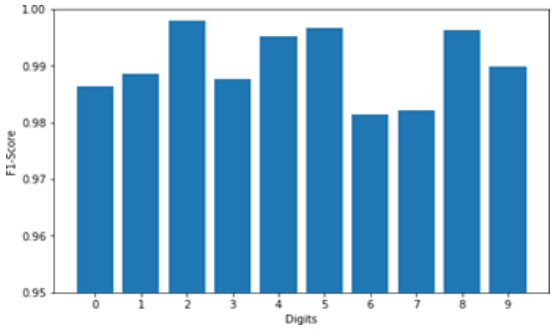


Fig. 13. F1 Score for Individual Digits.

6-4 Confusion Matrix

Confusion matrix is used in classification problems having two or more classes in the output for measurement of performance. It is also known as Error matrix. It is a particular table design which shows the representation of the presentation of an algorithm, normally a managed learning algorithm. The instances in a predicted class are represented by the rows of the matrix while instances in an actual class are represented by the columns. The confusion matrix shown in Fig. 14 clearly shows that a maximum misclassification error is observed between the digits 0 and 1. The misclassification of handwritten 0 is shown in Fig.15 The second most misclassification is seen between the digits 6 and 7. The reasons for misclassification may be due to similarity of shape found between some of the digits, but it also depends on different writing styles followed by individuals, which make the samples of a particular class closer to other class. Training accuracies for SVM along with PCA have peaked at 99.02% and the test accuracy of 95.44% as mentioned in Table IV

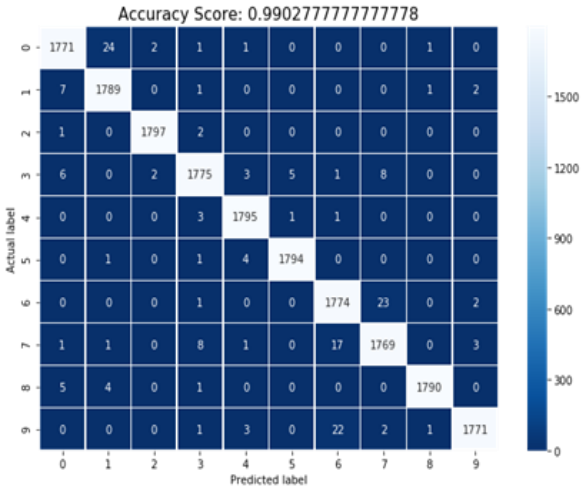


Fig. 14. Confusion Matrix for SVM+PCA model.

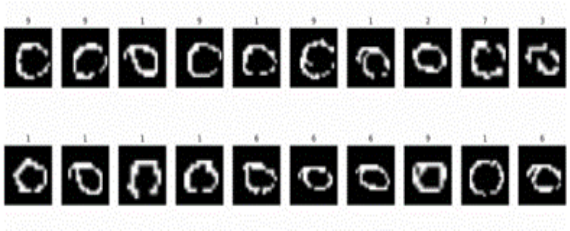


Fig. 15. Misclassification error between ’0’ and ’1’.

Table IV Evaluation Results.

Author	Method	Accuracy
UÇAR et al.,[22] 2019	Capsule Network	81.63%
Hallur et al., [20], 2020	DCNN	96.00%
Proposed Method	**PCA+ SVM**	**Validation - 99.02% Test data - 95.44%**

7- Conclusion

In this work, we present the combined model of PCA and SVM for classification of Kannada handwritten numerals. We also present how PCA can boost the performance of SVM in terms of better classification accuracies and at improved training speed. The proposed method outperforms the trivial method of using only the Support Vector Machines without Component Analysis. Training accuracies for SVM along with PCA have peaked at 99.02% and the test accuracy of 95.44%. With these great outcomes, the real-life requirements that motivated us to take a shot at the problem statement are reachable with greater reliability. The introduced strategy’s performance measures such as Precision, Recall and F1-scores have also been summarized and studied for carefully tuning the hyper parameter. A portion of the improvements required lie in handling misclassification of digits whose physical appearance are similar. These improvements can make the framework much more strong, enabled its favorable circumstances with to deal with raw images, saving pre-processing time, to save positional data of substances, which helps in the recognition of kannada handwriting digit.

References

[1] R. R. Kunte and R. Samuel, "170wavelet features based on-line recognition of handwritten," Journal of the Visualization Society of Japan, vol. 20, no. 1, pp. 417–420, 2000.

[2] G. Rajput, H. Rajeswari, and C. Sidramappa, "Printed and handwritten kannada numeral recognition using crack codes and fourier descriptors plate," International Journal of Computer Application (IJCA) on Recent Trends in Image Processing and Pattern Recognition (RTIPPR)}, pp. 53-58, 2010.

[3] C. Chiang, R.-H. Wang, and B.-R. Chen, "Recognizing arbitrarily connected and superimposed handwritten numerals in intangible writing interfaces," Pattern Recognition, {Elsevier} vol. 61, pp. 15--28, 2017

[4] M. H. Ramappa and S. Krishnamurthy, "A comparative study of different feature extraction and classification methods for recognition of hand- written kannada numerals," International Journal of Database Theory & Application, vol. 6, no. 4, pp. 71–90, 2013.

[5] B.V.Dhandra, G. Mukarambi, and M. Hangarge, "Zone based features for handwritten and printed mixed kannada digits recognition," IJCA Proceedings on International Conference on VLSI, Communications and Instrumentation (ICVCI), no. 7, pp. 5–8, 2011

[6] S. Karthik and K. Murthy, "Handwritten kannada numerals recognition using histogram of oriented gradient descriptors and support vector machines," Advances in Intelligent Systems and Computing, vol.2, pp. 51–57, 2015.

[7] S. V. Rajashekararadhya and P. Vanaja Ranjan, "Neural network based handwritten numeral recognition of kannada and telugu scripts," in TENCON 2008 - 2008 IEEE Region 10 Conference, pp. 1–5, 2008.

[8] G. Rajput, Horakeri, Rajeswari, and C. Sidramappa, "Printed and handwritten mixed kannada numerals recognition using svm," International Journal on Computer Science and Engineering, vol. 2, pp. 1622- 1626, 2010.

[9] V. Hallur and R. Hegadi, "Offline kannada handwritten numeral recognition: Holistic approach," Proceeding of Second International Conference on Emerging Research in Computing, Information, Communication and Applications, vol. 3, pp. 632-637, 2014.

[10] U. Pal, N. Sharma, T. Wakabayashi, and F. Kimura, "Handwritten numeral recognition of six popular indian scripts," in Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), vol. 2, pp. 749–753, 2007.

[11] F. Bovolo, L. Bruzzone, and L. Carlin, "A novel technique for subpixel image classification based on support vector machine," IEEE Transactions on Image Processing, vol. 19, no. 11, pp. 2983–2999, 2010.

[12] M. Maloo and K. Kale, "Support vector machine based gujarati numeral recognition," International Journal of Computer Science Engineering (IJCSE), ISSN 0975-3397, vol. 3, pp. 2595–2600, 07 2011.

[13] H. Sajedi, "Handwriting recognition of digits, signs, and numerical strings in persian," Computers Electrical Engineering, vol. 49, pp. 52– 65, 01 2016.

[14] W. Lu, "Handwritten digits' recognition using pca of histogram of oriented gradient," in 2017 IEEE Pacific Rim Conference on Communi- cations, Computers and Signal Processing (PACRIM), pp. 1–5, 2017.

[15] E. S. GATI, B. D. NIMO, and E. K. ASIAMAH, "Kannada-mnist classification using skip cnn," in 2019 16th International Computer Conference on Wavelet Active Media Technology and Information Pro- cessing, pp. 245–248, 2019.

[16] G. Jha and H. Cecotti, "Data augmentation for handwritten digit recognition using generative adversarial networks," Multimedia Tools and Applications, pp. 1–14, 2020.

[17] S. Aly and S. Almotairi, "Deep convolutional self-organizing map network for robust handwritten digit recognition," IEEE Access, vol. 8, pp. 107035–107045, 2020.

[18] V. U. Prabhu, "Kannada-mnist: A new handwritten digit's dataset for the kannada language," arXiv preprint arXiv:1908.01242, 2019.

[19] H. Cocotte, "Active graph based semi-supervised learning using image matching: application to handwritten digit recognition", Pattern Recognition Letters, vol. 73, pp. 76--82, 2016.

[20] Hallur, Vishweshwrayya C., and R. S. Hegadi. "Handwritten Kannada numerals recognition using deep learning convolution neural network (DCNN) classifier." CSI Transactions on ICT, vol. 8, pp. 295-309, 2020.

[21] Aly, Saleh, and Ahmed Mohamed. "Unknown-length handwritten numeral string recognition using cascade of pca-svmnet classifiers." IEEE Access vol. 7, pp. 52024-52034. 2019.

[22] UÇAR, Emine, and Murat UÇAR. "Applying Capsule Network on Kannada-MNIST Handwritten Digit Dataset." Natural and Engineering Sciences (2019).

[23] Gonzalez, Rafael C., and Richard E. Woods. "Digital image processing." (2002).

Ramesh G is currently a Research Scholar in the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering (UVCE), Bangalore University, Bangalore. He has completed his B. E and M.Tech from Vishveswaraya Technological University (VTU), Karnataka. All the degrees are in Computer Science and Engineering (CS&E) discipline. He has published papers in International Reputed Journals and International Conferences. He has attended various FDP programs. His current research lies in the areas of Image Processing, Machine learning, deep learning. He is a student member of the IEEE.

Prasanna G B is currently working as a Software Engineer in a reputed Automotive Company. He has completed his B.E in Information Science and Engineering from Department of Computer Science and Engineering, University Visvesvaraya College of Engineering (UVCE), Bangalore University, Bangalore. His current interest lies in the field of Data Processing, Image Processing and Machine Learning.

Santosh V Bhat is currently working as a Software Engineer in a reputed multinational software company. He has completed his B.E in Information Science and Engineering from Department of Computer Science and Engineering,

University Visvesvaraya College of Engineering (UVCE), Bangalore University, Bangalore. His current interests lie in field of Computer Vision and Machine Learning.

Chandrashekar Naik is currently working as a Software Engineer in a reputed Multinational Company. He has completed his B.E in Information Science and Engineering from Department of Computer Science and Engineering, University Visvesvaraya College of Engineering (UVCE), Bangalore University, Bangalore. His current interests lie in field of Data Processing, Data Analytics and Data Visualization.

Champa H N has completed Bachelor of Engineering, Masters of Technology and Doctoral Degree in Computer Science and Engineering. She has 30 years of teaching experience. Currently she is Professor in the Dept. of CSE, University Visvesvaraya College of Engineering, Bangalore. She has over 20 research papers to her credit. She is currently guiding 04 Ph.D Students. Her research interests include Image processing, Artificial Intelligence, Machine learning and Database systems.

Overcoming the Link Prediction Limitation in Sparse Networks using Community Detection

Mohammad Pouya Salvati
Faculty of Electrical and Computer Engineering, Urmia University, Iran
pouya.salvati@yahoo.com
Jamshid Bagherzadeh Mohasefi
Faculty of Electrical and Computer Engineering, Urmia University, Iran
j.bagherzadeh@urmia.ac.ir
Sadegh Sulaimany*
Faculty of Computer Engineering, University of Kurdistan, Iran
S.Sulaimany@uok.ac.ir

Received: 01/Jul/2020 Revised: 15/May/2021 Accepted: 05/Jun/2021

Abstract

Link prediction seeks to detect missing links and the ones that may be established in the future given the network structure or node features. Numerous methods have been presented for improving the basic unsupervised neighbourhood-based methods of link prediction. A major issue confronted by all these methods, is that many of the available networks are sparse. This results in high volume of computation, longer processing times, more memory requirements, and more poor results. This research has presented a new, distinct method for link prediction based on community detection in large-scale sparse networks. Here, the communities over the network are first identified, and the link prediction operations are then performed within each obtained community using neighbourhood-based methods. Next, a new method for link prediction has been carried out between the clusters with a specified manner for maximal utilization of the network capacity. Utilized community detection algorithms are Best partition, Link community, Info map and Girvan-Newman, and the datasets used in experiments are Email, HEP, REL, Wikivote, Word and PPI. For evaluation of the proposed method, three measures have been used: precision, computation time and AUC. The results obtained over different datasets demonstrate that extra calculations have been prevented, and precision has been increased. In this method, runtime has also been reduced considerably. Moreover, in many cases Best partition community detection method has good results compared to other community detection algorithms.

Keywords: link Prediction; Sparse Network; Clustering; Time efficient.

1- Introduction

As networks grow, link prediction greatly helps our trade and communication in many large-scale online commercial and social networks. Besides attempting to find missing links, link prediction also seeks to predict new links that may establish in the future. It is precious in a complex network to predict this category of links. On the other hand, high costs are required in laboratories to detect new or missing relations or links for some networks, such as protein-protein interaction relations. Clearly, prediction of correct links in such networks can play a pivotal role in treatment of many diseases such as AIDS and cancer. However, these networks are almost imperfect, low-density, and sparse. Also, practical experimentation to correct them, especially for biological networks, causes high costs to incur.

Link prediction can predict and subsequently improve the structure of the networks[1]. Many prediction methods have been presented, attempting to improve prediction results. Many of the available networks are sparse, which causes high extra calculation. This means that number of zero entries that needs to be scored in the associate adjacency matrix are far more than the existing ones, in computation and loss of time and resources. To the best of our knowledge, this issue has been mentioned implicitly or explicitly in some researches, but the appropriate solution has not been found [2][3].

This paper seeks to present a new, more accurate approach for link prediction in sparse networks. Regarding the main pitfalls of sparse networks for link prediction, we reduce the time consuming computations in addition to improve the precision as well. Eliminating the extra computations will be possible by removing the unnecessary predictions that do not have significant effect on the main results. We will achieve this aim by clustering the nodes and localizing the computations on the compacted parts of the network.

* Corresponding Author

After that, we consider some effective strategies to implement between clusters link predictions. The proposed method can be used for both, predicting new links or finding missing links correctly, especially in sparse networks, somehow as the networks are sparse, the result becomes better. We may use the terms clustering or community detection interchangeably throughout this paper.

The rest of the paper is organized as follows. In section 2, the related works are illustrated. After that in section 3, the proposed method and the evaluation are explained. In section 4, results and discussion are reported, and finally, in section 5, future work and conclusion will be discussed.

2- Related Works

We review the related researches about link prediction using community detection and link prediction for sparse networks, in this section, after a short overview of the primary related concepts. Link prediction methods have mainly two major categories: unsupervised and supervised. There are several unsupervised methods where the score $score(x,y)$ is considered for each pair of nonexistent links. Clearly, the higher the score, the greater the probability of establishment of a link is. The methods are divided into two broad categories: neighborhood-based and path-based methods [4]–[7]. It is worth mentioning that we use the neighborhood-based methods, we will refer to them as basic methods, including CN, JC, AA, RA, and PA. The full name and ranking formula for the methods are shown in Table 1. It is popular for new ideas to be tested with basic methods.

Table 1. Different scoring functions for neighbourhood-based unsupervised link prediction. $\boldsymbol{\Gamma(x)}$ is the set of neighbours of node x, and $|\boldsymbol{\Gamma(x)}|$ is the number of neighbors of node *x*

Neighborhood-based	**Common Neighbors (CN)**	$score(x,y) = \lvert\Gamma(x) \cap \Gamma(y)\rvert$
	Jaccard Coefficient (JC)	$score(x,y) = \frac{\lvert\Gamma(x) \cap \Gamma(y)\rvert}{\lvert\Gamma(x) \cup \Gamma(y)\rvert}$
	Adamic-Adar (AA)	$score(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log(\lvert\Gamma(z)\rvert)}$
	Resource Allocation (RA)	$score(x,y) = \sum_{c \in x \cap y} \frac{1}{\lvert\Gamma(c)\rvert}$
	Preferential Attachment (PA)	$score(x,y) = \lvert\Gamma(x)\rvert \times \lvert\Gamma(y)\rvert$

2-1- A Review of link Prediction using Community Detection

The community structure can be observed in many of the available networks. The notion of community or cluster depends largely on the type of the network or the information it contains[8]. In a metabolic network or an inter-protein interaction (protein-protein) network, for instance, a community can be a series of adjacent proteins that perform a biological operation inside a cell [9]. In a commercial network, a cluster can be a series of customers with similar purchasing backgrounds or similar tastes [10]. On the web, a cluster can be a series of pages about a certain issue [11].

In [12], the number of links between every two nodes $\boldsymbol{u}$ and $\boldsymbol{v}$ is calculated, and is normalized using the number of possible links between them. This value is referred to as the probability that there is a link between the two nodes $\boldsymbol{u}$ and $\boldsymbol{v}$. The drawback of this method is that prediction is made among all the nodes, and the network undergoing prediction is not necessarily sparse. Another cluster related link prediction type involves the stochastic block model [13], [14]. In this type of model, all the nodes are summed for categorization. The probability that two nodes are connected is obtained based on their membership in the relevant clusters. The most significant disadvantage of these methods is that they are impractical for large-scale networks due to the high time complexity of obtaining the optimal clustering.

In a similar study conducted in 2012 [15], community information has been used differently for prediction as a characteristic of the nodes. The major drawback of these methods involves the high time complexity of obtaining a clustering in large-scale networks and computation for both nodes. A method referred to as the spectral algorithm has been presented in [3]. The approach is similar to a semi-local method, which uses neither local information nor general network paths, making it highly time-consuming and infeasible in large networks.

The authors of [2] have proposed a distributed method based on clustering for link prediction, which depends on Google's MapReduce technology. Although, it has been mentioned in the paper's abstract that clustering is performed basically on dispersed vertices, so that they are grouped in an integrated fashion, the paper does not claim that it applies to sparse graphs.

2-2- A review of link Prediction in Sparse Graphs

Although numerous works on link prediction have been presented that have attempted to improve the precision of the results, the sparsity has been slightly considered in the works. However, many real-world networks are sparse, which causes poor prediction results and loss of time. Hence, the question in some works since 2013 is what is the best way of avoiding this issue [3]. In a supervised solution, the method used in [16] has utilized incidence rather than adjacency matrix factorization, demonstrating that the incidence matrix factorization (IMF) method performs better than adjacency matrix factorization (AMF) in a sparse matrix as well.

It has been mentioned in [17] that the available link prediction algorithms have focused on triangular structures. The method exhibits low efficiency over sparse tree networks. A method based on network degree heterogeneity has been presented in that paper. As authors stated, however, they have examined only tree structures, whereas many complex networks in the real world are sparse, and do not necessarily contain tree structures. In [18], it has been assumed that social network users' habits and characteristics correspond to their social communication on the networks. That is, links are predicted through the notion of aligned social networks.
Besides, in [19], the focus is mainly on the structure of the network, and the paper models the problem based on intrinsic characteristics of the network. A drawback of these models is the cost of training the model to handle the big data. Another interesting research used for sparse networks is [20], where the relationship between clustering and the precision of the methods has been investigated based on network structure.
Even though, all the unsupervised link prediction methods mentioned above attempt to improve results, reducing the extra computation in sparse networks by splitting it into separate communities and so improving the results in this way has not been considered yet. In this article, we try to overcome the poor result of link prediction in the sparse networks by dividing networks into multiple communities and concentrating on the inter and intra community computations. Subsequently, we will shorten the execution time of link prediction in the sparse networks also.

3- Proposed Method

The algorithm presented in the proposed method involves three major phases, and the validity of each phase affects the final results. The first step is data preparation and pre-processing, which is explained in the next paragraph. Other steps are clustering the network into some partitions and performing the link prediction inter and intra communities, and integrating the results according to some specific policies. The steps are mentioned below.

3-1- Data

Since some datasets are directional, we need to convert them to un-directional graphs because of the nature of the basic link prediction algorithms that do not consider the direction of nodes [21]. First, the dataset is mapped to a matrix, then the matrix needs to be symmetric, and the elements on the main diameter needed to be zero. In this research, five datasets have been used for experimentation. Email[1] (the email communications at the Rovira i Virgili University), the collaboration network on high-energy physics[2], the collaboration network of co-authors on physics-related topics on the arXiv website[3], the communication network of associated words[4], and the communication network of human protein[5]. Table 2 describes the properties of each data set, respectively.
The quality and precision of link prediction in this research depend to a large extent on correct cluster detection. The utilized clustering methods are as follows. Fast unfolding [22] is a link-based community detection algorithm. Link-community [23] which finds communities such that it may contain nodes overlapping others. Another method used in this research involves the InfoMap community detection algorithm [24], [25]. The Girvan-Newman algorithm utilizes the edge betweenness feature [26].

Table 2: Examined networks and their basic properties

Network	Nodes	Edges	Mean clustering coefficient	Density
Email	1133	5452	0.22	0.0085
Collaboration network on high-energy physics	9877	25998	0.47	0.0005
Collaboration network on general physics communication	5242	14496	0.52	0.001
Network of associated words	23219	305500	0.099	0.001
Wikipedia's network of manager selection	7115	100762	0.14	0.003
Human protein communications	30047	41327	0.101	0.00009

3-2- Cluster-Based Sparse Link Prediction (CBSLP)

For easy referencing to the algorithm, the abbreviation CBSLP, which stands for Cluster-Based Sparse Link Prediction, has been used hereafter. The data are first mapped into a graph after pre-processing, and the community detection algorithms mentioned in the previous section are then applied to them (line 9 of Figure 2). Prediction is made within each community; thereafter a matrix is defined for the inter-community step, in the relevant entries of which, all the edges between pair of communities are located. All the edges are traversed for finding inter-community edges, and each edge is inserted in the relevant entry of the matrix. Thus, graphs of inter-

[1] http://konect.uni-koblenz.de/networks/arenas-email

[2] http://snap.stanford.edu/data/ca-HepTh.html
[3] http://snap.stanford.edu/data/ca-GrQc.html
[4] http://vlado.fmf.uni-lj.si/pub/networks/data/dic/eat/Eat.htm
[5] http://www.hprd.org/

community edges are finally obtained. Next, each of the communities is subject to link prediction, each of the four basic neighborhood algorithms is examined (Table 1), and new links are predicted. Of course, probable repetitive edges resulting from the prediction in both steps are eliminated (Figure 2).

3-2-1- Intra-Cluster link Prediction

Using community detection, we divide the whole graph into several separated subgraphs that can be investigated independently for link prediction with more confidence of the closely connected links for better prediction results. Performance of CBSLP is as well as a divide and conquer method. First of all, seeking for communities and after that searching for the relation between those communities is performed. As seen in Figure 1(a), the obtained communities are represented as C_i. C_1 and C_2 are two of these clusters. Edges and vertices located in a single community are separated, and prediction is made within each of the communities, as clear from Figure 1(b). For edges indicted by dashed lines, link prediction is very likely made with the basic methods.

3-2-2- Inter-Cluster link Prediction

After dividing the main graph into communities and predicting the intra links in each community, it is necessary to investigate the probable links between each pair of communities. Because there are certainly several edges between communities that have not been considered in the calculations.

Here, we generate a graph between every two separate communities for the interconnected edges, and predicts links within each connected pair of the communities. The number of communities depends on the community detection algorithms. Some algorithms, like Best partition, automatically determine the appropriate number of communities, while some other clustering algorithms need a predefined number to break down the network into that number of communities. We utilize the elbow method to automatically determine the number of communities.

In order to perform the inter-community link prediction, first, we collect the common links between every two communities. Then we consider and add the links between the nodes located in each community, that participate in inter-community relations for the increment of the accuracy of the computations. For example, in Figure 1(d), we form an inter-community network including the {(a,b), (b,c), (c,d), (e,i), (i,h)} U {(a,h), (d,e)} edges.

Thus, the inter-community edges are taken into account, the total capacity of the network is used for prediction, and extra calculation is avoided at the same time as well. Traversing all the common edges between communities for finding inter-community relations that participate in the intersection communities' results in isolating new communities between pair of connected communities. Figure 1(c) shows the approach for two different communities. Implementation of the proposed method using inter-communities link prediction is also shown in Figure 2.

3-3- Evaluation

Three factors can be used for measuring the success of link prediction in large sparse networks: precision, AUC (Area Under Curve), and runtime. To calculate precision, 10-fold cross validation is performed. For each fold, 10% of the existing links are removed randomly to predict by the algorithm again. This is done ten times, and each time, a different 10% of the links are selected to be removed. This ensures that each link is withheld exactly once, so all links are present in the training data and the test data an equal number of times. Another evaluation metric for link prediction in unsupervised methods is AUC, also. It can be interpreted as the probability that a randomly chosen missing link is given a higher similarity score than a randomly chosen pair of unconnected links. If among n independent comparisons, there are n′ times the missing link having a higher score and n″ times they have the same score, the AUC value is calculated as the following[13]:

$$\text{AUC} = \frac{n' + 0.5n''}{n} \qquad (1)$$

The link prediction detailed above is taken, with an accurate chronometer measuring the time from the beginning to the end of the implementation, and average time, *i.e.*, mean runtime in each of the ten iterations, is calculated. This measure can be used for the assessment of the algorithm speed.

4- Results and Discussion

In this section, we will investigate the results of using the proposed method from different viewpoints including: decreasing the number of checked edges, comparing the best performance link prediction functions, and runtime comparison of CBSLP with basic methods.

4-1- Number of Edges under Examination

An interesting difference between the proposed method and the basic algorithms such as AA, PA, JC, and CN, lies in the numbers of edges and nodes under examination. This causes computations to be carried out in shorter times, regardless of the processing hardware that has been utilized, leading to good results over sparse networks. A summary of the comparison is provided in Table 3. It is

worth paying attention in CBSLP that we attempted to remove or ignore the lowest importance links. This table demonstrates the number of initial zero entries in the similarity matrix that should be calculated by basic methods and the proposed method. For example, for the

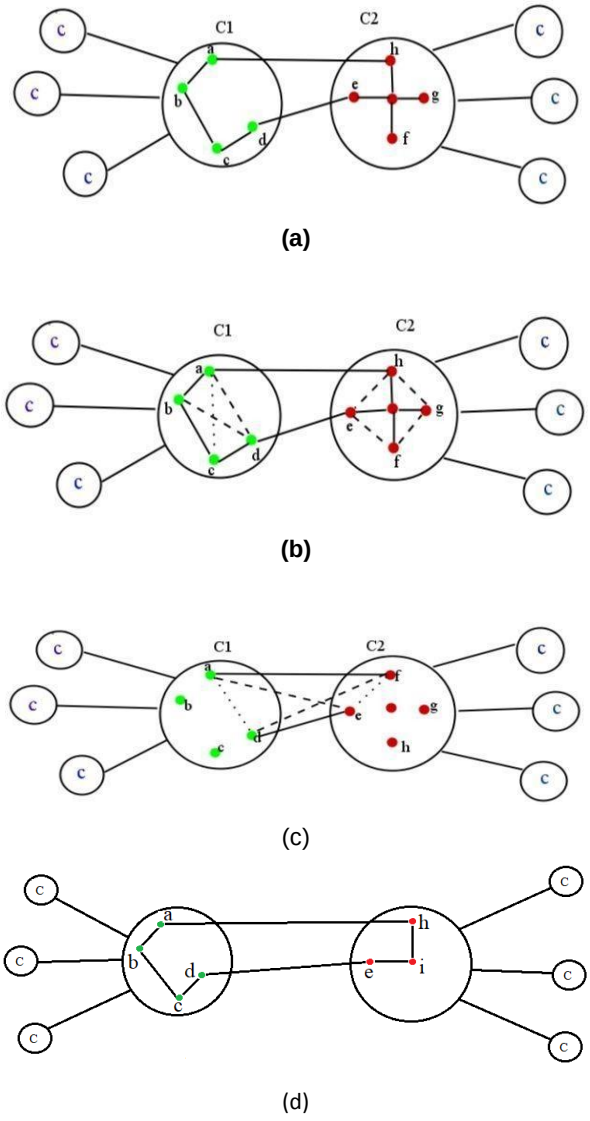


Fig 1: Detection of clusters and prediction of links within and between communities (a) Sample for obtained communities, such as C_1 and C_2 after running community detection algorithms (b) Performing intra-community link prediction (c) Considering inter-community links, (d) Inter-community graph formation based on the related links, for complementary link prediction.

Email dataset, the former methods have about 641844 calculations, while the latter method makes this value lower approximately one-fourth about 163350 in the worst case. Indeed, there are some inter community edged that should be taken into account, but they are few and can be ignored.

CBSLP(Cluster-Based Sparse Link Prediction)

[1]**Input: array of dataset as data$_a$**
[2]**Output: list of prediction item conducting CBSLP algorithm**

First phase starting (intra cluster)
[3] $array_t \leftarrow data_a$
[4] For item in $array_t$:
[5] **make $array_t$ symmetric and remove self loop and put in $array_{cp}$**
[6]**Return $array_{cp}$**
[7]**For** item **in $array_{cp}$**:
[8] **C**reate graph G_t from $array_{cp}$
[9]**R**un community detection algorithm ϵ
{best partition, link community, infomap,Girvan and newman} on G_t
[10]**For** $node_1$ **in** G_t:
[11] $node_1 \in \{C_1, C_2, C_3, \ldots, C_n\}$ # C_i Means community
[12]**Return** Clusters
[13]**Create** subgraph G_t
[14]**For** C_i **in** range (C_1, C_2, C_3,…., C_n)
[15] **IF** (Len C_i>2):
[16] $G_s \leftarrow G_i$
[17] Tupple(x,y,**score**)$_{fg}$ ←basic approach ∈ {AA,PA,CN,RA,Jc}(G_s)
[18] $G_s \leftarrow \emptyset$
[19]Return $Tupple_{fg}$ #can use output of first phase
First phase end
Second phase starting(inter cluster)
[20]array[][]$_a \leftarrow \emptyset$ #Make two-dimensional array
[21]**For** $edge_{u,v}$ **in** G_t:
[22] **If** (C_u !=C_v):
[23] $array[u][v]_a \leftarrow (u, v)$
[24]**For** edge$_{u,v}$ **in** G_t:
#Find edges between each two clusters and then make subcluster
[25] $array[u][v]_a \leftarrow neighbours(u, v)$
#Find all neighbor to make prediction more accurate
[26]$G_s \leftarrow \emptyset$
[27]**For** C_i **in** $array_a$: #**edges were put in** *array*[][]$_a$ **from cluster C**
[28] **If** (**len** c_i> **2**):
[29] $G_s \leftarrow C_i$
[30] $item$ ←basic approach ∈ {AA,PA,CN,RA,Jc}(G_s)
[31] for(item in range(0,len(Tupple$_{fg}$):
[32] #merge of result
[33] **If** (item **not in** Tupple$_{fg}$):
[34] **append** item to Tupple$_{fg}$
[35] **else:**
[36] Compare their score remove the less one from Tupple$_{fg}$
[37] Sort Tupple$_{fg}$ again by score
[38] $G_s \leftarrow \emptyset$
[39] Return Tupple$_{fg}$ #including cumulative results
Second phase end (inter cluster)

Fig 2: Pseudo-code of the proposed method for link prediction in sparse networks

4-2- Comparison with Similar Competing Methods

For evaluation of CBSLP, its performance is compared with primary methods. In Table 4, a summary of the results obtained by the proposed method is provided, along with

comparing to those of different community detection methods mentioned above. It should be mentioned that the column containing the cumulative results involves the overall results obtained from both intra-community and inter- community phases. The proposed method has no claim on dense graphs such as HEP or Rel, because it may not be appropriate for such a graph structure in a particular application, and may also be led to the elimination of valuable predictions from the graph. In Table 4 BP, LC, info are the abbreviations of best partition, link community and Infomap respectively where all of them are community detection methods that were mentioned before. The bold numbers show the best result in each column of Table4. As a result, CBSLP achieved better results in sparse networks such as Email, Word, Wiki-Vote, PPI. It is worth to mention that (–) in each column means that the pertaining method could not terminate the calculations within a reasonable time (72 hours). Another evaluation metric is AUC. Results in table 5 also confirm the precision metric findings.

Table 3: Comparison between the number of calculations in CBLSP and basic methods. Dividing the investigating graph into clusters reduces the total numbers of calculations considerably. Columns three to seven are the top most populated clusters for each dataset in order to take into account the upper bound calculation numbers in CBLSP method in comparison.

Network	Total number of clusters	Node-cluster 1	Node-cluster 2	Node-cluster 3	Node-cluster 4	Upper bound of the sum of entries examined in the CBSLP	Size of the matrix examined in the basic methods
Email	12	165	165	134	126	12*(165*165)/2=163350	641844
HEP	209	756	620	418	410	209*(756*756)/2=59725512	48778564
REL	210	308	267	258	251	210*(308*308)/2=9960720	13739282
Word	9	4211	3354	3216	3096	9*(4211*4211)/2=79796344	269560980
Wiki-vote	6	1704	1610	1593	1384	6*(1704*1704)/2=1451808	25311612
PPI	48	1497	1152	925	777	48*(1497*1497)/2=53784216	450000000

Table 4: Precision results obtained from the basic methods, and the proposed method using different clustering algorithms. Cells with dash sign are the calculations has not committed in a rational time, 72 hours of computation with our hardware. Precision of the CBSLP for the inter-community relations is not remarkable compared to intra-community results.

Network	Basic methods	CBSLP BP method	CBSLP LC method	CBSLP INFO method	CBSLP Girvan-Newman method	CBSLP with cumulative results	Precision of the proposed method for inter-communities
Email	0.141AA	**0.146AA**	0.139AA	0.141AA	0.141AA	0.141AA	0.033(AA)
HEP	**0.37CN**	0.35CN	0.37CN	0.35CN	0.35CN	0.34CN	0.033(CN)
REL	**0.5RA**	0.49RA	0.48RA	0.42RA	0.42RA	0.49RA	0.04(RA)
Word	-	-	**0.11AA**	-	-	0.1RA	0.021(AA)
Wiki-vote	0.09RA	**0.11AA**	0.09RA	0.11AA	-	0.11AA	0.036(AA)
PPI	0.06AA	**0.062AA**	0.057	0.043	-	0.062AA	0.014(AA)

Table 5: AUC results obtained from the basic methods, and the proposed method using different clustering algorithms. Cells with dash sign are the calculations has not committed in a rational time, 72 hours of computation with our hardware.

Network	Basic methods	CBSLP BP method	CBSLP LC method	CBSLP INFO method	CBSLP Girvan-Newman method	CBSLP with cumulative results	AUC of the proposed method for inter-communities
Email	0.87AA	**0.89AA**	0.83AA	0.821AA	0.823AA	0.821AA	0.95(AA)
HEP	**0.597CN**	0.591CN	0.592CN	0.63CN	0.621CN	0.61CN	0.80(CN)
REL	**0.63RA**	0.624RA	0.611RA	0.655RA	0.63RA	0.62RA	0.79(RA)
Word	-	-	**0.89RA**	-	-	-	-
Wiki-vote	0.88RA	**0.91RA**	0.90RA	0.90RA	-	0.91RA	0.91(RA)
PPI	0.91AA	**0.92AA**	0.91AA	0.91AA	-	0.89AA	0.89(AA)

4-3- Runtime Analysis and Comparison

In the above four sections, it was discussed that the basic methods have not been successful in link prediction over the Word network, and could not solve it within a reasonable time (72 hours). It is also noticeable that the basic methods CN could probably not be implemented over several similar large networks within a logical time, while the CBSLP in this research successfully computed a sample within a proper time. Therefore, this method has improved time as well, as shown in Table 3. The specification of the system used in this research is shown in Table 6.

Table 6: Specification of the system used in the research

Processor	**Intel Core i5 3250M**
Main Memory	8 Gigabytes
Hard disk memory	500 Gigabytes
Operating system	Linux Ubuntu

First, it is necessary to know what percentage of the links would be predicted correctly if and only if the links and nodes between communities were investigated and evaluated with the methods introduced in the inter-community step. The answer could be found in the Precision of inter-community results column in Tables 4 and 5. The runtimes of methods was calculated for each dataset, and the results can be observed in Table 7.

Table 7: Runtimes of the proposed method, and the basic methods

Network	**Runtime of the clustering method (Mean in one iteration)**	**Runtime of the basic (Mean in one iteration)**
Email	1-2 minutes	5-10 minutes
HEP	4-5 minutes	25-30 minutes
REL	3-4 minutes	15-20 minutes
Word	15 minutes	6 hours
Wiki-vote	2 hours	-
PPI	30 minutes	5 hours

Clearly, about 0.031 of the links predicted to occur between communities over a network like Email, which means that about 20% of the links occur between communities rather than within them. Unfortunately, however, not much change occurs when the inter- and intra- community links are predicted and evaluated at the same time, as clear from the proposed method with cumulative results' column in Table 4. This is because two lists with different scores are merged, which causes the scores to drift on the list with higher precision, and the results not to change and the final result to worsen even. If the results are cumulated correctly, the method will definitely succeed in denser graphs as well.

5- Conclusion and Future Works

The proposed method, CBSLP, involves a framework for large sparse graphs, since it prevents extra computation, improves runtime, and saves memory. Besides, it can be regarded as a new link prediction method for sparse networks due to its strategy details. However, CBSLP is an initial version of the framework, which should evolve greatly. In the proposed method, clustering was used as a tool not only for improvement of the prediction results but also for elimination of extra calculation. In addition, there is a lot that needs to be done for its evolution. For the precision of the proposed method to increase, attempts can be made to make link prediction also using path-based methods. An appropriate method among path-based algorithms that is recommended in sparse graphs is the SRW[1] method, which improves the results probably. One can attempt to experiment newer and better community detection algorithms for higher precision, such as [27] or [28]. Moreover, a mechanism has been sought to utilize weighted graph version of the network for improvement of the results using inter-cluster relations and their outcomes. It is possible even applying rank aggregation to link prediction lists with different scores for achieving better results. Methods such as that in [15] or [29] can be used to employ cluster information in order to improve the proposed method in terms of precision.

References

[1] D. Caiyan, L. Chen, and B. Li, "Link prediction in complex network based on modularity," Soft Comput., 2016, doi: 10.1007/s00500-016-2030-4.

[2] H. Yuan, Y. Ma, F. Zhang, M. Liu, and W. Shen, "A distributed link prediction algorithm based on clustering in dynamic social networks," in 2015 IEEE International Conference on Systems, Man, and Cybernetics, 2015, pp. 1341–1345.

[3] P. Symeonidis and N. Mantas, "Spectral clustering for link prediction in social networks with positive and negative links," Soc. Netw. Anal. Min., vol. 3, no. 4, pp. 1433–1447, 2013.

[4] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," J. Am. Soc. Inf. Sci. Technol., vol. 58, no. 7, pp. 1019–1031, 2007.

[5] G. SALTON and M. J. MCGILL, "Introduction to Modern Information Retrieval (pp. paginas 400)." 1986.

[6] M. E. J. Newman, "Clustering and preferential attachment in growing networks," Phys. Rev. E, vol. 64, no. 2, p. 25102, 2001.

[1] Superposed Random Walk

[7] P. Wang, B. W. Xu, Y. R. Wu, and X. Y. Zhou, "Link prediction in social networks: the state-of-the-art," Sci. China Inf. Sci., vol. 58, no. 1, pp. 1–38, 2014, doi: 10.1007/s11432-014-5237-y.

[8] M. K. Khouzani and S. Sulaimany, "Identification of the effects of the existing network properties on the performance of current community detection methods," J. King Saud Univ. - Comput. Inf. Sci., Apr. 2020, doi: 10.1016/j.jksuci.2020.04.007.

[9] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, "A network-based method for target selection in metabolic networks," Bioinformatics, vol. 23, no. 13, pp. 1616–1622, 2007.

[10] N. Benchettara, R. Kanawati, and C. Rouveirol, "A supervised machine learning link prediction approach for academic collaboration recommendation," in Proceedings of the fourth ACM conference on Recommender systems, 2010, pp. 253–256.

[11] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee, "Self-organization and identification of web communities," Computer (Long. Beach. Calif)., vol. 35, no. 3, pp. 66–70, 2002.

[12] A. Clauset, C. Moore, and M. E. J. Newman, "Hierarchical structure and the prediction of missing links in networks," Nature, vol. 453, no. 7191, pp. 98–101, 2008.

[13] Z. Liu, Q.-M. Zhang, L. Lü, and T. Zhou, "Link prediction in complex networks," EPL (Europhysics Lett., vol. 96, no. 4, p. 48007, 2011.

[14] E. M. Airoldi, D. M. Blei, S. E. Fienberg, E. P. Xing, and T. Jaakkola, "Mixed membership stochastic block models for relational data with application to protein-protein interactions," in Proceedings of the international biometrics society annual meeting, 2006, vol. 15.

[15] J. H. S Soundarajan, "Using community information to improve the precision of link prediction methods," WWW (Companion Vol., vol. 2012, pp. 607–608, 2012.

[16] S. Yokoi, H. Kajino, and H. Kashima, "Link prediction in sparse networks by incidence matrix factorization," J. Inf. Process., vol. 25, pp. 477–485, 2017.

[17] K. Shang, T. Li, M. Small, D. Burton, and Y. Wang, "Link prediction for tree-like networks," Chaos An Interdiscip. J. Nonlinear Sci., vol. 29, no. 6, p. 61103, 2019.

[18] J. Zhang, J. Chen, S. Zhi, Y. Chang, S. Y. Philip, and J. Han, "Link prediction across aligned networks with sparse and low rank matrix estimation," in 2017 IEEE 33rd International Conference on Data Engineering (ICDE), 2017, pp. 971–982.

[19] C. H. Nguyen and H. Mamitsuka, "Latent feature kernels for link prediction on sparse graphs," IEEE Trans. neural networks Learn. Syst., vol. 23, no. 11, pp. 1793–1804, 2012.

[20] X. Feng, J. C. Zhao, and K. Xu, "Link prediction in complex networks: a clustering perspective," Eur. Phys. J. B, vol. 85, no. 1, p. 3, 2012.

[21] H. Liu, Z. Hu, H. Haddadi, and H. Tian, "Hidden link prediction based on node centrality and weak ties," EPL (Europhysics Lett., vol. 101, no. 1, p. 18004, Jan. 2013, doi: 10.1209/0295-5075/101/18004.

[22] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," J. Stat. Mech. theory Exp., vol. 2008, no. 10, p. P10008, 2008.

[23] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," Nature, vol. 466, no. 7307, pp. 761–764, 2010.

[24] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," Phys. Rev. E, vol. 80, no. 5, p. 56117, 2009.

[25] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," Proc. Natl. Acad. Sci., vol. 105, no. 4, pp. 1118–1123, 2008.

[26] K. Esders, "Link Prediction in Large-scale Complex Networks," Bachelor's Thesis Karlsruhe Inst. Technol., no. February, 2015, [Online]. Available: https://hackernoon.com/link-prediction-in-large-scale-networks-f836fcb05c88.

[27] M. Hajiabadi, H. Zare, and H. Bobarshad, "IEDC: An integrated approach for overlapping and non-overlapping community detection," Knowledge-Based Syst., vol. 123, pp. 188–199, 2017.

[28] H. Zare, M. Hajiabadi, and M. Jalili, "Detection of community structures in networks with nodal features based on generative probabilistic approach," IEEE Trans. Knowl. Data Eng., 2019.

[29] H. Zare, M. A. N. Pour, and P. Moradi, "Enhanced recommender system using predictive network approach," Phys. A Stat. Mech. its Appl., vol. 520, pp. 322–337, 2019.

Mohammad Pooya Salavati received his MSc. degree in Computer Engineering from Urmia University, West Azarbaijan, Iran in 2020. His research interests include Link prediction, Community detection and Big data analysis.

Jamshid Bagherzadeh Mohasefi received bachelor of computer engineering from Sharif University of Technology in Iran at 1996 and master of computer engineering from Tarbiat Modares University in Iran at 1999. He got his PhD in computer engineering from Indian Institute of Technology Delhi (IITD) in India at 2006. He joined Urmia University as a faculty member in 2006. He has worked since there in two major fields including information security and artificial intelligence. He has published more than 80 referred journal and conference papers. He is currently associate professor at Urmia University, Iran. His current research interests include machine learning, data/text mining, and network security.

Sadegh Sulaimany is assistant professor at the department of computer engineering in University of Kurdistan, Iran. He received his PhD in Bioinformatics from Tehran University in 2017. His Master degree is in Computer Science from Amirkabir University of Technology, Tehran, and his Bachelor degree is in Computer Engineering from Isfahan University of Technology, Isfahan, Iran. His research interest lies in but not limited to the biological and social network analysis, especially link prediction algorithms for different application areas.

Diagnosis of Gastric Cancer via Classification of the Tongue Images using Deep Convolutional Networks

Elham Gholami
Department of Computer Engineering, Neyshabur branch, Islamic Azad University, Neyshabur,Iran
gholami.elh@gmail.com
Seyed Reza Kamel Tabbakh *****
Department of Computer Engineering, Mashhad Branch,Islamic Azad University, Mashhad, Iran
rezakamel@computer.org
Maryam Kheirabadi
Department of Computer Engineering, Neyshabur Branch,Islamic Azad University, Neyshabur,Iran
maryam.abadi@gmail.com

Received: 09/ Sep /2020 Revised: 01/May/2021 Accepted: 04/Jun/2021

Abstract

Gastric cancer is the second most common cancer worldwide, responsible for the death of many people in society. One of the issues regarding this disease is the absence of early and accurate detection. In the medical industry, gastric cancer is diagnosed by conducting numerous tests and imagings, which are costly and time-consuming. Therefore, doctors are seeking a cost-effective and time-efficient alternative. One of the medical solutions is Chinese medicine and diagnosis by observing changes of the tongue. Detecting the disease using tongue appearance and color of various sections of the tongue is one of the key components of traditional Chinese medicine. In this study, a method is presented which can carry out the localization of tongue surface regardless of the different poses of people in images. In fact, if the localization of face components, especially the mouth, is done correctly, the components leading to the biggest distinction in the dataset can be used which is favorable in terms of time and space complexity. Also, since we have the best estimation, the best features can be extracted relative to those components and the best possible accuracy can be achieved in this situation. The extraction of appropriate features in this study is done using deep convolutional neural networks. Finally, we use the random forest algorithm to train the proposed model and evaluate the criteria. Experimental results show that the average classification accuracy has reached approximately 73.78 which demonstrates the superiority of the proposed method compared to other methods.

Keywords: Gastric Cancer; Deep Convolutional Networks; Image Classification; Fine-grained Recognition.

1- Introduction

Cancer is the second leading cause of death after cardiovascular diseases throughout the world [1]. Gastric cancer has a high mortality rate [2] and is the fourth prevalent cancer type, as well as the second most fatal cancer [3, 4]. However, the emergence of gastric cancer has decreased, especially in developed countries [5, 6]. In Iran and unlike most developed countries, the emergence of gastric cancer is on the rise, which is particularly significant in the north and northwest of Iran [8].

Traditional Chinese medicine (TCM) is a form of natural and comprehensive healthcare system dating back to 3-5 thousand years ago [13]. TCM has historically been used for the treatment of various diseases in East Asia and is known as a complementary and alternative medical system in western countries [15]. In TCM, diseases are diagnosed based on the information obtained through observing, hearing, smelling, and touching. Most diagnoses are based on pulse measurement and tongue examination [18].

In Chinese medicine, the tongue is essentially examined as it is located in the mouth and is not affected by external and environmental factors [19]. Recently, several studies have investigated gastric cancer diagnosis using the images of the color and texture of the tongue [14, 16, 18]. Object, texture, and component recognition in an image are the most important criteria in image processing, and the main challenge in this regard is the diversity of natural images, which results from differences in objects and cameras, illumination variations, movement, metamorphosis, and background congestion. In this study, we used an approach based on recent developments in deep learning for the visual recognition of tongue images, and a new method was proposed based on deep convolutional networks and random forest to solve fine-grained image classification, which could be applied in other areas than the detection of tongue texture images.

* Corresponding Author

2- Literature Review

Recently, significant developments have been achieved in image classification. Image classification has also become a commercial and applied issue within the past decade rather than a research subject. In the present study, we initially evaluated the basic classification method and achieved developments. Fine-grained classification has recently attracted the attention of researchers as well. For instance, a human recognizes a chair by recognizing its components, such as the stands and back. The ability to recognize the side level is associated with the ability to discriminate between similar objects; this observation inspired our proposed method and other approaches.

In a study in this regard, Branson et al. assessed object classification using a semi-automatic method where the user would be asked about the object for a given set of images, and the type of a bird was recognized based on the user's response. In the mentioned study, the accuracy of the applied method was reported to be 19% [20]. On the other hand, Wlinder et al. performed automatic classification using color histogram characteristics and the KNN classifier, reporting low accuracy [21]. To increase the accuracy of the method, Moghimi extracted features from the area where the probability of the presence of an object was high. In the mentioned study, the area where an object was present was selected manually, ultimately resulting in the accuracy of 18.9% [22].

Zhang et al. proposed a method to significantly match the sections and feature extraction with a smaller dimension. Initially, the important areas were selected using the selective search algorithm. Following that, the SVM classifier was used to detect the areas with the maximum score, which resulted in the accuracy of 82.8% [24]. In another research, machine learning was applied to determine the components and extract features, and the optimal results were obtained with the CUB-200-2011 database.

The methods used in the aforementioned studies could be classified into three categories. The first category includes the primary methods with the classification accuracy of 10-30%, in which conventional classification methods are often used for fine-grained classification without remarkable results [17, 23, 25, 26]. The second category includes the methods that are used to better recognize the granular classification problem with the reported accuracy of 40-60%; the low accuracy of these methods might be due to non-deep features. The third category contains the methods based on deep learning, which are used to solve this problem with the accuracy of 80-90% [12, 37, 41]. The proposed method in our study has been classified into the third category.

3- The Proposed Method

We proposed a method to locate the tongue surface independent of various gestures in images. If the face components (especially the mouth) are located correctly, the component resulting in a higher discrimination of the dataset can be used, which is cost-efficient in terms of tempo-spatial complexity. Since we had the optimal estimate of the components, the optimal features could be extracted, and the best possible accuracy was calculated as well. The initial image of the tongue was with the face and head, and the dimensions of the initial raw image were 2988×5312 pixels. Figure 1 shows an example of a raw (initial) image of the healthy image category.

Fig. 1. Example of Initial Image

As mentioned earlier, the problem of fine-grained image classification in computer vision has been resolved by deep convolutional networks, which are highly diverse and very similar to each other in terms of structure. In the present study, we applied the deep network structure of AlexNet, which consists of eight main layers. Figure 2 depicts a schematic view of the selected network.

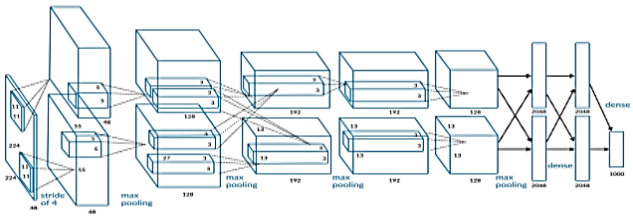


Fig. 2. AlexNet Network Structure

Notably, we had a different view of neural networks in this study. When it comes to neural networks, they are often viewed as 1D connected layers, while in convolutional neural networks, the layers are considered as 3D information. Training is considered to be a major process of deep convolutional networks. Since these networks have more than 120 million parameters, their training is rather difficult due to over-fitting problems. In addition, forward pass for calculating the values of all the network nodes layer by layer using the input information and backward pass for calculating the errors and network learning are time-consuming processes. The network

retuning method is aimed at applying deep learning methods to small databases.

The fine-tuning of a network is used to enhance transmission learning, with the new database used for learning. In this method, a pre-trained network is used on a database with more images (e.g., ILSVRC) as the initial values for another network that is similar to the target network where the only difference is in the probability generator layer. The target network differs from the origin network only in terms of the number of the outputs of the probability layer. Therefore, the weights of the probability layer should be estimated again. On the other hand, the layers before the probability layer in the target network could be initialized using the learned weights of the corresponding layers in the original network. As a result, learning is carried out using the data of the target database. Our proposed method was divided into training and test steps, which have been further discussed below.

3-1- Training Step

The training step had three main stages, as follows:

1. Random forest training: A rectangle represented the location of each element of the image and used for training. The random forest model could distinguish the pixels inside the rectangle from those outside the rectangle.

The required training samples for random forest training were the pixels of a component and the pixels not belonging to the component. In addition, each rectangle could be considered as a component; such examples are the peripheral rectangle of the individual's mouth and the peripheral rectangle of the tongue (Figure 3).

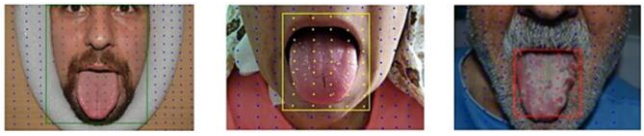


Fig. 3. Different Examples of Random Forest Training (In the image on the right, the red rectangle is the peripheral rectangle of the individual's tongue; the red dots in the red rectangle belong to the component, and the blue dots do not belong to the component; in the middle image, the yellow dots belong to the component; in the image on the left, the green rectangle is the peripheral rectangle of the individual's face, and the positive points inside belong to the component)

The training data required by the random forest were obtained using the following equation:

$A = \{(X_i . Y_i)\}$.

where $Y_i = +1$ shows the ith data belonging to one pixel of the images of the database and the component of interest, while the ith data does not belong to the component of interest. In Figure 3, the blue dots are the negative points ($Y_i = -1$), and the red ones ($Y_i = +1$) are the positive points. The algorithm used to generate the training data of A was implemented in several steps. For each training image, the peripheral rectangle of the components was obtained, and its zoning was also prepared. The deep features of each pixel were calculated as well. An arbitrary number of pixels (20 pixels) were generated inside the peripheral rectangle using random or uniform approaches. Each pixel inside the tongue region was added to set A, and its deep feature were considered as positive data. Moreover, an arbitrary number of pixels (100 pixels) were generated in the entire image using random or uniform approaches. Each pixel outside the peripheral rectangle and its deep features were also added to set A as negative data.

After completing the training dataset A, the model was trained using the random forest. Due to using one forward pass in the neural network and the random forest (including 10 decision trees with maximum depth of 10), the likelihood estimate of the membership of all the pixels could be calculated rapidly.

2. Retuning the deep convolutional neural network (DCNN): Three DCNNs were used for piecewise feature extraction and retuned for the entire image, mouth image, and tongue image. The purpose of using the features of the layers of these networks was for one feature vector to be generated per each pixel. To calculate the deep pixel features of each image or image section, the image was fed as input to the network, and forward pass was calculated once. Following the forward pass, the values of the features of the input image were also calculated in all the images and referred to as the feature channels (Figure 4). As is shown in Figure 4, the feature channels had a specific size, which was changed by up-sampling. Notably, one of the limitations of this method was the size of the input network. Currently, the input image size should be 227×227 to use the employed network.

After changing the size of the feature channels to the input image size, all the feature channels were inserted into the main channel to constitute a column of feature channels with the length of 1,376.

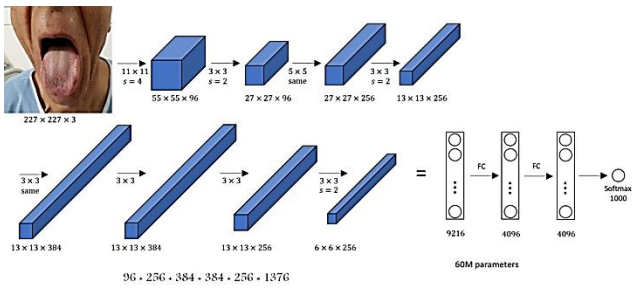


Fig. 4. Calculation of Feature Channels in Different Layers (After giving an arbitrary image to the input, all the feature channels of the middle layers or the data values in the middle layer were calculated by forward pass; the size of the channels was changed to the size of the input (up-sample); 1,376 different channels were obtained; the total depth of layers conv1 to conv5 in the AlexNet network was 1,376 with the same dimension as the input image.)

The corresponding random forest was used to retune the mouth and tongue area, and the estimated peripheral rectangle was extracted from the training data. At the next stage, this

image section was cut, and retuning was performed. To extract the piecewise features, the following steps were taken:

- We selected the image section from which the features had to be extracted.
- The selected image section was resized to the dimension of the input data (222×227), and the resized image was fed as input to the network.
- By calculating the forward pass, the values of the layers were calculated for the network input.
- The values of the data in the fc7 layer (4,096-dimensional) were retuned as a feature.

The proposed method could generate an excellent 4,096-dimensional feature vector for the arbitrary segments of multiple images or one image. The feature vector could be fed as input to classifiers such as the SVM.

3. Classifier training: For the final classifier, the one-vs.-all method was used; the employed classifier was an SVM with a linear core. The final feature vector of each image was obtained by combining the three-piece deep features extracted from the mouth and tongue of an individual and the entire image.

3-2- Test Step

In the test step, the locations of the components and feature extraction were determined for each test image. Following that, the classifier was used to estimate the test image class. The peripheral rectangle of the tongue, the peripheral rectangle of the mouth area, and the peripheral rectangle of the face were also estimated for each test image using the random forest algorithm. The retuned neural networks were used to extract the three-piece deep features similar to the training step. The final vector was obtained by combining these feature vectors, and the final feature vector was classified as healthy and unhealthy using the classifier.

4- Results

The empirical results of the proposed method have been discussed in this section. For the test, we used the database of the tongue images, which contained 700 images of the patients and 800 images of the healthy subjects. For the training, 500 healthy images and 470 patient images (total: 970 images) were used. In addition, 300 healthy images and 330 patient images (total: 630 images) were used for the test. Initially, the evaluation metrics of the classification in the database were introduced, and the proposed method was assessed based on these metrics. Notably, we had limited choices for the evaluation of the general image classification and fine-grained image classification. If a computer vision such as object detection, pose estimation or the segmentation of the estimated class were considered equivalent to the real class, the classification would be correct; otherwise, the classification would be incorrect. For more accurate outcomes, the classification accuracy metric was assessed in detail.

We assumed that the test data were in the form of ordered pairs such as $\{(X_i, Y_i)\}_{i=}^{N_{test}}$, in which X_i is the image, Y_i is its label, and $Next_{test}$ is the number of the test images. The following formula was used to calculate the mean accuracy to evaluate the classifier function (f[x]):

$$mA = \frac{1}{C}\sum_{c=1}^{c}\frac{1}{|\propto(c)|}\sum_{i\in\propto(c)} I(f(X_{i)=}\ Y_{i)}$$

In the formula above, C is the number of the classes, and $\propto(c)$ shows the set of the indices of the test data belonging to class C, which would be calculated as:

$$\propto(c) = \{i|Y_i = c\}$$

Notably, $\propto(c)$ represents the number of the test samples belonging to class C, and II(.) is the mathematical indicator function. The value of mA was the mean classification accuracy among different classes.

The proposed deep non-parametric transfer (DNPT) method was evaluated based on various features in the neighboring detection section (Table 1). In addition, the deep feature space was used to detect the neighbor inside the parenthesis. In other words, DNPT(conv3) indicated that the proposed transfer method could be employed by the conv3 feature to detect the neighbors. DNPT(oracle) represents the tests in which the location of the main components was used instead of estimating the location of the components. If a method was available for the accurate estimation of the location of the components, the mean accuracy of the DNPT(oracle) could be obtained as the method provides the upper bound of the mean accuracy for the enhanced transfer method.

Table 1: Results of Non-parametric Transfer Method (Names specified with * represent the proposed methods)

	Training		Test		
Name	**Peripheral Rectangle**	**Location Of Objects**	**Peripheral Rectangle**	**Location Of Objects**	**Mean Classification Accuracy**
[30]NPT	√	√	√		57.84
*DNPT(fc7)	√	√	√		56.52
*DNPT(conv3)	√	√	√		67.89
*DNPT(pool5)	√	√	√		68.56
*DNPT(oracle)	√	√	√	√	78.08

Two states could be considered in the comparison of the proposed method with the results of the fine-grained classification; one is DeepRF, in which it is assumed that the bounding box is available to the test images and only requires the proposed method to estimate the peripheral rectangle of the individual's mouth and tongue. Another is DeepRF(AII), which requires the estimation of the peripheral rectangle of the face in addition to the

peripheral rectangle of the mouth and tongue. Since the random forest was employed in this method and was inherently random, the tests were performed in triplicate, and the mean accuracy was reported.

According to the information in Table 2, the proposed method (DeepRF) could achieve the mean accuracy of 73.78, which is comparable to the knowledge boundary method with the mean accuracy of 76.37. Furthermore, the proposed method of DeepRF(II) could achieve the mean accuracy of 72.02, while the knowledge boundary method also had the mean accuracy of 73.89 in case of an unavailable peripheral rectangle.

Table 2: Accuracy of Proposed Method (specified with *) Compared to Knowledge Boundary Method

	Training		Test		
Name	**Peripheral Rectangle**	**Location of Objects**	**Peripheral Rectangle**	**Location of Objects**	**Mean Classification Accuracy**
[9]PRCNN	√	√	√		76.37
*DeepRF	√	√	√		73.78 (±0.32)
[9]PRCNN	√	√			73.89
*DeepRF(All)	√	√			72.02 (±0.33)

5- Conclusion

In this research, we studied the granular classification problem and its importance along with the methods available for the early detection of stomach cancer. To this end, deep convolutional neural networks were used for finding the neighbors and extracting features. Using this network has, to a certain degree, led to the improvement of the average classification accuracy on the dataset of tongue images. Also, to eliminate the lack of generalization problem of the deep convolutional network, we have used a random forest and deep pixel features. Using this method has led to high speed and simplicity compared to other methods in addition to acceptable results. The results show a 73.78 accuracy for the proposed method, which is comparable to the cutting-edge 76.37 average accuracy. Also, the proposed method called DeepRF (All) reaches an average accuracy of 72.02. The state-of-the-art method also reaches an average accuracy of 73.89 when bounding boxes are not available during testing.

References

[1] B. R. Bistrian, "Modern Nutrition in Health and Disease (Tenth Edition)," Crit. Care Med., vol. 34, no. 9, p. 2514, Sep. 2006, doi: 10.1097/01.CCM.0000236502.51400.9F.

[2] R. A. Smith et al., "American Cancer Society Guidelines for the Early Detection of Cancer," CA. Cancer J. Clin., vol. 52, no. 1, pp. 8–22, Jan. 2002, doi: 10.3322/canjclin.52.1.8.

[3] G. Murphy, R. Pfeiffer, M. C. Camargo, and C. S. Rabkin, "Meta-analysis Shows That Prevalence of Epstein–Barr Virus-Positive Gastric Cancer Differs Based on Sex and Anatomic Location," Gastroenterology, vol. 137, no. 3, pp. 824–833, 2009, doi: https://doi.org/10.1053/j.gastro.2009.05.001.

[4] H. H. & J. M. Azizi F, "Epidemiology and control of common diseases in Iran," 3th ed. Tehran Khosravi Publ., pp. 45–7, 2010.

[5] P. Bertuccio et al., "Recent patterns in gastric cancer: A global overview," International Journal of Cancer, vol. 125, no. 3. Wiley-Liss Inc., pp. 666–673, Aug. 01, 2009, doi: 10.1002/ijc.24290.

[6] A. Baranovsky and M. H. Myers, "Cancer Incidence and Survival in Patients 65 Years of Age and Older," CA. Cancer J. Clin., vol. 36, no. 1, pp. 26–41, Jan. 1986, doi: 10.3322/canjclin.36.1.26.

[7] J. D. Emerson and G. A. Colditz, " Use of Statistical Analysis in The New England Journal of Medicine ," N. Engl. J. Med., vol. 309, no. 12, pp. 709–713, Sep. 1983, doi: 10.1056/nejm198309223091206.

[8] et al. Hasanzadeh, J., "No TitleGender differences in esophagus, stomach, colon and rectum cancers in fars, Iran, during 2009-2010: an epidemiological population based study," J. Rafsanjan Univ. Med. Sci., vol. 12.5, pp. 333-342., 2013.

[9] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-Based R-CNNs for Fine-Grained Category Detection BT - Computer Vision – ECCV 2014," 2014, pp. 834–849.

[10] H. H. Hartgrink, E. P. M. Jansen, N. C. T. van Grieken, and C. J. H. van de Velde, "Gastric cancer," Lancet, vol. 374, no. 9688, pp. 477–490, 2009, doi: https://doi.org/10.1016/S0140-6736(09)60617-6.

[11] S. Sarebanha, A. Hooman Kazemi, P. Sadrolsadat, and N. Xin, "75 Comparison of Traditional Chinese Medicine and Traditional Iranian Medicine in Diagnostic Aspect," Mar. 2016. Accessed: Nov. 19, 2020. [Online]. Available: http://jtim.tums.ac.ir.

[12] J. Hu, S. Han, Y. Chen, and Z. Ji, "Variations of Tongue Coating Microbiota in Patients with Gastric Cancer," Biomed Res. Int., vol. 2015, 2015, doi: 10.1155/2015/173729.

[13] T. Ma, C. Tan, H. Zhang, M. Wang, W. Ding, and S. Li, "Bridging the gap between traditional Chinese medicine and systems biology: The connection of Cold Syndrome and NEI network," Molecular BioSystems, vol. 6, no. 4. The Royal Society of Chemistry, pp. 613–619, Mar. 17, 2010, doi: 10.1039/b914024g.

[14] X. Liu et al., "The Metabonomic Studies of Tongue Coating in H. pylori Positive Chronic Gastritis Patients," Evidence-Based Complement. Altern. Med., vol. 2015, p. 804085, 2015, doi: 10.1155/2015/804085.

[15] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[16] X. Liu et al., "The Metabonomic Studies of Tongue Coating in H. pylori Positive Chronic Gastritis Patients," Evidence-based Complement. Altern. Med., vol. 2015, 2015, doi: 10.1155/2015/804085.

[17] C.-C. Chiu, "A novel approach based on computerized image analysis for traditional Chinese medical diagnosis of the tongue," Comput. Methods Programs Biomed., vol. 61, no. 2, pp. 77–89, 2000, doi: https://doi.org/10.1016/S0169-2607(99)00031-0.

[18] S. Branson et al., "Visual Recognition with Humans in the Loop BT - Computer Vision – ECCV 2010," 2010, pp. 438–451.

[19] M. Moghimi, "Using color for object recognition," Calif. Inst. Technol. Tech. Rep, 2011.

[20] E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars, "Fine-Grained Categorization by Alignments," pp. 1713–1720, 2013, doi: 10.1109/ICCV.2013.215.

[21] X. Zhang, H. Xiong, W. Zhou, and Q. Tian, "Fused One-vs-All Features with Semantic Alignments for Fine-Grained Visual Categorization," IEEE Trans. Image Process., vol. 25, no. 2, pp. 878–892, Feb. 2016, doi: 10.1109/TIP.2015.2509425.

[22] B. Yao, A. Khosla, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2011, pp. 1577–1584, doi: 10.1109/CVPR.2011.5995368.

[23] B. Yao, G. Bradski, and L. Fei-Fei, "A codebook-free and annotation-free approach for fine-grained image categorization," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012, pp. 3466–3473, doi: 10.1109/CVPR.2012.6248088.

[24] N. Zhang, R. Farrell, and T. Darrell, "Pose pooling kernels for sub-category recognition," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012, pp. 3665–3672, doi: 10.1109/CVPR.2012.6248364.

[25] Y. J. Lee, A. A. Efros, and M. Hebert, "Style-aware mid-level representation for discovering visual connections in space and time," in Proceedings of the IEEE international conference on computer vision, 2013, pp. 1857–1864.

[26] T. Berg and P. N. Belhumeur, "Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 955–962.

[27] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2014, pp. 806–813.

[28] N. Zhang, R. Farrell, F. Iandola, and T. Darrell, "Deformable part descriptors for fine-grained recognition and attribute prediction," in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 729–736.

[29] J. Donahue et al., "Decaf: A deep convolutional activation feature for generic visual recognition," in International conference on machine learning, 2014, pp. 647–655.

Elham Gholami received the B.Sc. degree in Computer Engineering from Ferdowsi University, mashhad, Iran in 1999, and M.Sc. degree in Software Engineering from Islamic Azad University, Mashhad branch, Iran 2005. Currently she is PhD Candidate in Azad University, Neyshabur Branch, Iran. Her research interests include Information retrieval, Recommendation systems, Human-computer interaction, Database Systems, Web mining and Data mining.

Seyed Reza Kamel Tabbakh is Assistant Professor in Department of Computer Engineering, Faculty of Engineering, Islamic Azad University, Mashhad branch, Iran. He received his B.Sc. degree in Software Engineering from Islamic Azad University, Mashhad branch, Iran 1999, his M.Sc. degree in Software Engineering from Islamic Azad University, South Tehran branch, Iran 2001, and his PhD in Communication and Network Engineering from Universiti Putra Malaysia (UPM), in 2011. He has several publications in national and international journals and conferences. His research interests include Internet of Things and IPv6 networks. He is an IEEE member.

Maryam Kheirabadi is Assistant Professor in Department of Computer Engineering, Faculty of Engineering, Islamic Azad University, Neyshabur branch, Iran. she received B.Sc. degree in Software Engineering from Islamic Azad University, Mashhad branch, Iran,2002. her M.s.c degree in Business Information Systems, Royal Holloway, London, UK 2005 and her PhD degree in Computer Science from Kingston University, London, UK 2012. My specialty is on Software Engineering, also worked and researched on Cloud Computing, Data Mining and etc.

Cluster-based Coverage Scheme for Wireless Sensor Networks using Learning Automata

Seyyed Keyvan Mousavi
Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran
mosavikeyvan90@gmail.com
Ali Ghaffari*
Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran
A.ghaffari@iaut.ac.ir

Received: 27/Dec/2020 Revised: 11/May/2021 Accepted: 25/May/2021

Abstract
Network coverage is one of the most important challenges in wireless sensor networks (WSNs). In a WSN, each sensor node has a sensing area coverage based on its sensing range. In most applications, sensor nodes are randomly deployed in the environment which causes the density of nodes become high in some areas and low in some other. In this case, some areas are not covered by none of sensor nodes which these areas are called coverage holes. Also, creating areas with high density leads to redundant overlapping and as a result the network lifetime decreases. In this paper, a cluster-based scheme for the coverage problem of WSNs using learning automata is proposed. In the proposed scheme, each node creates the action and probability vectors of learning automata for itself and its neighbors, then determines the status of itself and all its neighbors and finally sends them to the cluster head (CH). Afterward, each CH starts to reward or penalize the vectors and sends the results to the sender for updating purposes. Thereafter, among the sent vectors, the CH node selects the best action vector and broadcasts it in the form of a message inside the cluster. Finally, each member changes its status in accordance with the vector included in the received message from the corresponding CH and the active sensor nodes perform environment monitoring operations. The simulation results show that the proposed scheme improves the network coverage and the energy consumption.

Keywords: Wireless sensor networks; Clustering; Learning automata; Network coverage.

1- Introduction

Wireless sensor networks are consist of a large number of sensor nodes which are densely deployed inside a phenomenon or very close to it [1-6]. One of the basic challenges in wireless sensor networks is the coverage problem [7, 8]. Coverage favors the placement of sensors in the environment and determines how much the environment is monitored by the sensors. In WSNs, sensor nodes are usually distributed randomly in the environment which causes the density of nodes become high in some areas and low in some other [9, 10]. The redundancy of sensor nodes in an area, firstly leads to waste the energy of some sensor nodes which causes to reduce the network lifetime and secondly, leads to overlap that area with a high probability while some areas may remain out of coverage [11-14]. Hence, in these networks in order to reduce the amount of overlap of sensor nodes and optimal coverage of the network and also reducing the energy consumption and prolonging the network lifetime, identifying the redundant nodes seems to be an essential problem [15-18].

Three main categories of coverage related problems can be identified into three classes: (*i*) target coverage, (*ii*) barrier coverage, and (*iii*) area coverage. In target coverage class, we try to monitor various targets with sensor nodes in target tracking applications. In barrier coverage, we try to minimize the probability of undetected penetration through a sensor node. On the other hand, area coverage can be divided into two groups: partially coverage and full coverage.

In partially coverage, a determined percent of monitored area can be covered with distributed sensor nodes. While in full coverage class we must continuously monitor whole interested area [19, 20].

This paper proposes a new method based on learning automata to improve the coverage and to prolong the lifetime of WSNs. The proposed scheme reduces the

* Corresponding Author

amount of energy consumption in the network and as a result increases the network lifetime by identifying and disabling the redundant nodes. The simulation result shows that proposed method improves the network performance metrics such as coverage and energy consumption.

Briefly, the main contribution of this paper are as follows:

- Discussing the various related works about WSNs coverage optimization strategies.
- Proposing a new coverage optimization mechanism in WSNs using learning automata.
- Evaluation and validation of the effectiveness of proposed mechanism in MATLAB simulator under varying sensor nodes.

The rest of the paper is organized as follows. Section 2 indicates related works. Section 3 describes the proposed scheme. Section 4 describes the performance evaluation of the proposed scheme and finally, Section 5 provides the conclusion and future works.

2- Related Works

This section provides an overall picture of the literature related to the coverage problems in WSNs.

In [21], a method based on target coverage in heterogeneous wireless sensor networks has been proposed. This method is one of the connected k-coverage methods. In addition, each node in the coverage set can communicate with the sink either directly or via a multi-hop connection through its neighbors. In this method, neighbor nodes can be used as a relay to establish connection between nodes in the network. The disadvantages of this method are high energy consumption and as a result low network lifetime.

In [22], a method called coverage and energy strategy has been proposed for WSNs. In this method, each sensor node has four modes including INITIAL, WORKING, SLEEPING and CHECKING. This algorithm includes two phases. In the first phase, all nodes are initially in WORKING mode. In WORKING mode, firstly each node exchanges its location information with its neighbors and then estimates the coverage share of itself with the neighbor nodes which are in WORKING mode. If the coverage share is independent or the neighbor node which is in WORKING mode is a redundant node, the node goes to SLEEPING mode otherwise it stays in WORKING mode. In the second phase, the nodes try to exchange their duties between SLEEPING and WORKING modes based on the coverage and connectivity conditions. In SLEEPING mode, the nodes will periodically wake up and switch to CHECKING mode. A node which is in WORKING mode goes to SLEEPING mode if its neighbor node in CHECKING mode, has a greater level of residual energy. The large number of active nodes in this method leads to waste the energy and also redundancy in the coverage. In [23], an algorithm called Imprecise Detections Algorithm (IDA) has been proposed to improve coverage in wireless sensor networks. In this algorithm, firstly the environment is modeled as a grid and the sensor nodes are located in the grid points. Suppose the probability of detecting a target by a node has an exponential relation with the distance - between the target and the sensor node. This means a target located at distance d from a sensor node is detected by that node with the probability of $e^{-\alpha d}$. The algorithm presented in [24] by using of Voronoi diagram, tries to select the most suitable nodes for complete coverage of the environment. One of the factors that has been considered in this algorithm for selecting active nodes is the residual energy of nodes. This algorithm tries to increase the network lifetime by balancing the energy consumption of entire network. In this algorithm, all nodes are initially in the sleep mode. Then, one of the sensor nodes is selected as the starting node and creates a Voronoi cell for other sensors.

In [25], a protocol called Coverage-Preserving Clustering Protocol (CPCP) has been presented for wireless sensor networks. To ensure balanced energy consumption among CHs in the network, most of existing clustering protocols favor uniformly distribution of clusters with stable average cluster sizes and obtaining the best cluster distribution number per a unit of time is the main goal of clustered wireless sensor networks. In this method, CHS are distributed uniformly entire the network by limiting the maximum cluster area. Thus, clusters in a sparse area are formed as well as clusters in a dense area which prevents the costly transfer by the costly nodes to the remote CHS. In [26], the authors proposed PCLA (Partial Coverage with Learning Automata), a novel algorithm that relies on learning automata to implement sleep scheduling approaches. It minimizes the number of sensors to activate for covering a desired portion of the region of interest preserving the connectivity among sensors. In [27], the authors proposed an integrated and energy-efficient protocol for Coverage, Connectivity, and Communication (C3) in WSNs. This protocol uses RSSI (received signal strength indicator) to divide the network into virtual rings, defines clusters with CH at alternating rings, and defines dings that are rings inside a cluster. This scheme uses triangular tessellation to identify redundant nodes, and sends gathered data to sink through CHs. In C3, the distance of CHs from the sink node is too long. This weakness makes the CHs apply more intermediate nodes to send their data directly to the sink node and consequently leads to more energy consumption in the network.

In [28], the authors proposed a novel algorithm called complex alliance strategy with multi-objective

optimization of coverage (CASMOC) which could improve node coverage effectively. CASMOC achieves the energy balance of whole network through giving the proportional relationship of the energy conversion function between the working node and its neighbors, and applies this relationship in scheduling low energy mobile nodes. In [29], the authors presented a distributed protocol, called single phase multiple initiator (SPMI). Based on determining a connected dominating set (*CDS*), SPMI connects cover set for assuring the coverage and connectivity in *WSN*. Without using sensors' location information, SPMI only requires a single phase to construct a *CDS* in distributed manner.

In [30], the authors proposed a cover set to find the minimum set of sensors that completely cover the sensing ranges within an interest area as a criterion by selecting an optimal number of active sensors considering residual energy and the cover set and to keep alive the important sensors for the sensing coverage task. Also, the authors proposed an area coverage-aware clustering protocol (ACACP) with optimum energy consumption considering the activation node, network clustering, and multi-hop communication to improve overall network lifetime while preserving coverage.

3- The Proposed Method

To address the mentioned problem of C3, in the proposed method the sink node plays its main role and acts as an interface between user and the network. Due to this reason, in the proposed method the probability of being CH for the nodes that are closer to the sink will be greater than C3 [27] method. This will lead to reduce the distance from the sink that makes the CHs to send their data directly to the sink apply fewer intermediate node and consequently less energy is consumed in the network. In C3 [27] method, the energy of sensor nodes is not considered to select the redundant nodes. Hence in the proposed method, the energy of sensor nodes is also considered to select the redundant nodes. Another weakness of C3 [27] method is that the selection process of redundant nodes is done greedily. Though this method leads to increase the amount of coverage at the beginning of network lifetime, but with the passage of time, the algorithm will converge to the local optimal. While in the proposed method, the action of selecting the redundant nodes is performed by the use of intelligent learning automata algorithm that causes the proposed method converge to the optimal solution as time goes by and doesn't trap in a local optimal. In the proposed method, each sensor node can be in two different modes, including: Active mode and sleep mode.

In the proposed method, it is assumed that all the sensors deployed in the environment have the same sensing area and initial energy and each sensor node is aware of its geographic location by using global positioning system (GPS) or other positioning devices of the network. Also, it is assumed that all sensor nodes are initially in the active mode. After deploying the sensors in the environment, the sensor nodes by broadcasting the Hello packets, identify all its neighbors and send their status information to these neighbor nodes. The proposed algorithm consists of three phases in each round. The first phase includes network clustering, formation of the action set vector and the action probability vector. In the second phase, the coverage set within each cluster is created by the CH node using learning automata. Finally, in the third phase, the coverage set begin to monitor the environment. The details of each of these phases are explained below.

3-1 First Phase: Network Clustering, Formation of Action Set Vector and Action Probability Vector

In this phase, the following three types of messages can be used:

i) Status Message: by using this message, each sensor node sends the information about its status such as neighborhood degree, overlap degree and residual energy in order to become a CH to all its neighbors.

ii) CH Message: The candidate node to become a CH, by using this message announces itself as a CH along with the latest information about its status to all its neighbor nodes.

iii) Membership Message: Each sensor node by using this message, declares itself as a single-hop member of a certain CH.

In this phase, it is tried to select the CHs among the nodes located in the densely areas of network and with the highest overlap degree. The degree of a node, is referred to the number of single-hop neighbors of that node in the network and the overlap degree is referred to the number of neighbors that overlap with this node at the certain points. To calculate the degree of a node, after network setup and deployment of the sensors in a given environment, the sink node sends a setup message to all nodes in the network. Each sensor node upon receiving this message, at a random time in the interval between zero and T_{max}, transmits its information to all its neighbors. Each neighbor node upon receiving this message and by checking the content of the message, can determine its degree. The reason of using a random time in the range between zero and T_{max} is to reduce collision while sending the HELLO packets. To determine the overlap degree of a node with its neighbors, firstly with considering the fixed points in the sensing area of the sensor nodes, the area coverage problem is converted to

the target coverage problem and then by using Eq. (1), the overlap degree of nodes is calculated:

$$OD_i = \frac{\sum CT_i}{\sum NT_i} \quad (1)$$

In Eq. (1), ΣCT_i indicates the total number of times that the target points existing in the sensing area of node i are covered by its neighbors and ΣNT_i indicates the sum of the target points existing in the sensing area of node *i*. In figures 1 and 2 the hypothetical sensor nodes A and B have the same number of neighbors, but their overlap degrees are different from each other. The neighborhood degrees of both nodes A and B are equal to 4, but since the target points of node A have been covered 6 times by its neighboring nodes and the target points of node B have been covered 4 times by its neighbors.

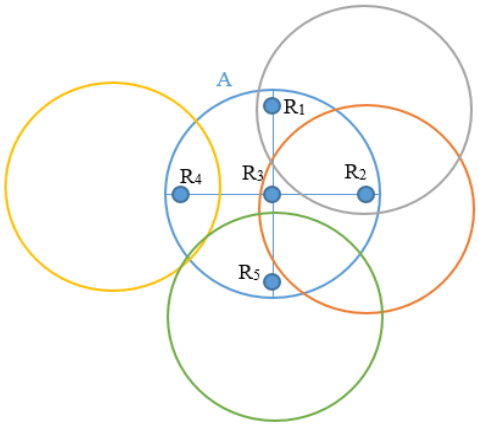


Fig. 1. The status of the target coverage of neighbors of the hypothetical node A

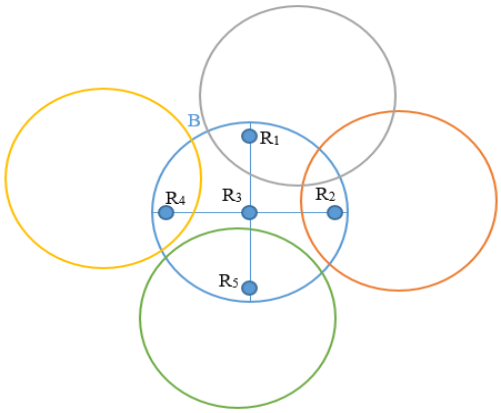


Fig. 2. The status of the target coverage of neighbors of the hypothetical node B

After determining the neighborhood degree and the overlap degree of sensor nodes, in the next step each node after sending its status to all its neighbors, waits for a certain time period T_{ni} until it receives a CH message from the nearest candidate CH. If within this time period it doesn't receive any message, announces itself as a CH and broadcasts a CH message along with the information about its status throughout the network. The waiting time T_{ni} of each node is calculated using Eq. (2) [31]:

$$T_{ni} = \alpha . e^{\frac{1}{d(ni)+od(ni)}} \quad (2)$$

In Eq. (3), $d(n_i)$ shows the neighborhood degree or the number of neighbors of node n_i and $od(n_i)$ indicates the overlap degree of node n_i. Also, the constant factor α should be set such that the condition $0<T_{ni}\leq T_{max}$ is satisfied.

To minimize the number of clusters and select the best node in that area as CH, the residual energy of nodes is also considered. In this case Eq. (2) turns into Eq. (3) and the node having the highest residual energy is chosen as CH. As a result, instead of having multiple clusters in an area, we will have a single cluster.

$$T_{ni} = \alpha . e^{\frac{1}{d(n_i)+od(n_i)} + \frac{E_{res}(n_i)}{E_{ini}}} \quad (3)$$

where in Eq. (4), E_{ini} and $E_{res}(n_i)$ indicate the initial and the residual energy of node n_i. In the next phase, each sensor node upon receiving the CH message from the nearest candidate node, declares itself as a member of that CH by sending a message containing its information. The CH upon receiving this message, stores its member node's information in its database and identifies this node as a single-hop member of itself.

After selecting the cluster heads and clustering the network, the CHs determine a certain time interval for each of its members to send their information so that each member can send its information to the corresponding cluster only within this time interval. This determined time interval to send the information by each cluster member is announced through sending a message by the cluster head node. With scheduling the member nodes by the cluster heads, it is prevented any collision of information while sending data to the cluster head. In order to balance the network load, at the beginning of each period, the clustering process is repeated again. After selection of the cluster heads and clustering the network, firstly inside the clusters each node creates the action vector and the probability vector of learning automata for itself and its neighbors. Then each node according to the probability vector of learning automata, determines the state of itself and its neighbors randomly and sends it as a message to its corresponding cluster head. The action vector of automata of a node includes the status of that node and its neighbors and is a vector as VA= {a, s}. In this vector, "a" indicates the active mode and "s" shows the sleep mode of a node. The probability vector of a node is equal to 0.5 initially and is a vector as (0.5, 0.5). In figure 4, the six nodes named A, B, C, D, E

and F are the members of a cluster and the node CH is the cluster head. Also, nodes B, C, D, E and F are the one-hop neighbors of node A.

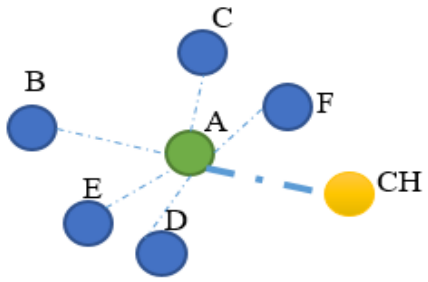


Fig. 3. The hypothetical node A and its single-hop neighbors inside the cluster

In figure 4, the node A firstly creates the action vector and the probability vector of itself and its neighbors as Table 1. Then as Table 1, according to the probability vector of itself and its neighbors, randomly selects action "a" or "s" for itself or its single-hop neighbors and determines the state of itself and its neighbors accordingly. Finally, as Table 1, sends the final status of itself and its neighbors to the cluster head as a message. The process of selecting the automata action by a node for itself and its neighbors is as follows: firstly, the node generates a random number between 0 and 1 for its intended node and then compares this number with the range of existing numbers in the probability vector related to each action of its intended node. Afterward, it considers the action that the generated number is within its probability range as the state of the node. For example, in figure 4, the node A to select the state of node B, firstly generates a random number as 0.6. Then, according to the probability vector of node B where the selection range of action "a" is (0, 0.5) and the selection range of action "s" is (0.5, 1), determines the placement range of the random number 0.6. Then, since the random number 0.6 is within range (0.5, 1), thus node A selects action "s" as the status of node B. The node A continues this process until it has not set the status of itself and all its neighbors.

Table 1. The action and probability vectors of the generated automata by node A for itself and its neighbors

Node	Action vector	Probability vector
A	{a, s}	{0.5, 0.5}
B	{a, s}	{0.5, 0.5}
C	{a, s}	{0.5, 0.5}
D	{a, s}	{0.5, 0.5}
E	{a, s}	{0.5, 0.5}
F	{a, s}	{0.5, 0.5}

Table 2. The Selected action by node A for itself and its neighbors

node	Generated number	Selected action
A	0.4	a
B	0.6	s
C	0.3	a
D	0.7	s
E	0.1	a
F	0.6	s

Table 3. The transmitted message from node A to the cluster head

node	Action vector	Probability vector	Selected action
A	{a, s}	{0.5, 0.5}	a
B	{a, s}	{0.5, 0.5}	s
C	{a, s}	{0.5, 0.5}	a
D	{a, s}	{0.5, 0.5}	s
E	{a, s}	{0.5, 0.5}	a
F	{a, s}	{0.5, 0.5}	s

3-2 Second Phase: Formation of Coverage Set

In this phase, first of all, the cluster head according to the information received from its cluster members, starts to reward or penalize the received messages by the use of the relations of learning automata [7] and then transmits the results to the sender of each received message. Afterward, among the transmitted vectors to its cluster members, it selects the best vector of actions and broadcasts the final status of nodes to its members in the form of a message for optimal covering the cluster. For example, suppose that the cluster head CH has received Table 3 as a message from node A. Among the one-hop members of the cluster, the nodes having the highest overlap degree are considered as the redundant nodes and should go into the sleep mode. To achieve this goal, the cluster head according to Table 3, firstly creates Table 4 based on the overlap degree and the residual energy of nodes and to calculates the value of *un* field of each node by using Eq. (4):

$$un_i = \beta \times \frac{od(ni)}{Max(od)} + (1-\beta) \times \frac{E_{res}(n_i)}{E_{res}(ni)} \quad (4)$$

In *Eq.* (5), *Max(od)* indicates highest overlap degree among the nodes included in the forwarded message to the CH, $E_{res}(n_i)$ indicates the minimum residual energy among the nodes, $E_{res}(n_i)$ indicates the residual energy of node n_i and β is a constant number between 0 and 1. In this paper, β is set to 0.65.

Table 4. Details of the existing sensor nodes within the received message by the cluster head

Node name	Status	Residual energy	Overlapping degree	Amount of *un*
A	a	1.3 j	1.2	0.7010
B	s	1.5 j	1.3	0.7088
C	a	1.6 j	1.4	0.7008
D	s	1.5 j	0.8	0.9588
E	a	1.5 j	1.1	0.7816
F	s	1.7 j	1.2	0.7833

Afterward, the cluster head sorts Table 4 based on the un values of nodes in descending order from the highest value to the lowest value and by using Eqs. (5) to (8), learning automata [7] starts to reward or penalize the nodes and finally converts Table 4 into Table 5. Ideally, the nodes having the highest *un* values should go into the sleep mode, because they have the highest overlap degree and the lowest residual energy. Accordingly, if the sleep mode is selected for a node while the node with a higher *un* value is in the active mode, this action is penalized by the cluster head via Eqs. (7) and (8), otherwise it is rewarded using *Eqs*. (5) and (6).

$$P_i(n+1) = P_i(n) + a \times (1 - P_i(n)) \quad (5)$$

$$P_j(n+1) = (1-a) \times P_j(n+1) \quad \forall_{j,j \neq i} \quad (6)$$

$$P_i(n+1) = (1-b) \times P_i(n) \quad (7)$$

$$P_j(n+1) = \left(\frac{b}{r-1}\right) + (1-b) \times P_i(n) \forall_{j,j \neq i} \quad (8)$$

In relations (5) and (7), $P_i(n)$ indicates the probability of i^{th} action in the n^{th} step and $P_i(n+1)$ indicates the probability of i^{th} action in the $(n+1)^{th}$ step. In Eqs. (5) and (6), a shows the coefficient of reward and in relations (6) and (8), b shows the coefficient of penalty. In relations (7) and (8), $P_j(n)$ indicates the probability of j^{th} action in the nth step and $P_j(n+1)$ indicates the probability of j^{th} action in the $(n+1)^{th}$ step. Furthermore in Eq. (8), r shows the number of automata actions.

Table 5. The sorted version of table 4 based on *un* values

Hostname	Status	Amount of *un*	Reward / penalty
D	s	0.9588	reward
F	s	0.7833	reward
E	a	0.7816	penalty
B	s	0.7088	reward
A	a	0.7010	reward
C	a	0.7008	reward

As an example, in Table 5, to assign a reward or penalty to the actions selected by node A, the cluster head node starts from the node with the highest value of *un*. If the selected node is in status "a" and the status of the next neighbor node that has a fewer *un* value is "s", thus that action will be penalized. If the selected node is in status "s" and the status of the next neighbor node that has a fewer *un* value is "a", hence that action will be rewarded. If the next node is in the same status as the selected node, thus it will be considered a reward for that action. In Table 5, the node D has the highest value of *un* and thus the allocation of reward or penalty to the actions selected by node A starts from this node. Since node D is in status "s" and its adjacent node in the table (i.e., node F) is also in status "s", then it is rewarded the actions selected by node D using *Eqs*. (6) and (7). Similarly, since node E is in status "a" and its adjacent node in the table (i.e., node B) is in status "s", then it is penalized the actions selected by node D using *Eqs*. (8) and (9). The last node in the table that has generated less number by using relation (5), according to the tests carried out, it is better to be activated. In other words, if the last node in the table is in status "a", then it is rewarded the selected action using *Eqs*. (6) and (7), otherwise this action is penalized by *Eqs*. (8) and (9). In Table 5, since node C is in status "a", it is rewarded the selected action of node C by *Eqs*. (6) and (7). After the allocation of reward and penalty to the vectors within the received messages from the members, the cluster head firstly sends each message to the sender of that message so that the sender node update its vector accordingly. Then among these messages, it selects the message that leads to a high reward and broadcasts the final status of nodes to its members in the form of a message for optimal covering the cluster. Each member node after receiving this message from the cluster head, changes its status based on the vector included in the received message. With the passage of time and network lifetime, the parameters of nodes may be changed. For example, the node C which has the least amount of *un* in Table 6, may achieve a better un value than its neighbors in the next clustering round. Also, at the beginning of the formation of clusters, this action of automata to form the coverage set in the second phase can be repeated for n times.

Table 6. Final status of the nodes

Hostname	Status
D	s
F	s
E	a
B	s
A	a
C	a

3-3- Third Phase: Monitoring

After selecting the active nodes in the previous phase, in this phase the active nodes start to perform monitoring operations in their sensing areas until the next clustering round is began or the lifetime of cluster head node is terminated. Figure 5, shows the flowchart of the proposed method.

Algorithm 1 shows the pseudo-code of the proposed method.

Algorithm 1. **Pseudo code of the proposed scheme**

1. **Start**
2. Distribute sensor nodes randomly.
3. Each node waits for a certain time period T_n which is calculated via relation (3).
4. **If** node receives a message within T_n time period
5. **Then**
6. Join a cluster.
7. **Else**
8. Announce itself as a cluster head.
9. Start the coverage algorithm.
10. **Do** for each node
11. **If** node has an automata action vector for its neighbors
12. **Then**
13. Randomly select an action for each neighbor.
14. **Else**
15. Create the action vector for each neighbor.
16. Randomly select an action for each neighbor.
17. **End If**
18. Send these selected actions as a message to the CH.
19. **End Do**
20. **Do** these operations for each CH
21. After receiving the message, CH does as follow:
22. Sort the message by using relation (4).
23. Assign reward or penalty by considering the values obtained from relation (4) and nodes status.
24. Announce the status of the performed reward/penalty
25. Broadcast the message having the maximum reward as the best message.
26. **End Do**
27. Do monitoring.
28. **If** network lifetime has over
29. **Then**
30. End.
31. **Else**
32. **If** cluster lifetime has over or cluster head is dead
33. **Then**
34. Go step 4.
35. **Else**
36. Go step 25.
37. **End If**
38. **End If**
39. **End**

3-4- Energy Consumption Model

If we assume that *l* is the number of transmitted bits, d is the distance between transmitter and receiver, E_{elec} is the energy required for transmission of a bit, ε_{amp} is the energy required to relay the transmitted bits and d_0 is the transmitting distance threshold for amplifier circuit, then the energy required for sending *l* bits of data is calculated according to *Eq*. (9):

$$E_{Tx}(K,d) = \begin{cases} k\left(E_{elec} + \varepsilon_{fs} d^2\right) & d < d_0 \\ k\left(E_{elec} + \varepsilon_{amp} d^4\right) & d \geq d_0 \end{cases} \quad (9)$$

In *Eq*. (9), the amount of d_0 is calculated via *Eq*. (10):

$$d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{amp}}} \quad (10)$$

The energy required for receiving the packets of size *l* bits is also calculated by *Eq*. (11):

$$E_{Rx}(K,d) = k.E_{elec} \quad (11)$$

4- Performance Evaluation

In this section, the performance of the proposed method is compared with that of C3 method in terms of the network coverage percentage and the average residual energy of sensor nodes in an environment of size 160×160 m^2 and with a uniform distribution of 100 to 300 sensor nodes. Both proposed and C3 [27] methods have been implemented via MATLAB simulation software. All simulations have been repeated for 30 independent runs. The simulation parameters are given in Table 7.

Table 7. Simulation parameters

Parameter	Value
Number of sensors	100 to 300
Sensing radius	10 m
Sending and receiving radius	20 m
Network area	160 × 160 m^2
Number of running	30
Position of the sink node	(80,80), (35,25), (130,65)
α, β	0.18, 0.55
E_{elec}	50 nj/bit
E_{DA}	5 nj/bit/packet
ε_{Fs}	10 pj/bit/m^2
ε_{amp}	0.0013 Pj/bit/m^4
Initial energy	2 j
Packet size	4000 bits
Control packet size	200 bits

4-1 Comparison in Term of Network Coverage Percentage

In this section, the performance of the proposed method is compared with C3 [27] method in term of the network coverage percentage under the following three scenarios.

- **First Scenario:** sink is located in coordinate (80, 80) of the environment. Based on figure 7, with the increasing the number of nodes, the network coverage percentage generally increases, but the rate of increase in the proposed method is greater than C3 [27] method.

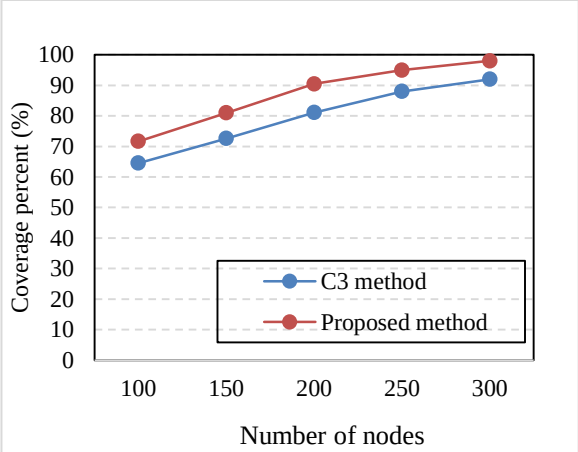


Fig. 5. Network coverage percentage in the first scenario

- **Second Scenario**: sink is located in coordinate (35, 25) of the environment. Based on figure 8, by increasing the number of nodes, the network coverage percentage also increases, but the rate of increase in the proposed method is more than C3 [27] method.

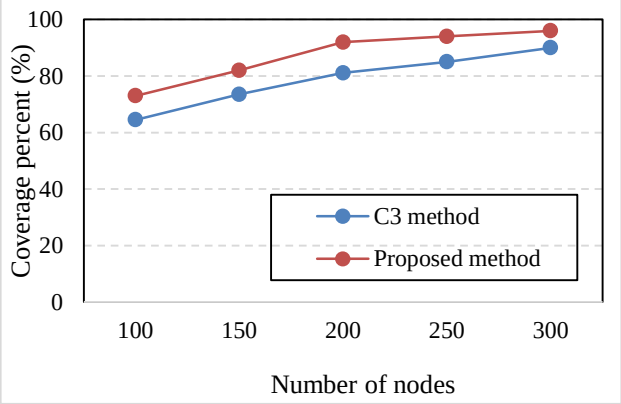


Fig. 6. Network coverage percentage in the second scenario

- **Third Scenario:** sink is located in coordinate (130, 65) of the environment. As shown in figure 9, with the increasing the number of nodes, the network coverage percentage increases, but the rate of increase in the proposed method is more than C3 [27] method. The simulation results of the three scenarios show that the proposed method performs better than C3 [27] method in term of the network coverage percentage.

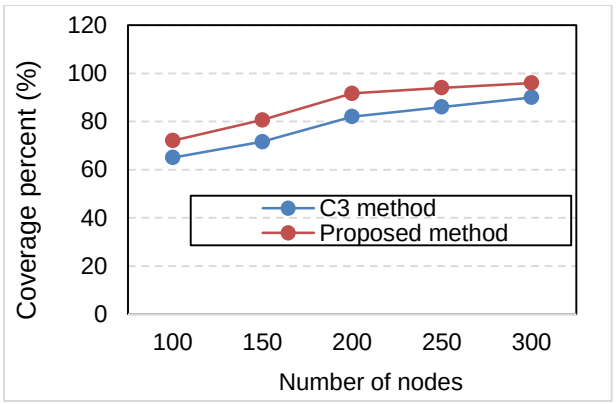


Fig. 7. Network coverage percentage in the third scenario

4-2 Comparing in Term of Average Residual Energy of Nodes

In this section, the efficiency of the proposed method is compared against C3 [27] method in term of the average residual energy of sensor nodes.

- **First Scenario**: sink is located in coordinate (80, 80) of the environment

According to figure 10, we can conclude that with the increasing the number of nodes, the average residual energy of network nodes generally increases, but the rate of increase in the proposed method is higher than C3 [27] method.

- **Second Scenario***:* sink is located in coordinate (35, 25) of the environment. Based on figure 11, we can conclude that by increasing the number of nodes, the average residual energy of network nodes increases, but the rate of increase in the proposed method is more than C3 method.

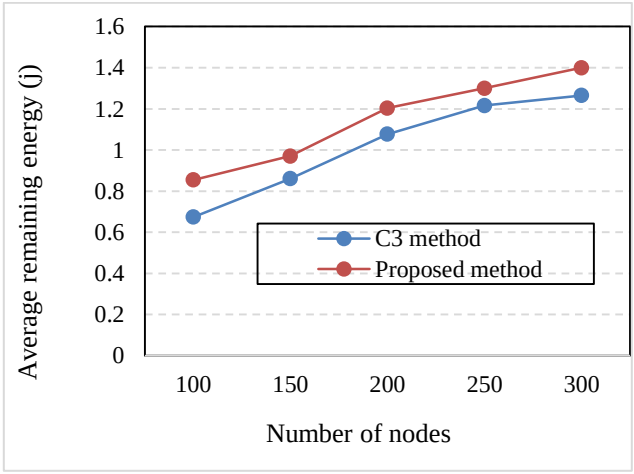


Fig. 8. Average residual energy of nodes in the first scenario

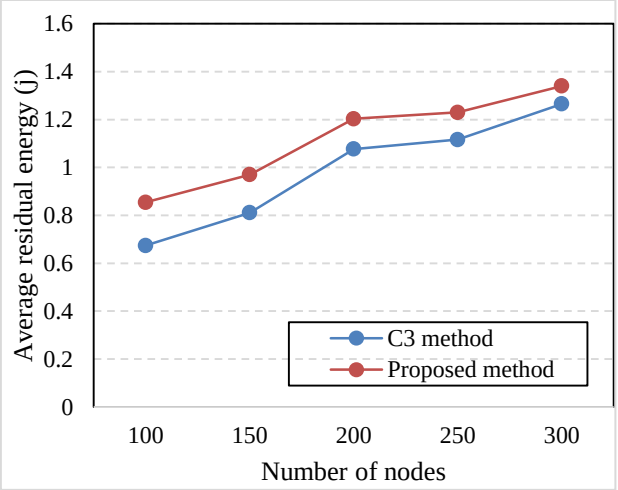


Fig. 9. Average residual energy of nodes in the second scenario

- **Third Scenario:** sink is located in coordinate (130, 65) of the environment. According to figure 12, we can conclude that with the increasing the number of nodes, the average residual energy of sensor nodes increases, but the rate of increase in the proposed method is greater than C3 [27] method. The simulation results of the three scenarios indicate that in the presence of different number of sensor nodes, the average residual energy of network nodes in the proposed method is higher than C3 method and as a result, the network lifetime of the proposed method is also higher than C3 method.

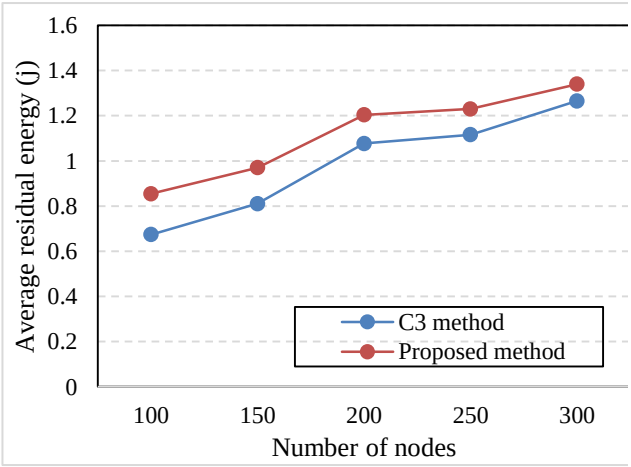


Fig. 10. Average residual energy of nodes

5- Conclusions

In this paper, a new method to improve the network coverage percentage and prolong the lifetime of wireless sensor networks was proposed. In the proposed method, after clustering the network, firstly inside the clusters each node creates the action and probability vectors of learning automata for itself and its neighbors, then determines the status of itself and all its neighbors and finally sends them in the form of a message to the cluster head. Afterward, each cluster head starts to reward or penalize the vectors within the received messages and sends the results to the sender of each message for updating purposes. Thereafter, among the sent vectors, the cluster head node selects the best action vector and broadcasts it in the form of a message inside the cluster. Finally, each member changes its status in accordance with the vector included in the received message from the corresponding cluster head and the active sensor nodes perform environment monitoring operations. The environment monitoring operations are performed by the active nodes until the beginning of next clustering round or termination of the cluster head lifetime. The simulation results showed that the proposed method in terms of the network coverage percentage and the average residual energy of nodes performs better than C3 method. As a future work, we will perform the proposed scheme in other areas with holes or obstacles.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," Computer networks, vol. 38, no. 4, pp. 393-422, 2002.

[2] D. KeyKhosravi, A. Ghaffari, A. Hosseinalipour, and B. A. Khasragi, "New Clustering Protocol to Decrease Probability Failure Nodes and Increasing the Lifetime in WSNs," Int. J. Adv. Comp. Techn., vol. 2, no. 2, pp. 117-121, 2010.

[3] A. Ghaffari, "Designing a wireless sensor network for ocean status notification system," Indian Journal of Science and Technology, vol. 7, no. 6, pp. 809-814, 2014.

[4] A. Ghaffari, "Congestion control mechanisms in wireless sensor networks: A survey," Journal of network and computer applications, vol. 52, pp. 101-115, 2015.

[5] A. Seyfollahi and A. Ghaffari, "A lightweight load balancing and route minimizing solution for routing protocol for low-power and lossy networks," Computer Networks, vol. 179, p. 107368, 2020.

[6] A. Seyfollahi and A. Ghaffari, "Reliable data dissemination for the Internet of Things using Harris hawks optimization," Peer-to-Peer Networking and Applications, vol. 13, no. 6, pp. 1886-1902, 2020.

[7] J. A. Torkestani, "An adaptive energy-efficient area coverage algorithm for wireless sensor networks," Ad hoc networks, vol. 11, no. 6, pp. 1655-1666, 2013.

[8] S. M. Mohamed, H. S. Hamza, and I. A. Saroit, "Coverage in mobile wireless sensor networks (M-WSN): A survey," Computer Communications, vol. 110, pp. 133-150, 2017.

[9] A. Ghaffari and A. Rahmani, "Fault tolerant model for data dissemination in wireless sensor networks," in Information Technology, 2008. ITSim 2008. International Symposium on, 2008, vol. 4, pp. 1-8: IEEE.

[10] H. D. Nikokheslat and A. Ghaffari, "Protocol for Controlling Congestion in Wireless Sensor Networks," Wireless Personal Communications, vol. 95, no. 3, pp. 3233-3251, 2017.

[11] P. Natarajan and L. Parthiban, "k-coverage m-connected node placement using shuffled frog leaping: Nelder–Mead algorithm in WSN," Journal of Ambient Intelligence and Humanized Computing, pp. 1-16, 2020.

[12] C. Shivalingegowda and P. Jayasree, "Hybrid gravitational search algorithm based model for optimizing coverage and connectivity in wireless sensor networks," Journal of Ambient Intelligence and Humanized Computing, pp. 1-14, 2020.

[13] M. El-Hosseini, H. ZainEldin, H. Arafat, and M. Badawy, "A fire detection model based on power-aware scheduling for IoT-sensors in smart cities with partial coverage," Journal of Ambient Intelligence and Humanized Computing, pp. 1-20, 2020.

[14] R. Yarinezhad and S. N. Hashemi, "A sensor deployment approach for target coverage problem in wireless sensor networks," Journal of Ambient Intelligence and Humanized Computing, pp. 1-16, 2020.

[15] A. Pakmehr and A. Ghaffari, "Coverage Improving with Energy Efficient in Wireless Sensor Networks," Information Systems & Telecommunication, p. 57.

[16] M. A. Benatia, M. h. Sahnoun, D. Baudry, A. Louis, A. El-Hami, and B. Mazari, "Multi-Objective WSN Deployment Using Genetic Algorithms Under Cost, Coverage, and Connectivity Constraints," Wireless Personal Communications, vol. 94, no. 4, pp. 2739-2768, 2017.

[17] A. Xenakis, F. Foukalas, G. Stamoulis, and I. Katsavounidis, "Topology control with coverage and lifetime optimization of wireless sensor networks with unequal energy distribution," Computers & Electrical Engineering, 2017.

[18] J. Roselin, P. Latha, and S. Benitta, "Maximizing the wireless sensor networks lifetime through energy efficient connected coverage," Ad Hoc Networks, vol. 62, pp. 1-10, 2017.

[19] D. Thomas, R. Shankaran, M. Orgun, and S. Mukhopadhyay, "A secure barrier coverage scheduling framework for WSN-based IoT applications," IEEE Transactions on Green Communications and Networking, 2021.

[20] C. Shivalingegowda and P. Jayasree, "Hybrid gravitational search algorithm based model for optimizing coverage and connectivity in wireless sensor networks," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 2, pp. 2835-2848, 2021.

[21] J. Yu, Y. Chen, L. Ma, B. Huang, and X. Cheng, "On connected target k-coverage in heterogeneous wireless sensor networks," Sensors, vol. 16, no. 1, p. 104, 2016.

[22] N.-T. Le and Y. M. Jang, "Energy-efficient coverage guarantees scheduling and routing strategy for wireless sensor networks," International Journal of Distributed Sensor Networks, vol. 11, no. 8, p. 612383, 2015.

[23] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections," in Information Fusion, 2002. Proceedings of the Fifth International Conference on, 2002, vol. 2, pp. 1581-1587: IEEE.

[24] K.-P. Shih, Y.-D. Chen, C.-W. Chiang, and B.-J. Liu, "A distributed active sensor selection scheme for wireless sensor networks," in Computers and Communications, 2006. ISCC'06. Proceedings. 11th IEEE Symposium on, 2006, pp. 923-928: IEEE.

[25] S. Soro and W. B. Heinzelman, "Cluster head election techniques for coverage preservation in wireless sensor networks," Ad Hoc Networks, vol. 7, no. 5, pp. 955-972, 2009.

[26] H. Mostafaei, A. Montieri, V. Persico, and A. Pescapé, "A sleep scheduling approach based on learning automata for WSN partial coverage," Journal of Network and Computer Applications, vol. 80, pp. 67-78, 2017.

[27] M. Akhlaq, T. R. Sheltami, and E. M. Shakshuki, "C3: an energy-efficient protocol for coverage, connectivity and communication in WSNs," Personal and Ubiquitous Computing, vol. 18, no. 5, pp. 1117-1133, 2014.

[28] Z. Sun, Y. Zhang, Y. Nie, W. Wei, J. Lloret, and H. Song, "CASMOC: a novel complex alliance strategy with multi-objective optimization of coverage in wireless sensor networks," wireless Networks, vol. 23, no. 4, pp. 1201-1222, 2017.

[29] A. Khelil and R. Beghdad, "SPMI: Single Phase Multiple Initiator Protocol for Coverage in Wireless Sensor Networks," Wireless Personal Communications, vol. 96, no. 2, pp. 3159-3178, 2017.

[30] T. G. Nguyen, C. So-In, N. G. Nguyen, and S. Phoemphon, "A novel energy-efficient clustering protocol with area

coverage awareness for wireless sensor networks," Peer-to-Peer Networking and Applications, vol. 10, no. 3, pp. 519-536, 2017.

[31] Y. Hu, Y. Niu, J. Lam, and Z. Shu, "An energy-efficient adaptive overlapping clustering method for dynamic continuous monitoring in WSNs," IEEE Sensors Journal, vol. 17, no. 3, pp. 834-847, 2016.

Seyyed Keyvan Mousavi received the B.Sc. degree in Software Engineering from the Payame Noor University, Miandoab, Iran, in 2012, the M.Sc. degree in Software Engineering from the Islamic Azad University, Miandoab branch, in 2017, and Ph.D. in software engineering from Urmia branch in 2020. His research interests include Internet of things, Information security, and Cryptography. Currently his research work is based on Security and Privacy in IoT networks.

Ali Ghaffari received his BSc, MSc and Ph.D. degrees in computer engineering from the University of Tehran and IAU (Islamic Azad University), TEHRAN, IRAN in 1994, 2002 and 2011 respectively. He has served as a reviewer for Applied Soft Computing, Ad Hoc networks, Future Generation Computer System (FGCS), Elsevier, International Communication System (IJCS), Journal of Ambient Intelligent and Humanized Computing (AIHC), Wireless Networks, Wireless Personal Communication. As an associate professor of computer engineering at Islamic Azad University, Tabriz branch, IRAN, his research interests are mainly in the field of software defined network (SDN), Wireless Sensor Networks (WSNs), Mobile Ad Hoc Networks (MANETs), Vehicular Ad Hoc Networks (VANETs), networks security and Quality of Service (QoS). He has published more than 100 international conference and reviewed journal papers.

Energy Efficient Cross Layer MAC Protocol for Wireless Sensor Networks in Remote Area Monitoring Applications

R.Rathna*
Department of Information Technology, SRM Institute of Science and Technology, Ramapuram. India
rathnar@srmist.edu.in
L.Mary Gladence
Department of Information Technology, Sathyabama Institute of Science and Technology, India
lgladence@gmail.com
J.Sybi Cynthia
Department of Computer Science and Engineering, School of Engineering, Saveetha Institute of Medical and Technical Sciences.India
cynthia.sybi@gmail.com
V.Maria Anu
Department of Information Technology, Sathyabama Institute of Science and Technology, India
mariaanu18@gmail.com

Received: 29/Apr/2020 Revised: 24/Apr/2021 Accepted: 29/May/2021

Abstract

Sensor nodes are typically less mobile, much limited in capabilities, and more densely deployed than the traditional wired networks as well as mobile ad-hoc networks. General Wireless Sensor Networks (WSNs) are designed with electro-mechanical sensors through wireless data communication. Nowadays the WSN has become ubiquitous. WSN is used in combination with Internet of Things and in many Big Data applications, it is used in the lower layer for data collection. It is deployed in combination with several high end networks. All the higher layer networks and application layer services depend on the low level WSN in the deployment site. So to achieve energy efficiency in the overall network some simplification strategies have to be carried out not only in the Medium Access Control (MAC) layer but also in the network and transport layers. An energy efficient algorithm for scheduling and clustering is proposed and described in detail. The proposed methodology clusters the nodes using a traditional yet simplified approach of hierarchically sorting the sensor nodes. Few important works on cross layer protocols for WSNs are reviewed and an attempt to modify their pattern has also been presented in this paper with results. Comparison with few prominent protocols in this domain has also been made. As a result of the comparison one would get a basic idea of using which type of scheduling algorithm for which type of monitoring applications.

Keywords: Clustering; Energy consumption; Load; Medium Access Control; Radio; Scheduling; Wireless Sensor Network.

1- Introduction

A Sensor node in WSN is small, its power supply unit should be very small and also it should support all its operations without degrading the performance. The communication protocol used should be of lightweight and it should not consume more energy. There are lot of issues in all the layers because of the energy constraint. The protocols used in transport layer, network layer, data link layer and physical layer of any WSN [1] should address all of those issues. In order to achieve energy efficiency any protocol involving a single layer is not enough. Few of such cross layer protocols are reviewed and a new cross layer protocol is proposed in this paper.

In a dense WSN, there will be very high data traffic resulting in high energy wastage if there is one to one communication between the sensing node and gateway node with increased collisions. All these factors lead to clustering. So before applying a sleep/wake-up scheduling, all the nodes must be clustered for an energy efficient routing. So a simple distance based hierarchical clustering is proposed. This works in the network layer. For every cluster the scheduling protocol should work locally. A Load and Energy Consumption based Scheduling Algorithm (LECSA) has also been proposed and it works efficiently in the MAC layer.

In this work, the implementation of LECSA is explained and compared with some of the equally good algorithms quantitatively. Most of the current research works are being carried out in the area of Internet of Things (IoT).

* Corresponding Author

Using this technology, all our day today activities can be made automatically manageable from remote places. Sensors play a major role in this domain. Particularly for remote monitoring of these sensors and the popular drone applications, very less energy consuming protocol and median technologies should be used. So the research subject of energy efficient protocol for WSN explained in this paper definitely will be a promising domain for most of the future research works.

2- Related Works

In the preliminary work of Wendi Rabiner Heinzelman et al., [2] the most popular LEACH protocol (Low Energy Adaptive Clustering Hierarchy) was explained and it was a new blooming domain for the energy conservation in WSN. That is taken as backbone for the research work explained here.

In the work of Al-Jemeli, Marwan, and Hussain, F. in [3], a cross-layer network operational model has been introduced for improving the energy efficiency. It integrates the functions of four layers, namely network layer, transport layer, MAC layer and physical layer. The location information about the nodes is used in the routing after discovering the shortest route. It is then used by the MAC layer for adjusting the transmission range of the nodes. The model proposed here minimizes the neighbor discovery process only to the current active routes. This leads to a reduction in the number of control packets transfer at the same time maintains the network packet delivery ratio. A roughly 10% reduction is said to be reduced by using the simulation of this model when compared with the conventional IEEE 802.15.4 based network.

In a survey paper [4] some of the cooperative diversity-enabled Medium Access Control (MAC) protocols for Wireless LANs and Wireless Sensor Networks (WSNs) are discussed. Cooperative diversity is a newly emerged technique to reduce fading and to improve reliability in a wireless environment. In these types of protocols, neighbour nodes act as virtual multiple-input-multiple-output (VMIMO) systems, where they cooperate with the transmitter-receiver pair of the network to send several copies of every packet to the receiver side through independent fading channels. These multiple copies of the same packet are combined at the receiver to reconstruct the original packet. This technique greatly improves reliability by using spatial diversity in the wireless channel.

Wireless Sensor Networks (WSNs) are nowadays used in many important applications such as intrusion detection, target tracking, industrial automation, smart building and so on [5]. In this work, evolution of WSN MAC layer protocols is surveyed and compared in four categories namely asynchronous, synchronous, frame-slotted, and multichannel.

Energy-efficiency is the challenging feature of all Wireless Sensor Network (WSN) applications [6]. Many WSN MAC (Medium Access Control) protocols use duty-cycling schemes, where the nodes need to switch a node's radio between active and sleep modes. However, a node needs to know in advance whether its neighbour node is active or not. Asynchronous duty-cycling schemes are always advantageous over synchronous ones to eliminate the need of clock synchronization. Many Asynchronous protocols are compared with respect to their latency in this works.

The importance of time synchronization for data consistency and coordination is discussed in detail in [7]. This protocol achieves better performance. Error rate is also greatly reduced here. It has achieved an accuracy of 0.3 microseconds in a multi-hop flat network while using five-times lesser energy than that of FTSP in simulation. It also works better in a clustered network.

Whenever wireless technologies are needed in industrial applications Wireless Sensor Networks (WSNs) are the best possible solutions. But for successful usage, any WSN should provide energy efficiency, scalability, reliability, and timeliness. Few IEEE 802.15.4 WSNs are surveyed and has been found out that there are unreliability issues. This reliability problem arises when measures are taken for enabling good power management. This finally leads to less packet delivery ratio. This problem also arises when there is less number of nodes in the network. In [8], a number of such scenarios are analysed both in simulation and real time and concluded that the reliability issue arises because of the usage of contention-based Medium Access Control (MAC) protocol with their default parameter values. So they have concluded that if appropriate values are given instead of default values for the MAC parameters, then 100% delivery ratio can be achieved. But flexible values for MAC parameters cannot be formally assigned according to the standard.

In this work [9], the energy efficiency and reliability issues of WSN are discussed in detail. They have proposed a new cross-layer framework which satisfies both the energy efficiency and reliability required for the application. This framework finds out the reliability requirements and configures the MAC layer based on the deployment of nodes and traffic at that instant. This new distributed and low complexity algorithm called Adaptive Access Parameters Tuning (ADAPT) makes the WSN to meet the reliability requirements of the concerned application. This is said to be successful for both single hop and multi hop IEEE 802.15.4 networks.

The newly emerged Vehicular Sensor Networks (VSNs) branch of WSN has promising future in the current scenario [10]. In this work they proposed a new vehicular clustering scheme in VSN for energy efficient routing. It is

compared with the existing algorithms like Direct, LEACH and DCHS.

Shanti, Chilukuri, and AnirudhaSahoo have proposed an integrated MAC and routing protocol called Delay Guaranteed Routing and MAC (DGRAM) [11] for WSN applications where delay is not desirable. This TDMA-based protocol provide delay guarantee with energy efficiency. It uses the slot reuse mechanism. At the same time the medium should be contention free. The sending and receiving nodes' slots are computed in such a way that apart from sending and receiving time they can sleep to reduce the energy consumption. Hence a separate routing protocol is not needed. Simulation results of this DGRAM are also provided. And it is compared with the results of FlexiTP. It has shown a less delay when compared with the FLexiTP without compromising the energy efficiency.

Several low-cost embedded devices are now available which can communicate with each other and different wireless devices [12]. Such multifunctional low end devices include RFID. A number of such RFID enabled devices come along with small sized cameras. These devices are deployed in harsh places. These nodes form an adhoc WSN. For several days or even months they should monitor for the desirable factors in such environment without power back-up and maintenance. In this work K.W.Chin has proposed a TDMA based protocol called PairWise for large scale WSNs. It does not follow global synchronization. Instead each node synchronizes only with its neighbours by establishing a pair of channels. Since the channels can hop in time using seed and maximum rendezvous period (MRP), the collisions are greatly reduced. The simulation results of this protocol are compared with that of S-MAC and another TDMA MAC protocol.

Choi et al., in their work [13] discussed briefly about the listen and sleep mode switching of nodes in WSNs. Although this mode switching helps in the reduction of power consumption, it increases the delay. If a packet is ready to be sent from a node and at that time, according to the sleep cycle, if the node goes to sleep, than that packet has to wait for a long time. So in this work, the authors proposed a new routing protocol called Average Velocity based Routing (AVR) protocol which takes into account the node location as well as the packet waiting time. Each node calculates the n-hop average velocity of each of its neighbour nodes and then decides to which node, it has to forward the data packet. If the knowledge range n increases then automatically the delay decreases.

Reference [14] describes about the low – power duty cycle MAC protocols in their work and about the trade off between energy efficiency and latency in such protocols. To reduce the latency problem they have proposed an Express MAC protocol (EX-MAC). It uses a short preamble scheme for sleep/wake-up scheduling to reduce the delay. Using a cross-layer API, it provides convergecast packets for performance optimization. Using simulation, they have proven that EX-MAC provides adjustable wake-up time reservation for convergecast traffic and thereby reduces the delay time.

WSN for detecting unusual events use multimodality sensor nodes. They are used for applications to avoid accidents in roadside [15]. For such applications, an event oriented and application specific MAC protocol (App-MAC) is proposed in this work. According to the application requirements and current event status, the time slots are allocated for the nodes. They have executed the protocol in TOSSIM simulator and in real time using Berkeley TelosB motes. It has also been proved that this protocol outperforms S-MAC and Traffic Adaptive Medium Access (TRAMA) protocols.

The work [16] helps to understand the energy model used for mobile devices. In particular, to decide which type of protocol can be used for sending the multimedia files over wireless networks. The Reed Solomon (RS) MAC layer coding scheme is focused. For analysing the energy consumption, they have calculated the energy consumed by each computational unit of the RS decoder of ARM7TDMI test-bed in real time. And it has been concluded that using appropriate error free control block based on channel conditions required for the multimedia files will reduce the energy wastage significantly.

Usage of two key physical layer parameters namely channel state and the residual node energy for the design of distributed MAC protocol is explained in detail in [17]. This design is aimed at improving the network lifetime. It has been suggested that when the network is young the sensor nodes are provided with better channels for data transmission and when the network is old, the nodes with more residual energies are given priority. Using this principle a protocol called Dynamic Protocol for Lifetime Maximization (DPLM) is proposed. It utilizes both the channel state and the residual node energy. Its results are also compared with the existing protocols.

A cross-layer protocol named Low Energy Self-Organizing Protocol (LESOP) is presented in [18]. It uses the interaction between MAC layer and the application layer for high efficiency. This innovative model takes care of the error control and network energy consumption issues by using a QOS knob. This is the first work using Embedded Wireless Interconnect (EWI) architecture platform instead of the traditional OSI layer. It is a light weight protocol using only 2 layers to reduce the complexity.

Fallahi, Afshin, and EkramHossain developed a queueing framework for analyzing their newly proposed distributed and energy aware MAC protocol [19]. In that protocol, a node is assumed as a priority queue with a low priority and high priority traffic. The node's MAC layer is modeled as a server and a vacation queuing model is used to model the sleep and wakeup mechanism of the MAC layer. The factors such as packet dropping probability, access delay,

and queue length distribution, for high-priority packets and also the energy saving factors are analyzed in this framework under different conditions. They have also validated this protocol using extensive simulation.

Even though there is a new standard - IEEE 802.11a/b/g for supporting multiple data rates in Wireless LANs, there is always a difference in the availability of high data rate between the node near the access point and the one in the far away place near the border of the access point's range. A study using the combination of Medium-access and next-hop address lookup based on fixed-length labels without taking into account the host protocol stack was done in [20]. And based on that they have proposed Energy-Efficiency Cut-Through Protocol which selects the appropriate PHY model and a transmission power level that to achieve the maximum possible energy efficiency.

The work in [21] describes the usage of network coding in wireless networks along with MAC). Network coding has a lot of advantages. But the properties of wireless networks necessitate the time-varying network coding to interact with the MAC layer. This is contradictory with the classical approach used in wired network coding. A new method is presented in their work to construct linear wireless network codes by interacting with MAC schedules. Another method is also presented which creates conflict-free transmission schedules by splitting the wireless network into subtrees and applying graph coloring on them. Then the normal routing is compared with that of this network coding in terms of energy, delay and throughput.

Even in the recent survey paper [22], LEACH is taken as the important energy efficient clustering and scheduling protocol as the benchmark protocol. Hence that is taken as the first protocol for comparing the result of implementation of the proposed algorithm.

A new radio technology, Impulse-based ultra wideband (I-UWB) for large scale sensor networks was given in [23]. MAC protocols for I-UWB is centralized. But when the network size increases, only distributed MAC can be used. So a single protocol stack is impossible. Three MAC protocols are proposed. Two multi channel protocols, namely multichannel pulse sense multiple access (M-PSMA) and multichannel ALOHA and a busy-signal protocol, called Busy-Signal Multiple Access (BSMA). The simulation of the physical layer confirms the feasibility of the proposed protocols in wide WSN environments. The results show that the BSMA outperforms other centralized TDMA protocols, and has more energy efficiency when compared with that of the other distributed protocols.

2-1- Non Orthogonal Multiple Access (NOMA) Systems

The traditional Time/Code/Frequency Division Multiple Access come under the category of Orthogonal Multiple Access Scheme. The System described here uses Time Division Multiple Access Mechanism where the different nodes in the WSN will be using the same frequency band but each one will be given different time slots. Similarly the other two namely Code and Frequency division Multiple Access use different code sequence and different frequency bands to allot for multiple users or multiple nodes in a Wireless Sensor Network respectively.

Nowadays, the Non Orthogonal Multiple Access scheme allocates multiple users or nodes for medium access in same time, frequency and code by using different factors of Power and Code. In the work presented in [24], for a mixed type traffic, a new type of resource allocation mechanism was described for both distributed and centralized Antenna types.

A new scheme called green NOMA is used in the work of Basnayake et al. [25]. A comparison on energy efficient NOMA schemes were given elaborately in their work. The schema, they have proposed is mainly designed for multiple access in 5G communication.

But for monitoring of harmful radiations in a remote environment, a simple yet efficient mechanism is required. Hence TDMA based multiple access scheme is used throught in the work presented here.

3- Distance based Hierarchical Clustering

The WSN explained in this work clusters the nodes in a hierarchical manner. Randomly a node is taken as the initial node. From that node, the other nodes which are only one and two hops away are formed as a cluster. If a node goes out of this distance, then that one becomes the initial node for the next cluster. The first node which comes out of this condition is taken as the initial node for the second cluster. Euclidean distance is used for the cluster formation. The underlying concept is that the nodes within a short distance are grouped as a cluster.

In our test bed, 25 meters is set as maximum allowable distance between any two sensors in a cluster. The distance between the other sensor nodes from the random first node are calculated and decided to be placed in the same cluster or a different one using this 25 meters. Repeatedly this clustering process is done until all the nodes are associated with a particular cluster.

3-1- Algorithm

Clustering here is completely based on proximity. 25 meters is kept as the maximum distance of coverage for any node within a cluster. If any node's location is out of

this range,

	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6
Nodes in Cluster	1	11	2	6	12	21
	3	20	8	43	18	26
	7	25	10	98	23	29
	13	34	14		27	31
	17	37	15		38	33
	28	46	16		61	40
	32	49	19		67	52
	39	50	22		81	56
	45	51	41		82	66
	57	54	44		91	68
	62	55	47		94	71
	86	58	48			76
	97	60	65			84
		63	70			93
		64	74			
		77	78			
		85	99			
		87				
		92				
		95				
		96				
		100				

	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6
Nodes(IDs) in Clusters	1	2	3	4	5	9
	7	6	17	15	21	18
	13	8	39	22	29	20
	24	10	57	41	30	34
	28	14	86	70	33	38
	32	16	97	72	40	42
	35	19		75	52	55
	36	37		79	56	85
	45	43			80	90
	53	44			88	96
	59	46				
	62	47				
	69	48				
	73	54				
	83	64				
	89	77				
		78				
		95				
		98				
		99				

	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5	Cluster6
Nodes(IDs) in Clusters	1	2	3	4	5	8
	6	23	16	15	7	17
	12	27	24	44	9	25
	18	31	47	62	11	30
	36	53	58	69	14	33
	37	54	64	84	19	35
	42	57	95	98	20	45
	48	79		100	34	49
	50	82			38	61
	51	85			40	68
	52	86			41	70
	55	87			43	72
	60	93			46	75
	76	99			59	80
	78				63	
	90				67	
	91				81	
					83	
					89	
					92	

Fig. 1. Cluster formation using the proposed clustering algorithm using three different starting nodes

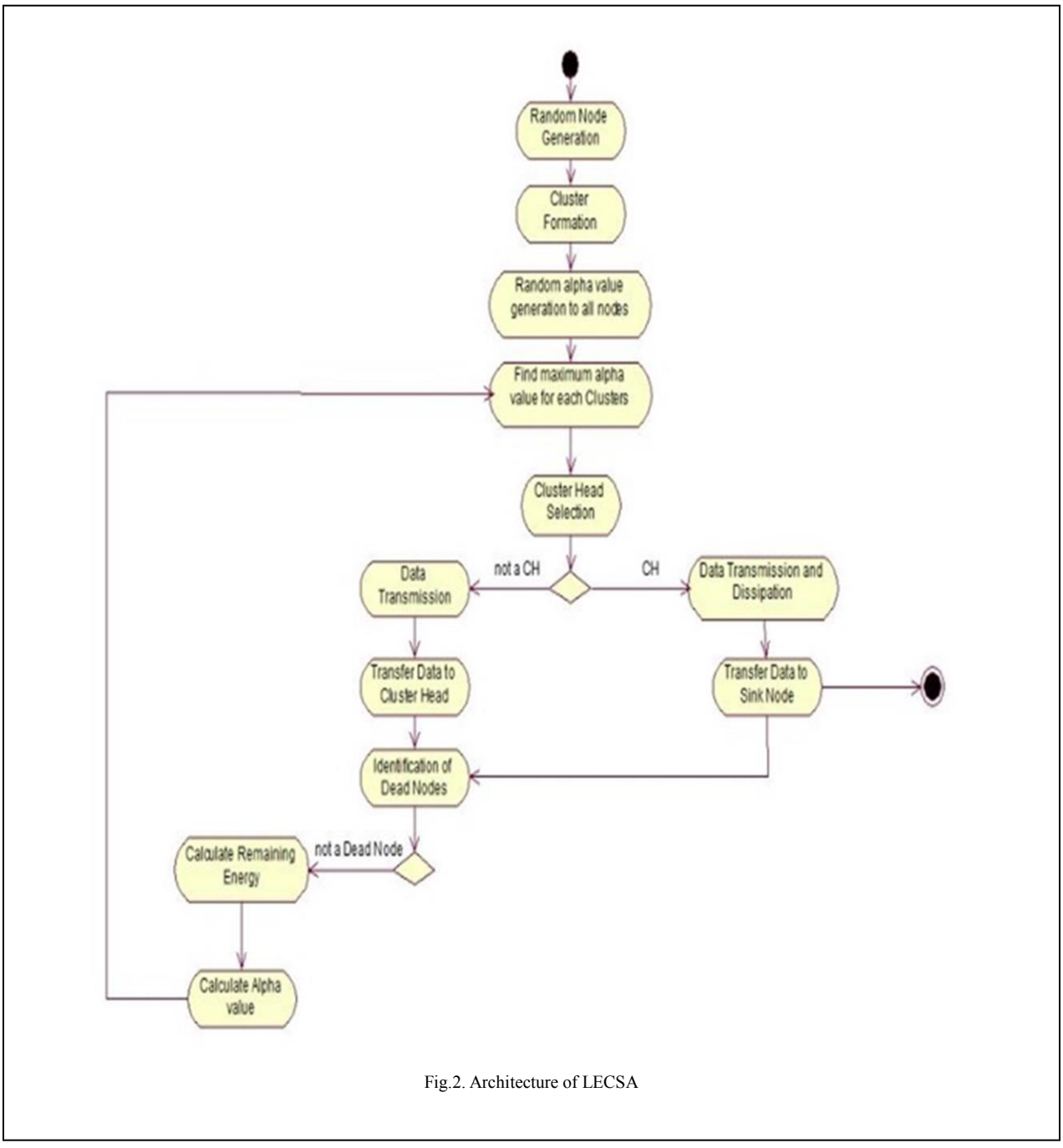


Fig.2. Architecture of LECSA

then that initiates the process of next cluster formation. This simple strategy is followed as algorithm (given down) for clustering.

Step1: Start the outer for loop

Step2: Start the inner for loop with a limit of 'n' nodes

Step3: Calculate Euclidean distance (dis) =((S(i).xd)-Xk)2 + ((S(i).yd)-Yk)2

Step4: Calculate d= $\sqrt{dis}$

Step5: If calculated value is greater than 25 and also the node S(i)'s flag is false, do:

Set S(i).C as 1

Set S(i)'s flag as true

Plot the node S(i) using a specific color

Mark the node id

Step6: Close the inner for loop

Step7: Start a small for loop with limt 'n'

Step8: Check for S(i).flag value. If it is false, store its location

Step9: Close the inner for loop

Step10: Close the outermost for loop

A Boolean variable is used for every node to indicate whether it is inside any cluster or not. This algorithm can be used for clustering of any number of nodes as it is proximity based. But the drawback here is the cluster size, which differs based on the distribution of nodes on the field. For the work presented here, 100 nodes are used. By applying the algorithm, six clusters are formed (Fig.1)

The turn of becoming the Cluster Head (CH) depends on 'α'-the node weighting parameter. For the first round, random values are assigned for α for finding the CH. For the remaining rounds, this is calculated based on the payload (L) and residual node energy (E) using equation (1).

$$S(i).\alpha = S(i).E + (1/S(i).L) \qquad (1)$$

Forwarding of data from nodes to the base station happens in an order based on the value of alpha. This value is taken as the critical parameter for the entire scheduling algorithm explained in the next section.

4- Load and Energy Consumption based Scheduling Algorithm

The main function of any MAC protocol is to control the usage of the medium, and this is done through a channel access mechanism. It provides a scheme to divide the main resource that is the medium between the sensor nodes. Medium Access Control protocols are of 3 types. They are Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA), and Code Division Multiple Access (CDMA).

TDMA based schemes provide inherent collision free access mechanisms. Hence this research work is focussed on developing energy efficient TDMA based MAC Protocol for scheduling the nodes. According to the proposed scheduling scheme (Fig.2), whenever a node of a cluster senses some data, its residual energy has to be checked initially (using equation (2)).

$$E_{iTx} = (E_{Elec} * b) + (\varepsilon_{amp} * b * d^2)$$

$$E_{iRx} = E_{Elec} * b$$

$$E_{it} = E_{it} - (\sum E_{iTx} + \sum E_{iRx}) \qquad (2)$$

If the residual energy of a node is positive, then according to it's α, a slot is allocated. During the allotted slot, the data sensed by the node is sent to the one which is having the next biggest α in the schedule. During every round, the CH collects the data from all its nodes at the end. Automatically the node which occupies the last slot gets the position of CH. The maximum 'α' value will be held by the CH. The sink node gets the data from the sensing node by strictly adhering to the schedule. It follows the pattern of a CH passing its payload to its nearest neighbour in the next closer cluster. That particular node passes the data to its CH. This pattern is used throughout until the data reaches the Base Station (BS).

5- Analysis of the Proposed Scheme

The performance of the proposed schemes is analyzed by the simulation for three iterations. Because of using the proposed Clustering and Scheduling schemes, dynamically the CH changes every time inside each cluster. So the node acting as CH will not drain out fast. Fig.3. depicts this.

Let us take cluster No.1 and analyze. The comparison is done for checking the dynamic nature of CH by analyzing the nodes having CH responsibility in three iterations. The graphs given in Fig.3. shows how dynamically the CH position is changing.

From this we can also infer that only two or three nodes are getting the CH position repeatedly. The overall energy of the WSN, at the end of certain rounds is also compared. That tabulation for the first, second and third iterations are given in Fig.4, Fig.5, Fig.6.

From the several rounds of simulation, it has been found that the nodes start draining from round 1000 onwards. At the end of 1100 rounds, the degradation starts. For about

100 rounds the WSN was able to run with limited number of nodes and not much loss.

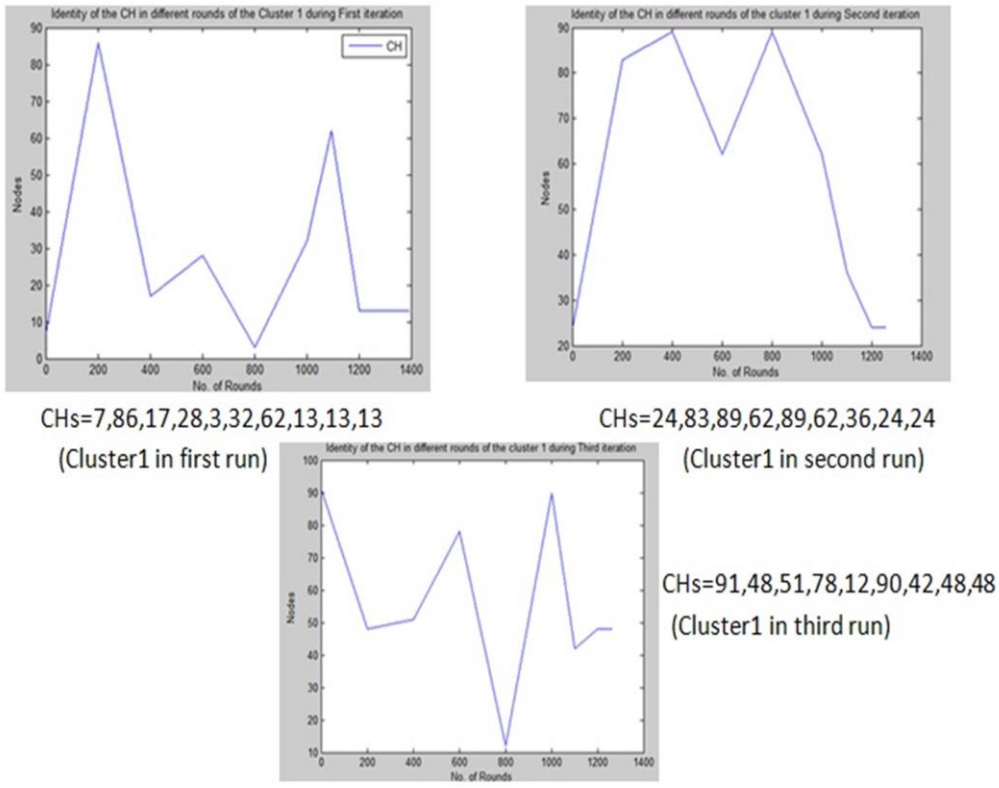


Fig. 3. Dynamically changing behavior of CH position

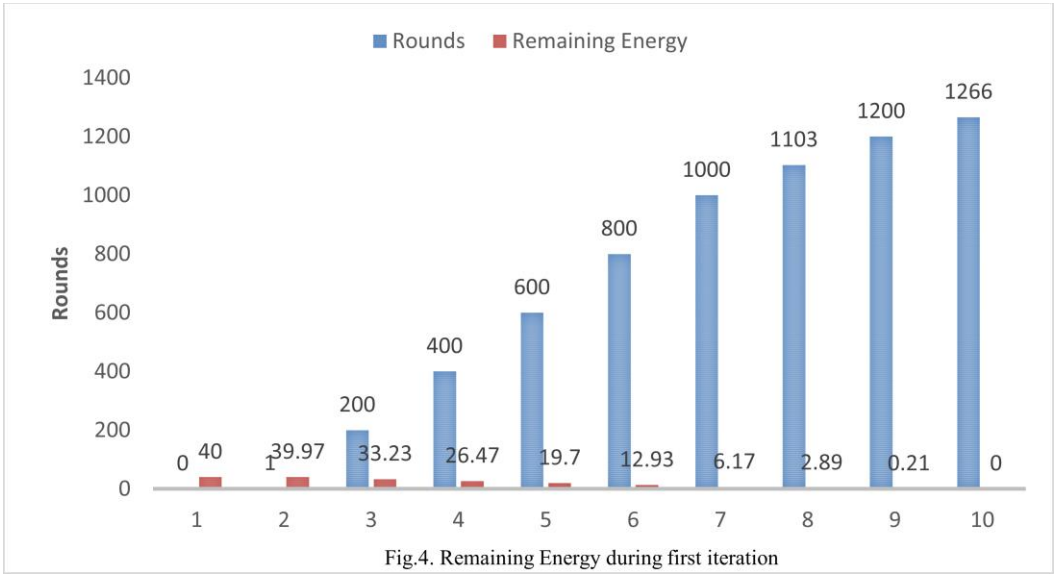


Fig.4. Remaining Energy during first iteration

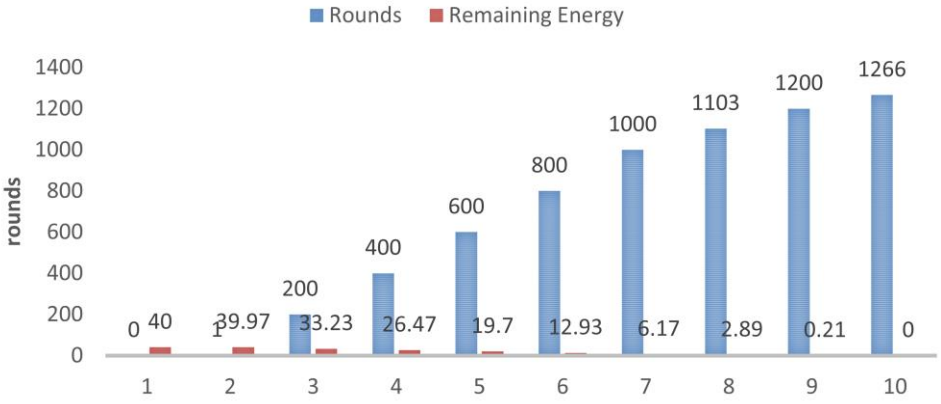


Fig.5. Remaining Energy during second iteration

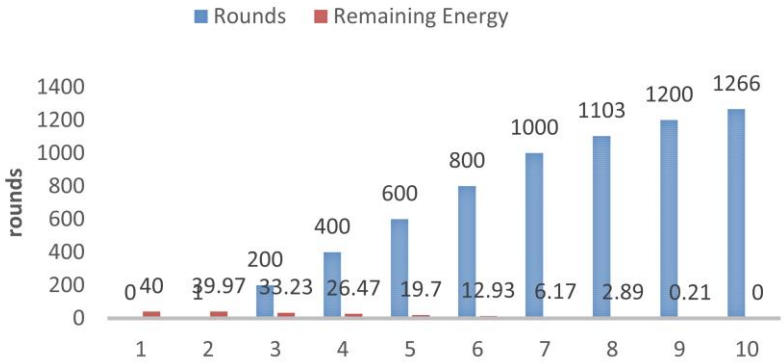


Fig.6. Remaining Energy during third iteration

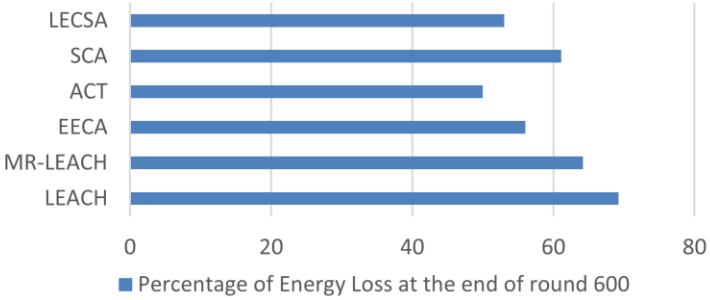


Fig.7. Percentage of Energy Loss at the end of round 600

5-1- Comparison based on Percentage of Energy loss with the Existing Protocols

The comparison is based on percentage of energy loss for the proposed schema with the tabulate percentage of energy loss for the proposed schema with the existing protocols, MR-LEACH (M. O. Farooq, A. B. Dogar, and G. A. Shah 2010), a cluster-based routing protocol called ACT (W. K. Lai, C. F. Fan, and L. Y. Lin 2012), an Energy efficient Clustering Algorithm –EECA (C.sha et al 2010) and a Separating Cluster-based Algorithm-SCA (Shuai Fu et al 2013).

The results are shown in Fig.7. The network simulation is done in NS2. Based on the data given by the authors of the similar protocols mentioned in the previous paragraph, the following inference is made.

From the values given in table (Table 4), it is obvious that the work proposed runs efficiently in terms of energy than that of the existing algorithms like LEACH, SCA and MR-LEACH, where balance in energy is achieved by adjusting among the sensors in each cluster and also compared with EECA where the cluster heads are alone assigned with weights to achieve energy efficiency. ACT is the algorithm which gives only 50 % energy loss after 600 rounds. They used the strategy of changing the cluster radius. But this strategy cannot followed for complex networks [26].

6- Conclusion

From the outcomes of the simulations, it has been observed that the nodes will not get depleted of energy as fast as that of the existing algorithms (Except ACT). The high priority is always given to the node with least energy. This simple technique reduces the overall energy consumption. Some limitations are 1) the developed LECSA is suitable only for WSN with multiple nodes used in Environmental Monitoring applications. 2) Nodes with camera as sensor, sends and receives video using compression [27] (T.Bernatin and G.Sundari 2015). For those types this protocol is tough to be implemented, since data aggregation leads to loss in actual video. 3) This algorithm design is applicable for Wireless Sensor Nodes working on Zigbee radio having low range communication. Accordingly, the distance is maintained between the nodes. In future, along with the parameters namely residual node energy and load, other parameters like distance, location and data priority of the node can also be taken for assigning the slots for scheduling. The entire algorithm was designed and implemented according to TDMA based multiple node access scheme. NOMA is the recently used multi access strategy and many research works in WSN and IOT are being carried out using this scheme. Hence the same implementation presented here can be extended in future using the NOMA.

References

[1] Jennifer Yick, Biswanath Mukherjee and Dipak Ghosal, “Wireless Sensor Network Survey” in Computer Networks, Elsevier, 2008, 52(12), 2292-2230.

[2] Wendi Rabiner Heinzelman, Anantha Chandrakasan, Hari Balakrishnan, “Energy Efficient Communication Protocol for Wireless Microsensor Networks” , Poceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, Hi, USA, 2000, Vol.2. pp.10.

[3] Al-Jemeli, Marwan, and F. Hussin, “An energy efficient cross-layer network operation model for IEEE 802.15. 4-based mobile Wireless sensor networks” in Sensors Journal, 2015, 15 (2) 684-692.

[4] Khan, RanaAzeem M., and Holger Karl, “Mac protocols for cooperative diversity in wireless lans and wireless sensor networks” in Communications Surveys & Tutorials, IEEE, 2015, 16(1), 46-63.

[5] Huang, Pei, et al., “The evolution of MAC protocols in wireless sensor networks: A survey” in Communications Surveys & Tutorials, IEEE. 2014, 15(1), 101-120.

[6] Doudou, Messaoud, DjamelDjenouri, and NadjibBadache, “Survey on latency issues of asynchronous mac protocols in delay-sensitive wireless sensor networks” in Communications Surveys & Tutorials, IEEE, 2013,15(2), 528-550.

[7] Akhlaq, Muhammad, and Tarek R. Sheltami, “RTSP: An accurate and energy-efficient protocol for clock synchronization in WSNs” in IEEE Transactions on Instrumentation and Measurement, 2013, 62(3), 578-589.

[8] Anastasi, Giuseppe, Marco Conti, and Mario Di Francesco, “A comprehensive analysis of the MAC unreliability problem in IEEE 802.15. 4 wireless sensor networks” in IEEE Transactions on Industrial Informatics, 2011, 7(1), 52-65.

[9] Di Francesco, Mario, et al. “Reliability and Energy-Efficiency inIEEE 802.15. 4/ZigBee Sensor Networks: An Adaptive and Cross-Layer Approach” in IEEE Journal on Selected Areas in Communications, 2011, 29(8), 1508-1524.

[10] Liu, Yuhua, et al., “Multi-layer clustering routing algorithm for wireless vehicular sensor networks” in IET communications, 2010, 4(7), 810-816.

[11] Shanti, Chilukuri, and AnirudhaSahoo, “DGRAM: a delay guaranteed routing and MAC protocol for wireless sensor networks” in IEEE Transactions on Mobile Computing, 2010, 9(10), 1407-1423.

[12] Chin, Kwan-Wu “Pairwise: a time hopping medium access control protocol for wireless sensor networks” in IEEE Transactions on Consumer Electronics, 2009, 55(4), 1898-1906.

[13] Choi, Sung-Chan, Seong-Lyong Gong, and Jang-Won Lee, “An average velocity-based routing protocol with low end-to-end delay for wireless sensor networks” in Communications Letters, 2009, IEEE. 13(8), 621-623.

[14] Hong, Sung-Hwa, and Hoon-ki Kim, “A multi-hop reservation method for end-to-end latency performance improvement in asynchronous MAC-based wireless sensor networks” in IEEE Transactions on Consumer Electronics, 2009, 55(3), 1214-1220.

[15] Du, Junzhao, and Weisong Shi, “App-MAC: An application-aware event-oriented MAC protocol for multimodality wireless sensor networks” in IEEE

Transactions on Vehicular Technology, 2008, 57(6), 3723-3731.
[16] Kang, Kyungtae, Yongwoo Cho, and Heonshik Shin, "Energy-efficient MAC-layer error recovery for mobile multimedia applications in 3GPP2 BCMCS" in IEEE Transactions on Broadcasting, 2007, 53(1), 338-349.
[17] Chen, Yunxia, and Qing Zhao, "An integrated approach to energy-aware medium access for wireless sensor networks" in IEEE Transactions on Signal Processing, 2007, 55(7), 3429-3444.
[18] Song, Liang, and Dimitrios Hatzinakos, "A cross-layer architecture of wireless sensor networks for target tracking" in IEEE/ACM Transactions on Networking, 2007, 15(1), 145-158.
[19] Fallahi, Afshin, and EkramHossain, "Distributed and energy-aware MAC for differentiated services, wireless packet networks: a general queuing analytical framework" in IEEE Transactions on Mobile Computing, 2007, 6(4), 381-394.
[20] Liu, Jain-Shing, and C-HR Lin, "ECTP: an energy-efficiency label-switching MAC protocol for infrastructure wireless networks" in IEEE Transactions on Vehicular Technology, 2007, 56(3), 1399-1417.
[21] Sagduyu, YalinEvren, and Anthony Ephremides, "On joint MAC and network coding in wireless ad hoc networks" in IEEE Transactions on Information Theory, 2007, 53(10), 3697-3713.
[22] Amin Shahraki, Amir Taherkordi, Øystein Haugen, and Frank Eliassen, "Clustering objectives in wireless sensor networks: A survey and research direction analysis",Computer Networks, Volume 180, 2020.
[23] August, Nathaniel J., and Dong Sam Ha, "Operation, system architectures, and physical Layer design considerations of distributed MAC protocols for UWB" in IEEE Transactions on Microwave Theory and Techniques, 2006, 54(7), 3001-3012.
[24] M. Youssef, J. Farah, C. A. Nour and C. Douillard, "Resource Allocation in NOMA Systems for Centralized and Distributed Antennas With Mixed Traffic Using Matching Theory," in IEEE Transactions on Communications, Jan 2020, vol. 68, no. 1, 414-428.
[25] Basnayake, Vishaka; Jayakody, Dushantha N.K.; Sharma, Vishal; Sharma, Nikhil; Muthuchidambaranathan, P. and Mabed, Hakim. 2020. "A New Green Prospective of Non-orthogonal Multiple Access (NOMA) for 5G" Information 11, no. 2: 89. https://doi.org/10.3390/info11020089.
[26] Joshua Onyeka Ogbebor, Agbotiname Lucky Imoize, and Aderemi Aaron-Anthony Atayero, "Energy Efficient Design Techniques in Next-Generation Wireless Communication Networks: Emerging Trends and Future Directions", in Wireless Communications and Mobile Computing, vol. 2020, Article ID 7235362, 19 pages, 2020.
[27] T.Bernatin and G.Sundari, "Video compression based on hybrid transform and quantization with Huffman coding for video codec" in ICCICCT-2014,452-456.

R.Rathna received the B.E degree in Information Technology from Bharathidasan University in 2004, M.Tech degree in Information Technology from Sathyabama University in 2007 and Ph.D in Computer Science and Engineering from Sathyabama Institute of Science and Technology in 2015. S Currently she is working as Assistant Professor in the Department of Information Technology in SRM Institute of Science and Technology, Ramapuram Campus, Chennai, India. Her research interests include Wireless Sensor Networks, Internet of Things, Machine Learning and Data Analysis.

L.Mary Gladence is an Associate Professor in the Department of Information Technology in Sathyabama Institute of Science and Technology for more than 15 years. She finished her Ph.D in the domain of Data Mining in the year 2017 from the Sathyabama Institute of Science and Technology. Her research interests include deep learning, artificial intelligence, data mining, sequential pattern Mining, pattern recognition, machine learning, bio computing and data analytics.

J. Sybi Cynthia received the M.E. degree in Computer Science & Engineering from the Anna University, Tamil Nadu, in 2010, and the Ph.D. degree in Computer Science & Engineering, Manonmaniam Sundaranar University in 2018, Tamil Nadu, India. She is currently working as an Assistant Professor with the Department of Computer Science & Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu. Her research interests include Wireless Sensor Networks and Network Security.

V.Maria Anu obtained her B.Tech Degree from University of Madras in Information Technology, M.E degree and Ph.D in Computer Science and Engineering from Sathyabama Institute of Science and Technology. Currently she is working as Professor in School of Computing, Sathyabama Institute of Science and Technology. She has 17 years of teaching experience. Her research interests include RFID data management, Data Analytics and Image Mining.